Published in Journals: Remote Sensing, Cancers, Mathematics, Sensors and Electronics

Topic Reprint

AI and Data-Driven Advancements in Industry 4.0

Volume II

Edited by
Teng Huang, Qiong Wang and Yan Pang

mdpi.com/topics

MDPI

AI and Data-Driven Advancements in Industry 4.0—Volume II

AI and Data-Driven Advancements in Industry 4.0—Volume II

Topic Editors

Teng Huang
Qiong Wang
Yan Pang

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Topic Editors

Teng Huang
Institute of Artificial Intelligence and Blockchain
Guangzhou University
Guangzhou
China

Qiong Wang
Shenzhen Institute of Advanced Technology
Chinese Academy of Sciences
Shenzhen
China

Yan Pang
Shenzhen Institute of Advanced Technology
Chinese Academy of Sciences
Shenzhen
China

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Topic, published open access by the journals *Remote Sensing* (ISSN 2072-4292), *Cancers* (ISSN 2079-9292), *Mathematics* (ISSN 2227-7390), *Sensors* (ISSN 1424-8220), and *Electronics* (ISSN 2072-4292) , freely accessible at: https://www.mdpi.com/topics/FLN51J9SH1.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

Volume II
ISBN 978-3-7258-4167-7 (Hbk)
ISBN 978-3-7258-4168-4 (PDF)
https://doi.org/10.3390/books978-3-7258-4168-4

Set
ISBN 978-3-7258-4223-0 (Hbk)
ISBN 978-3-7258-4224-7 (PDF)

© 2025 by the authors. Articles in this book are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The book as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (https://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

Ling Zhou, Ruilin Wang and Liyan Zhang
Accurate Robot Arm Attitude Estimation Based on Multi-View Images and Super-Resolution Keypoint Detection Networks
Reprinted from: *Sensors* **2024**, *24*, 305, https://doi.org/10.3390/s24010305 . . . . . **1**

Peng Chen, Yingzhi Liu, Wei Li, Jingyi Wang, Jianxiu Wang, Bei Yang and Gang Feng
Semi-Supervised Learning-Enhanced Fingerprint Indoor Positioning by Exploiting an Adapted Mean Teacher Model
Reprinted from: *Electronics* **2024**, *13*, 298, https://doi.org/10.3390/electronics13020298 . . . . . **22**

Yunyi Yuan, Yingkui Zhang, Lei Zhu, Li Cai and Yinling Qian
Exploiting Cross-Scale Attention Transformer and Progressive Edge Refinement for Retinal Vessel Segmentation
Reprinted from: *Mathematics* **2024**, *12*, 264, https://doi.org/10.3390/math12020264 . . . . . **46**

Dominik Schraml and Gunther Notni
Synthetic Training Data in AI-Driven Quality Inspection: The Significance of Camera, Lighting, and Noise Parameters
Reprinted from: *Sensors* **2024**, *24*, 649, https://doi.org/10.3390/s24020649 . . . . . **63**

Emmanuel Stathatos, Evangelos Tzimas, Panorios Benardos and George-Christopher Vosniakos
Convolutional Neural Networks for Raw Signal Classification in CNC Turning Process Monitoring
Reprinted from: *Sensors* **2024**, *24*, 1390, https://doi.org/10.3390/s24051390 . . . . . **81**

Mohamed Baraya, Mohamed S. El-Asfoury, Omnia O. Fadel and Ahmed Abass
Experimental Analyses and Predictive Modelling of Ultrasonic Welding Parameters for Enhancing Smart Textile Fabrication
Reprinted from: *Sensors* **2024**, *24*, 1488, https://doi.org/10.3390/s24051488 . . . . . **105**

HoeJun Jeong, SeongYeon Jeung, HyunJun Lee and JangWoo Kwon
BiVi-GAN: Bivariate Vibration GAN
Reprinted from: *Sensors* **2024**, *24*, 1765, https://doi.org/10.3390/s24061765 . . . . . **123**

Elena V. Varlamova, Maria A. Butakova, Vlada V. Semyonova, Sergey A. Soldatov, Artem V. Poltavskiy, Oleg I. Kit and Alexander V. Soldatov
Machine Learning Meets Cancer
Reprinted from: *Cancers* **2024**, *16*, 1100, https://doi.org/10.3390/cancers16061100 . . . . . **140**

Jie Cai, Marie Lützen, Adeline Crystal John, Jakob Buus Petersen and Niels Gorm Maly Rytter
Method for Identification of Aberrations in Operational Data of Maritime Vessels and Sources Investigation
Reprinted from: *Sensors* **2024**, *24*, 2146, https://doi.org/10.3390/s24072146 . . . . . **163**

Xiao Tan, Jianfeng Yang, Zhengang Zhao, Jinsheng Xiao and Chengwang Li
Improving Graph Convolutional Network with Learnable Edge Weights and Edge-Node Co-Embedding for Graph Anomaly Detection
Reprinted from: *Sensors* **2024**, *24*, 2591, https://doi.org/10.3390/s24082591 . . . . . **179**

Jun Chen, Yongxi Luo, Jie Wang, Honghua Tang, Yixian Tang and Jianhui Li
Elimination of Irregular Boundaries and Seams for UAV Image Stitching with a Diffusion Model
Reprinted from: *Remote Sensing* **2024**, *16*, 1483, https://doi.org/10.3390/rs16091483 . . . . . **201**

Binzi Xu, Kai Xu, Baolin Fei, Dengchao Huang, Liang Tao and Yan Wang
Automatic Design of Energy-Efficient Dispatching Rules for Multi-Objective Dynamic Flexible Job Shop Scheduling Based on Dual Feature Weight Sets
Reprinted from: *Mathematics* **2024**, *12*, 1463, https://doi.org/10.3390/math12101463 . . . . . **220**

Xiancheng Peng, Yangzhuo Chen, Xiaowen Cai and Jun Liu
An Improved YOLOv7-Based Model for Real-Time Meter Reading with PConv and Attention Mechanisms
Reprinted from: *Sensors* **2024**, *24*, 3549, https://doi.org/10.3390/s24113549 . . . . . **244**

Nisar Hakam, Khaled Benfriha, Vincent Meyrueis and Cyril Liotard
Advanced Monitoring of Manufacturing Process through Video Analytics
Reprinted from: *Sensors* **2024**, *24*, 4239, https://doi.org/10.3390/s24134239 . . . . . **258**

Chenqin Xiong, Tarek Zayed, Xingyu Jiang, Ghasan Alfalah and Eslam Mohammed Abelkader
A Novel Model for Instance Segmentation and Quantification of Bridge Surface Cracks—The YOLOv8-AFPN-MPD-IoU
Reprinted from: *Sensors* **2024**, *24*, 4288, https://doi.org/10.3390/s24134288 . . . . . **274**

Xuejun Li, Xu Lei, Lingli Jiang, Tongguang Yang and Zhenyu Ge
A New Strategy: Remaining Useful Life Prediction of Wind Power Bearings Based on Deep Learning under Data Missing Conditions
Reprinted from: *Mathematics* **2024**, *12*, 2119, https://doi.org/10.3390/math12132119 . . . . . **296**

Kaixun He and Haixiao Ding
Prediction of NOx Emissions in Thermal Power Plants Using a Dynamic Soft Sensor Based on Random Forest and Just-in-Time Learning Methods
Reprinted from: *Sensors* **2024**, *24*, 4442, https://doi.org/10.3390/s24144442 . . . . . **317**

Haoyu Wu, Jun Qi, Erick Purwanto, Xiaohui Zhu, Po Yang and Jianjun Chen
Multi-Scale Feature and Multi-Channel Selection toward Parkinson's Disease Diagnosis with EEG
Reprinted from: *Sensors* **2024**, *24*, 4634, https://doi.org/10.3390/s24144634 . . . . . **336**

Jiedeerbieke Madiniyeti, Qiujing Zhou, Huijun Qi, Yang Chao and Yu Zhang
Research on Automated Modeling Technology of Sluice Gate Project Based on BIM Technology
Reprinted from: *Sensors* **2024**, *24*, 5114, https://doi.org/10.3390/s24165114 . . . . . **351**

Sha Chang, Yahui Wu, Su Deng, Wubin Ma and Haohao Zhou
Task-Importance-Oriented Task Selection and Allocation Scheme for Mobile Crowdsensing
Reprinted from: *Mathematics* **2024**, *12*, 2471, https://doi.org/10.3390/math12162471 . . . . . **363**

Yan Liu, Yunbai Qin, Zhonglan Lin, Haiying Xia and Cong Wang
Detection of Scratch Defects on Metal Surfaces Based on MSDD-UNet
Reprinted from: *Electronics* **2024**, *13*, 3241, https://doi.org/10.3390/electronics13163241 . . . . . **388**

Junshuai Ni, Fang Ji, Shaoqing Lu and Weijia Feng
An Auditory Convolutional Neural Network for Underwater Acoustic Target Timbre Feature Extraction and Recognition
Reprinted from: *Remote Sensing* **2024**, *16*, 3074, https://doi.org/10.3390/ rs16163074 . . . . . **405**

Mohamed S. El-Asfoury, Mohamed Baraya, Eman El Shrief, Khaled Abdelgawad, Mahmoud Sultan and Ahmed Abass
AI-Based Prediction of Ultrasonic Vibration-Assisted Milling Performance
Reprinted from: *Sensors* **2024**, *24*, 5509, https://doi.org/10.3390/s24175509 . . . . . **421**

Xiaojun Liu, James A. Craven, Victoria Tschirhart and Stephen E. Grasby
Estimating Three-Dimensional Resistivity Distribution with Magnetotelluric Data and a Deep Learning Algorithm
Reprinted from: *Remote Sensing* **2024**, *16*, 3400, https://doi.org/10.3390/rs16183400 . . . . . **443**

Tong Zhang, Haowen Chen, Xianqun Mao, Xin Zhu and Lefei Xu
A Domain Generation Diagnosis Framework for Unseen Conditions Based on Adaptive Feature Fusion and Augmentation
Reprinted from: *Mathematics* **2024**, *12*, 2865, https://doi.org/10.3390/math12182865 . . . . . **462**

Lichun Guo, Hao Zeng, Xun Shi, Qing Xu, Jinhui Shi, Kui Bai, et al.
Semi-Supervised Interior Decoration Style Classification with Contrastive Mutual Learning
Reprinted from: *Mathematics* **2024**, *12*, 2980, https://doi.org/10.3390/math12192980 . . . . . **477**

Nouf Al-Shenaifi, Aqil M. Azmi and Manar Hosny
Advancing AI-Driven Linguistic Analysis: Developing and Annotating Comprehensive Arabic Dialect Corpora for Gulf Countries and Saudi Arabia
Reprinted from: *Mathematics* **2024**, *12*, 3120, https://doi.org/10.3390/math12193120 . . . . . **496**

Haiyan Jiang, Xiaoshuang Wang, Wei Tang, Qinghui Song, Qingjun Song and Wenchao Hao
Event Stream Denoising Method Based on Spatio-Temporal Density and Time Sequence Analysis
Reprinted from: *Sensors* **2024**, *24*, 6527, https://doi.org/10.3390/s24206527 . . . . . **514**

Musawa Yahya Almusawa and Hassan Almusawa
Exploring the Diversity of Kink Solitons in (3+1)-Dimensional Wazwaz–Benjamin–Bona–Mahony Equation
Reprinted from: *Mathematics* **2024**, *12*, 3340, https://doi.org/10.3390/math12213340 . . . . . **532**

Saleh Alshammari, Othman Abdullah Almatroud, Mohammad Alshammari, Hamzeh Zureigat and M. Mossa Al-Sawalha
Exploring Kink Solitons in the Context of Klein–Gordon Equations via the Extended Direct Algebraic Method
Reprinted from: *Mathematics* **2024**, *12*, 3433, https://doi.org/10.3390/math12213433 . . . . . **550**

Javier Ferreiro-Cabello, Francisco Javier Martinez-de-Pison, Esteban Fraile-Garcia, Alpha Pernia-Espinoza and Jose Divasón
Intelligent Sensor Software for Robust and Energy-Sustainable Decision-Making in Welding of Steel Reinforcement for Concrete
Reprinted from: *Sensors* **2025**, *25*, 28, https://doi.org/10.3390/s25010028 . . . . . **579**

Hyeonsu Lee and Dongmin Shin
Beyond Information Distortion: Imaging Variable-Length Time Series Data for Classification
Reprinted from: *Sensors* **2025**, *25*, 621, https://doi.org/10.3390/s25030621 . . . . . **601**

Ji Zhang, Guiping Liu, Zhen Wei, Shengge Li, Yeheya Zayier and Yuanfeng Cheng
Machine Learning-Based Prediction of Well Logs Guided by Rock Physics and Its Interpretation
Reprinted from: *Sensors* **2025**, *25*, 836, https://doi.org/10.3390/s25030836 . . . . . **619**

sensors

MDPI

Article

Accurate Robot Arm Attitude Estimation Based on Multi-View Images and Super-Resolution Keypoint Detection Networks

Ling Zhou, Ruilin Wang and Liyan Zhang *

College of Mechanical & Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; zhouling_ttyy@nuaa.edu.cn (L.Z.); wangrl97@foxmail.com (R.W.)
* Correspondence: zhangly@nuaa.edu.cn

Abstract: Robot arm monitoring is often required in intelligent industrial scenarios. A two-stage method for robot arm attitude estimation based on multi-view images is proposed. In the first stage, a super-resolution keypoint detection network (SRKDNet) is proposed. The SRKDNet incorporates a subpixel convolution module in the backbone neural network, which can output high-resolution heatmaps for keypoint detection without significantly increasing the computational resource consumption. Efficient virtual and real sampling and SRKDNet training methods are put forward. The SRKDNet is trained with generated virtual data and fine-tuned with real sample data. This method decreases the time and manpower consumed in collecting data in real scenarios and achieves a better generalization effect on real data. A coarse-to-fine dual-SRKDNet detection mechanism is proposed and verified. Full-view and close-up dual SRKDNets are executed to first detect the keypoints and then refine the results. The keypoint detection accuracy, PCK@0.15, for the real robot arm reaches up to 96.07%. In the second stage, an equation system, involving the camera imaging model, the robot arm kinematic model and keypoints with different confidence values, is established to solve the unknown rotation angles of the joints. The proposed confidence-based keypoint screening scheme makes full use of the information redundancy of multi-view images to ensure attitude estimation accuracy. Experiments on a real UR10 robot arm under three views demonstrate that the average estimation error of the joint angles is 0.53 degrees, which is superior to that achieved with the comparison methods.

Keywords: robot arm; attitude estimation; super-resolution keypoint detection network (SRKDNet); multi-view images

1. Introduction

In the context of intelligent manufacturing, robot arms with multiple joints play increasingly important roles in various industrial fields [1,2]. For instance, robot arms are utilized to accomplish automatic drilling, riveting and milling tasks in aerospace manufacturing; in automobile and traditional machinery manufacturing fields, robot arms can be frequently seen in automatic loading/unloading, automatic measurement and other production or assembly tasks.

In most industrial applications, a robot arm works in accordance with the pre-planned program. However, on occasions where the robot arm becomes out of control by mistake, serious collision or injury accidents may occur, especially in the work context of human–machine cooperation. Therefore, it is critical to configure monitoring means to ensure safety. On-site attitude monitoring of working robot arms is also essential for the collaborative work of multiple robot arms.

Machine vision is one of the most suitable and widely used monitoring means due to its relatively low cost, high applicability and good accuracy. To reduce the difficulty of image feature recognition and to improve monitoring accuracy and reliability, a common method in industry is to arrange cooperative visual targets on the monitored object [3,4].

However, arranging cooperative targets is usually cumbersome and time-consuming, and the targets in industrial sites tend to suffer from being stained or falling off. Studying methods for accurately estimating the attitude of robot arms without relying on cooperative visual markers presents a significant research challenge [5,6].

With their excellent ability to extract image features, deep neural networks have been widely used in the field of computer vision. They can extract natural feature information and deep semantic information from images and realize various computer vision tasks based on the rich extracted information without relying on cooperative visual markers. The base of the robot arm in most industrial scenes is fixed on the ground or a workbench. In this situation, the motion attitude of the robot arm is completely determined by the rotation angle of each joint. Therefore, monitoring the robot arm attitude is essential to determine the rotation angles of the arm joints. One approach to this is constructing an end-to-end neural network model to directly predict the attitude parameters of the robot arm through the input image of the robot arm. However, the end-to-end method requires more computing resources. In addition, it is not easy to make full use of the kinematic constraints of the robot arm and the imaging constraints from 2D to 3D space. Therefore, the attitude estimation accuracy of the end-to-end method is difficult to ensure. Another possible approach to attitude estimation is composed of two stages. First, the feature points of the robot arm are detected in the image, and then system equations are established to solve the angle of each joint. This strategy can better leverage the advantages of deep learning and 3D machine vision theory.

Keypoint detection is a major application of deep learning methods. Toshev et al. [7] directly located the image coordinates of the keypoints on the human body through convolutional neural networks to determine the human pose. Instead of outputting determined positions for the detected keypoints, the subsequent keypoint detection networks commonly output the positions in the form of heatmaps [8–11]. Newell et al. [8] proposed SHNet (Stacked Hourglass Network), which stacked several hourglass modules and detected keypoints based on multi-scale image feature information. Chen et al. [9] proposed CPNet (Cascaded Pyramid Network), which is cascaded by two convolutional neural network modules. The first module is used to detect all keypoints, and the second module corrects the poor-quality results detected by the first module to improve the final detection accuracy. Peng et al. [10] proposed PVNet (Pixel-wise Voting Network), which can obtain superior keypoint detection results when the target object is partially blocked. Sun et al. [11] proposed HRNet (High-Resolution Network), which processes feature maps under multiple-resolution branches in parallel, so that the network can maintain relatively high-resolution representation in forward propagation. These neural networks have found successful applications in tasks such as human body posture estimation, where qualitative understanding instead of quantitative accuracy is the main concern. Due to the huge demand for computing resources for network training, these neural networks have to resort to a downsampling process for front-end feature extraction, which leads to the resolution of the output heatmaps being insufficient for high-accuracy estimation.

Determining how to make the neural network output a higher-resolution heatmap without significantly increasing the consumption of computing resources is a significant problem worth investigating. Super-resolution image recovery based on deep learning has seen great progress in recent years. SRCNNet (Super-Resolution Convolutional Neural Network) [12], VDSRNet (Very Deep Super-Resolution Network) [13], EDSRNet (Enhanced Deep Super-Resolution Network) [14], etc., have been proposed. The early super-resolution reconstruction networks need to upsample a low-resolution input image to the target resolution for subsequent processing before training and prediction; therefore, the computational complexity is high. Shi et al. [15] proposed ESPCNNet (Efficient Subpixel Convolutional Neural Network) from the perspective of reducing computational complexity. This convolutional neural network deals with low-resolution feature maps in the training process and only adds a subpixel convolution layer to realize upsampling operation in the end, which effectively increases the speed of super-resolution image reconstruction. It has the potential

to improve the resolution of the output keypoint heatmaps and in turn improve the keypoint positioning accuracy by introducing the idea of super-resolution image reconstruction into the keypoint detection network.

To monitor the attitude of a robot arm, it is essential to solve the rotation angle of each joint. Based on the depth image of the robot arm, Widmaier et al. [16] used a random forest regression model to estimate the robot arm attitude. Labbe et al. [17] and Zuo et al. [18] estimated the robot arm attitude based on one single grayscale image. However, serious joint occlusion is inevitable in one single-perspective image, which makes it hard to detect some keypoints and may even lead to wrong estimation results. Moreover, the depth ambiguity problem in monocular vision may lead to multiple solutions in attitude estimation, reducing the monitoring reliability of the robot arm.

In this paper, we present a two-stage high-precision attitude estimation method for base-fixed six-joint robot arms based on multi-view images. The contributions include the following: (1) A new super-resolution keypoint detection network (SRKDNet for short) is proposed. The novelty of the SRKDNet lies in that a subpixel convolution module is incorporated in the backbone neural network HRNet [11] to learn the law of resolution recovery of the downsampled feature maps. This method can alleviate the disadvantages of low-resolution heatmaps and improve the keypoint detection accuracy without significantly increasing the computing resource consumption. (2) A coarse-to-fine detection mechanism based on dual SRKDNets is put forward. A full-view SRKDNet obtains a relatively rough keypoint detection result. Then, a close-up SRKDNet is executed to refine the results with a cropped image of the ROI determined by the results of the full-view SRKDNet. The dual-SRKDNet detection mechanism performs better than one-time detection, and the keypoint detection accuracy is drastically improved. (3) Efficient virtual-and-real sampling and neural network training methods are proposed and verified. The virtual sample data are first used to train the neural network, and then a small number of real data are applied to fine-tune the model. This method achieves accurate keypoint detection for real data without consuming a huge amount of time and manpower. (4) The constraint equations for solving the rotation angles of each joint are established; they depict the relation among the detected keypoints in the multi-view images, the camera imaging model and the kinematic model of the robot arm. A screening strategy based on the keypoint detection confidence is incorporated in the solving process and is proved to be critical for ensuring attitude estimation accuracy. Experiments demonstrate that the whole set of methods proposed in this paper can realize high-accuracy estimation of robotic arm attitude without utilizing cooperative visual markers.

The remaining contents of this paper are arranged as follows: In Section 2, we introduce the whole set of methods, including the approaches to high-precision keypoint detection (Section 2.1), automatic virtual sample generation (Section 2.2) and robot arm attitude estimation (Section 2.3). Experiments on virtual and real robot arms are reported in Section 3. We conclude the paper in Section 4.

2. Materials and Methods

2.1. High-Precision Detection of Keypoints

The first step of the two-stage attitude estimation method proposed in this paper is to detect the preset keypoints on/in the robot arm in the images. The detection accuracy directly affects the accuracy of the joint angle estimation.

The preset keypoints were selected under three basic criteria: (1) The keypoints should have distinctive features to be identified in the images. (2) The keypoints should be helpful for determining the attitude of the robot arm. (3) There are at least 2 keypoints on each rod. Taking the commonly used UR10 robot arm as the research object, we selected 20 keypoints on/in the robot arm (including the working unit attached at the end). The keypoint set was composed of the 3D center points (the red points) of each rotating joint, the 3D midpoint (the blue points) of each rod segment and some salient feature points (the green points) of the arm, as shown in Figure 1. The first two types of points are inside the structure, and the

third type of points is on the surface of the structure. The keypoints form a skeleton, which can effectively characterize the attitude of the entire robot arm.



Figure 1. The selected keypoints on/in the robot arm.

Each keypoint was directly selected on the 3D digital model of the robot arm when it was in the zero position. To generate the sample data for neural network training, either a virtual or real robot arm was controlled to move to the specified positions. The 3D coordinates of the preset keypoints at each specified position (for instance, the position as in Figure 1) could be obtained according to the kinematics of robotic arms [19], which will be detailed in Section 2.2.2.

Given any predefined keypoint position in/on the digital model, its corresponding image point can be calculated according to the camera model. In this way, we obtained a large number of training samples for the keypoint detection network. Experiments show that with the predefined keypoints, the keypoint detection network works well and the arm attitude estimation achieves high accuracy.

This section begins with a brief introduction to the backbone network HRNet [11] used in this paper. Then, we introduce the idea of image super-resolution reconstruction into keypoint detection and propose a super-resolution keypoint detection network, SRKDNet, which can alleviate the disadvantages of the low-resolution heatmaps without significantly increasing the computing resource consumption. A coarse-to-fine keypoint detection scheme based on dual SRKDNets is also presented in this section.

2.1.1. Brief Introduction to HRNet

To retain high resolution for the feature maps in the forward propagation, the high-resolution network HRNet processes feature maps at various resolutions, as shown in Figure 2. First, a preprocessing module is used to downsample the input image, which lowers the resolution of the output heatmaps as well. The main structure of HRNet can be divided into several stages, and the branches at different resolutions in each stage use the residual blocks to extract features. After each stage, a new branch is created without abandoning the original-resolution branch. The new branch is obtained by strided convolutions. The length and width of the new feature map are reduced to 1/2 of the original, but the number of channels becomes twice that of the original. In the new stage, the feature maps are created by fusing the multi-scale feature maps of each branch in the previous stage. The HRNet shown in Figure 2 has four branches with different resolutions. The final output feature map integrates the information extracted from the four branches and is used for generating the keypoint heatmap.

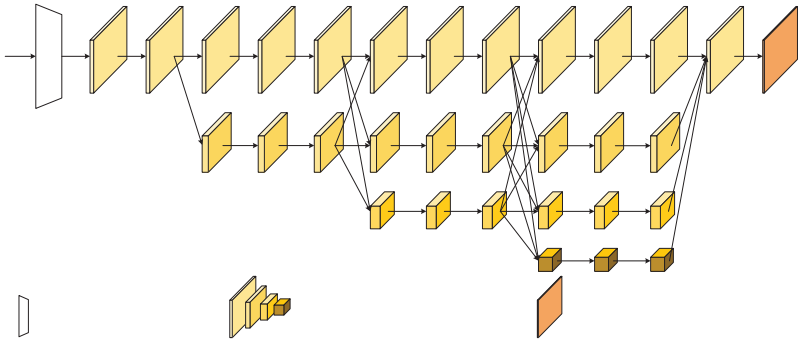


Figure 2. Network structure of HRNet.

The multi-scale fusion operation in HRNet is shown in Figure 3. The feature maps from the branches with the same resolution remain the same while those with different resolutions are converted to the same resolution first via upsampling or downsampling (strided convolutions). Then, they are aggregated to obtain the output maps. HRNet has powerful multi-scale image feature extraction capability and has been widely used in classification recognition, semantic segmentation and object detection. We take HRNet as the backbone in our super-resolution keypoint detection network (SRKDNet) which will be presented in the next subsection.

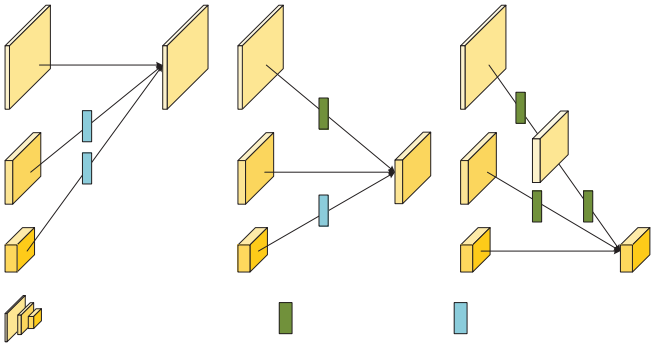


Figure 3. Multi-scale fusion method of HRNet.

2.1.2. Super-Resolution Keypoint Detection Network (SRKDNet)

If the resolution of the sample images is large, the neural networks will consume huge amounts of computing resources in the training. Therefore, HRNet and most related neural networks like SHNet [8] utilize a preprocessing convolution module to downsample the input images. Taking HRNet for instance, the original images need to be downsampled to 64×64 through the preprocessing convolution module. The so-called high-resolution feature map maintained in each subsequent layer of the convolutional neural network structure is only 64×64. If an ordinary upsampling method, e.g., neighbor interpolation, is used to restore the resolution, the downsampling operation will lead to precision loss of the keypoint positions in the heatmaps. Suppose the coordinates of a keypoint in the original image are (u, v), and the downsampling scale is a; then, the coordinates of the keypoint on the corresponding ground-truth heatmap are $(u/a, v/a)$, which have to be rounded as $([u/a], [v/a])$. Here, $[\cdot]$ represents the round operator. When the keypoint coordinates $([u/a], [v/a])$ are re-mapped back to the original resolution denoted as (u', v'), a maximum error is possibly generated:

$$\varepsilon_{\max} = \sqrt{(u - u')^2 + (v - v')^2} = \frac{\sqrt{2}}{2}a \tag{1}$$

It can be seen that the larger the downsampling scale is, the more possible accuracy loss of the keypoint position in the heatmap there will be.

Inspired by the methods of image super-resolution reconstruction, we propose a super-resolution keypoint detection network (SRKDNet). As shown in Figure 4, SRKDNet uses HRNet as the backbone network to extract multi-scale feature information in the images. The preprocessing module extracts shallow features from the input image and downsamples it. The generated low-resolution feature map is then sent to the core module of HRNet. Instead of directly sending the feature maps output by the core module to the 1×1 convolution layer for heatmap generation, SRKDNet incorporates a subpixel convolution layer [15] after the core module to learn the law of resolution recovery of the downsampled feature maps. In addition, a branch is added to the input side, and the resolution of the output feature map of the branch is the same as that of the input image (also called the original resolution). The branch is combined with the resolution-recovered feature map output by the resolution recovery module through channel fusion processing. Finally, a 1×1 convolutional layer is used for generating the heatmap with the original resolution.

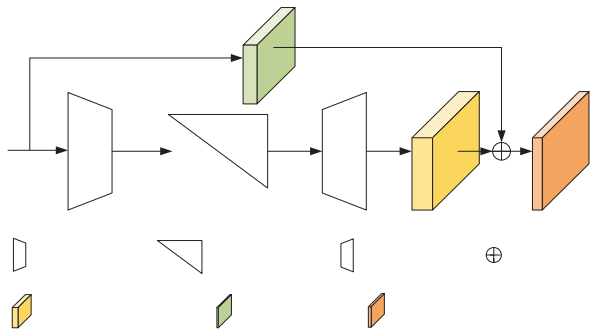


Figure 4. Structure of the proposed SRKDNet.

Adding a branch at the input side can provide additional shallow image features for the final heatmap generation. We believe that the combination of shallow and deep feature information is conducive to keypoint detection. In order to avoid the loss of image information, the branch consists of only one convolution layer and one batch normalization layer, with no activation layer involved.

The essence of subpixel convolution is to rearrange the pixels from a low-resolution image with more channels in a specific way to form a high-resolution image with fewer channels. As shown in Figure 5, pixels of different channels at the same position on the low-resolution image with size (r^2, H, W) are extracted and composed into small squares with size (r, r), which together form a high-resolution image with size $(1, rH, rW)$ Realizing the subpixel convolution requires convoluting the low-resolution feature maps first to expand the number of image channels. For instance, the low-resolution image in size of (C, H, W) needs to be expanded to size (r^2C, H, W) via convolution before it can be converted to a high-resolution image in size of (C, rH, rW). In our implementation, $r = 2$.

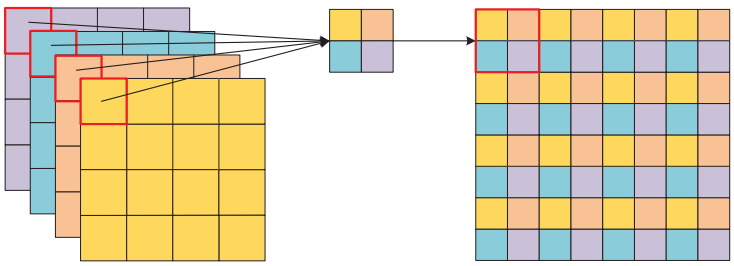


Figure 5. Subpixel convolution processing.

Since the backbone network HRNet has a strong ability to detect features in multiple scales, SRKDNet does not adopt a complex super-resolution neural network structure.

Instead, only a subpixel convolution module is applied to enable the convolutional neural network to learn the knowledge for generating high-quality and high-resolution heatmaps from low-resolution information. The subsequent experiments can prove its significant effects in improving the neural network performance.

2.1.3. Coarse-to-Fine Detection Based on Dual SRKDNets

For collecting the sample data of the robot arm, it is necessary to make the field of view of the camera completely cover the working space of the robot arm, which is much larger than the robot arm itself. Therefore, the robot arm occupies only a small region in some images, while the large background regions have no help in the keypoint detection. To further improve the detection performance, we propose a coarse-to-fine detection strategy based on dual SRKDNets. First, an SRKDNet, namely a full-view SRKDNet, is trained by using the original sample images, as shown in Figure 6a. By using the trained full-view SRKDNet, the coarse keypoint detection results are obtained (red points in Figure 6b). Based on the relatively rough detection results, the corresponding region of interest (ROI) of the robot arm in each image (blue bounding box) is determined, as shown in Figure 6b. According to the ROI, a new image is cropped from the sample image, as shown in Figure 6c. Using the cropped new images as the sample data, another SRKDNet, namely close-up SRKDNet, is trained. The two SRKDNets use the same convolutional neural network structure, the same training flow and the same setup. For an image to be detected, the trained full-view SRKDNet is first used for rough detection. The cropped image of the robot arm ROI is then put into the trained close-up SRKDNet to obtain the final keypoint detection result, shown as the blue points in Figure 6d. Our experiments have demonstrated that the detection scheme based on dual SRKDNets can drastically improve the keypoint detection performance. The details of the experiments can be found in Section 3.

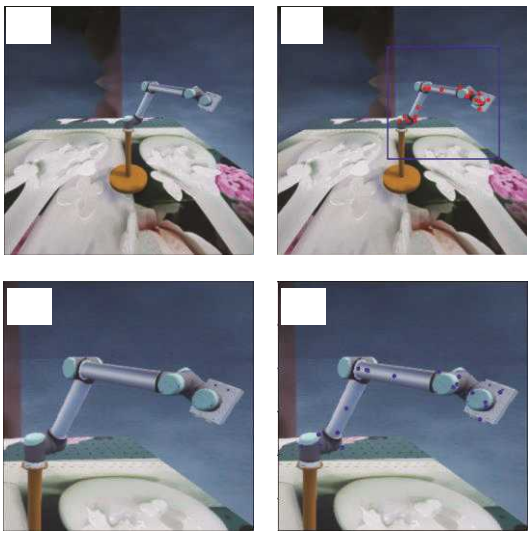


Figure 6. Coarse-to-fine keypoint detection based on dual SRKDNets. (**a**) Original sample image. (**b**) ROI of the robot arm. (**c**) Cropped image of the robot arm ROI. (**d**) Keypoint detection result.

2.2. Automatic Sample Generation Based on Virtual Platform

2.2.1. Virtual Platform Construction

The training of a neural network requires a large number of sample data. The predictive effect of the neural model is directly related to the quantity and quality of the sample data. To make the neural network fully "know" the robot arm to be detected in the image and make the trained model have a more stable performance, the sample images should be taken from various perspectives, under various backgrounds and lighting conditions. Obviously, collecting a large number of diverse sample data in real industrial scenarios will consume a lot of manpower and time.

For this paper, a virtual platform in UE4 [20] was established to simulate the working scene of the UR10 robot arm equipped with a working unit. The color, roughness, high brightness and metallicity of the appearance of the real robot arm are presented in the platform as far as possible. The base coordinate system of the UR10 robot arm is set as the world coordinate system. The robot arm in the zero position is shown in Figure 7a. A movable skeleton and a parent–child relationship between the adjacent bones are created according to the structure and kinetic characteristics of the robot arm, as shown in Figure 7b. Each bone has a head and a tail node. The head node is connected to the parent bone, and the tail node is connected to the child bone. Each bone can be rotated freely with the head node as a reference point, and the child bone connected to its tail node moves with it together. The 3D digital model of each arm segment is bound to the corresponding bone, so as to drive the articulated arm to move together with the skeleton, as shown in Figure 7c.

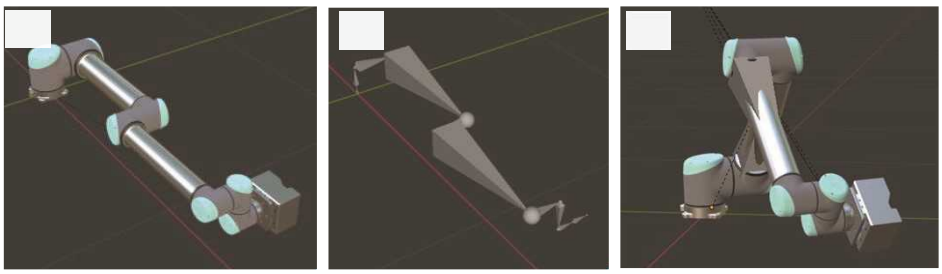


Figure 7. Binding of robot arm and skeleton. (**a**) Robot arm in zero position. (**b**) Movable skeleton of the arm. (**c**) Robot arm moving with the skeleton.

In UE4, the motion posture and speed of the robot arm can be easily set; one or more virtual cameras can be deployed in the scene; the internal and external parameters of the cameras, as well as the lighting conditions and the background of the scene, can be flexibly changed. In this way, we virtually collected a large number of sample data under various backgrounds and lighting conditions for training the SRKDNets. The background settings are randomly selected from the images in the COCO dataset [21]. A moving light source is used in the constructed virtual platform. The position and properties of the light source keep changing during the sample data collection. Rich background settings and lighting conditions in the sample data can make the neural network insensitive to the background/lighting changes and more focused on extracting the features of the robot arm itself.

After the virtual scene with the robot arm is established, the virtual cameras take virtual images of the scene to obtain the synthetic image of the robot arm. These virtual images will serve as the training samples. The attitude parameters of the robot arm, as well as the internal and external parameters of the virtual cameras corresponding to each virtual image, are recorded for image labeling, which will be detailed in the next subsection. Figure 8 shows three typical virtual sample images of the robot arm.



Figure 8. Three synthetic sample images of the robot arm against random backgrounds.

2.2.2. Automated Generation and Labeling of Virtual Samples

With the virtual platform constructed in Section 2.2.1, we generated a large number of virtual sample images using a Python program, utilizing the UnrealCV library [22] and the blueprint script in UE4. The collection process of a single sample is as follows:

1. The rotation angles of each joint of the robot arm, the pose parameters of the virtual camera, the light intensity and direction, and the background are randomly generated. They are used to automatically update the virtual sample collecting scene in UE4.
2. A virtual image of the current virtual scene is taken via the virtual camera.
3. For any keypoint P_j on/in the m-th arm segment, its relative coordinate ${}^m\boldsymbol{P}_j$ in the arm segment coordinate system C_m is converted to ${}^0\boldsymbol{P}_j$ in the robot base coordinate system C_0 according to the successive parent–child transformation relationship of the bone segments:
$$ {}^0\widetilde{\boldsymbol{P}}_j = {}^0_m\boldsymbol{T}(\theta_1, \theta_2, \cdots, \theta_m)\,{}^m\widetilde{\boldsymbol{P}}_j \tag{2} $$
where ${}^0\widetilde{\boldsymbol{P}}_j$ is the homogeneous form of ${}^0\boldsymbol{P}_j$, ${}^m\widetilde{\boldsymbol{P}}_j$ is the homogeneous form of ${}^m\boldsymbol{P}_j$ and ${}^0_m\boldsymbol{T}(\theta_1, \theta_2, \theta_m) \in \mathbb{R}^{4\times4}$ is the transformation matrix from C_m to C_0, which is determined by the rotation angles $\theta_1, \theta_2, \theta_m$ of the m joints. The coordinate values of ${}^m\boldsymbol{P}_j$ do not change with the movement of the robot arm and can be determined in advance according to the digital model of the robot arm.
4. According to the internal and external parameters of the virtual camera, the pixel coordinates of each keypoint on the virtual image are calculated by using the camera imaging model in Formula (3)
$$ s\widetilde{\boldsymbol{p}}_j = \boldsymbol{K}[\boldsymbol{R}\ \ \boldsymbol{t}]\,{}^0\widetilde{\boldsymbol{P}}_j \tag{3} $$
where $\widetilde{\boldsymbol{p}}_j$ represents the homogeneous pixel coordinate vector of keypoint P_j in the virtual image, ${}^0\widetilde{\boldsymbol{P}}_j$ is the homogeneous form of the 3D coordinates in the robot base coordinate system C_0, $\boldsymbol{R}$ and $\boldsymbol{t}$ are the rotation matrix and translation vector from C_0 to the virtual camera coordinate system C_c, $\boldsymbol{K}$ is the intrinsic parameter matrix of the camera and s is a scalar coefficient.
5. According to the pixel coordinates $\boldsymbol{p}_j$ of each keypoint P_j, the heatmap label of the current virtual image is generated. In the generated heatmap of P_j, the weight is set to the largest value at $\boldsymbol{p}_j$ and gradually decreases around $\boldsymbol{p}_j$ with Gaussian distribution. All the heatmaps of $P_j (j = 1, 2, \ldots\ldots, J)$ are concatenated as the training label of the current virtual sample.

In addition, data enhancements, including random image rotation, translation, scaling and gray changes, are also carried out on the generated sample images to improve the robustness and generalization of the model.

Using the above process, we automatically collected a large number of virtual sample data with training labels.

2.3. Attitude Estimation Based on Multi-View Images

In view of the inevitability of the keypoint occlusions and the lack of depth constraints in a single-view image, we used multi-view images combined with the proposed keypoint learning algorithm to estimate the attitude of the robot arm and verify its performance on accuracy and reliability.

2.3.1. Solving Rotation Angles of Robot Arm Joints

As stated in Section 2.2.2, for the keypoint P_j on the m-th arm segments, its 3D coordinate ${}^0\boldsymbol{P}_j$ in the base coordinate system C_0 under any robot attitude can be calculated from the 3D coordinate ${}^m\boldsymbol{P}_j$ in C_m according to Equation (2). Suppose that L cameras

are arranged to monitor the robot arm from different perspectives, then by combining Equations (2) and (3), we have the following:

$$s_j \widetilde{\boldsymbol{p}}_j^l = \boldsymbol{K}^l \left[\boldsymbol{R}^l \;\; \boldsymbol{t}^l \right] {}_m^0\boldsymbol{T}(\theta_1, \theta_2, \cdots, \theta_m)\, {}^m\widetilde{\boldsymbol{P}}_j, \;\; j = 1, 2, \cdots, J; l = 1, 2, \cdots, L \tag{4}$$

where $\overset{\sim}{\boldsymbol{p}}_j^l$ denotes the homogeneous pixel coordinates of the keypoint P_j in the l-th camera's image plane; $\boldsymbol{K}^l$, $\boldsymbol{R}^l$, $\boldsymbol{t}^l$ represent the intrinsic and extrinsic parameters of the l-th camera; and θ_m is the rotation angle of the m-th joint. For all the keypoints in the multi-view images, Formula (4) forms an equation system composed of $L \times J$ equations.

In the robot arm attitude monitoring process, the image coordinates $\boldsymbol{p}_j^l$ ($l = 1, 2, \cdots, L$) of the keypoints $P_j (j = 1, 2, \cdots, J)$ in the L images are located via the proposed dual SRKD-Nets; the camera parameters $\boldsymbol{K}^l$, $\boldsymbol{R}^l$, $\boldsymbol{t}^l$ are known in advance (In the virtual experiments, the camera parameters can be directly obtained from the settings. In the real experiments, the intrinsic parameters are determined with the popular calibration method presented in [23]. The relative geometry relationship between the UR10 robot arm and the cameras was calibrated in advance with the well-established off-line calibration method presented in [24,25].); the 3D coordinates of ${}^m\boldsymbol{P}_j$ are determined on the 3D digital model of the robot arm. Therefore, after removing the scale factor s_j, the unknowns in the equation system (4) are only the rotation angles of the joints. The LM (Levenberg–Marquardt) algorithm [26] can be used to optimize the equation system to obtain the joint angles $\theta_1, \theta_2, \cdots, \theta_m$. The initial values of $\theta_1, \theta_2, \cdots, \theta_m$ are randomly assigned within their effective ranges.

2.3.2. Keypoint Screening Based on Detection Confidence

Some keypoints, especially those on the first segment or on the flange of the robot arm, are prone to be occluded by other arm segments in certain perspectives, as shown in Figure 9. When a keypoint is blocked in the image, the detection reliability of the neural network will decline, and the error between the predicted position and the real position will be larger (see Section 3 for the experimental results). The accuracy decline of the keypoint detection will inevitably increase the attitude estimation error.



Figure 9. Three examples of robot arm with self-occlusion in real images.

However, in the case of monitoring with multi-view images, a keypoint is not likely to be occluded in all the images. Therefore, we propose a keypoint screening scheme, which is based on the detection confidence of the keypoint, to improve the attitude estimation accuracy.

As mentioned above, the value of each pixel in the heatmap output by the SRKDNet represents the probability that the image of the keypoint is located on that pixel. The pixel with the largest probability value (i.e., the detection confidence) in the heatmap will be selected as the detection result. For the L images from different perspectives, each keypoint will have L detection results, whose confidence values are different. The L detection results are sorted from high to low according to their confidence values. Then, the results with low confidence scores are discarded, and at least two results with the highest scores are kept. The screened results with high detection quality are substituted into Formula (4) so that

the attitude of the robot arm can be solved more accurately and reliably. It should be noted that with this screening scheme, the number of equations in (4) will be less than $L \times J$, but still far more than the number of unknowns. Therefore, it can ensure robust solutions.

3. Experiments

3.1. Experiments on Virtual Data

The virtual sample acquisition and labeling methods described in Section 2.2 were used to generate 11,000 labeled sample images with a resolution of 640 × 640. We randomly selected 9000 virtual samples as the training set and 1000 virtual samples as the validation set. The validation set was not included in the training and was only used to verify the effect of the model after each round of training. The other 1000 virtual samples served as the test set to demonstrate the final effect of the model after all rounds of training. All the sample images in this study were monochrome. Before the sample images were put into the convolutional neural network for training, they were reduced to the resolution of 256 × 256. All the experiments in this study were performed on a Dell workstation with an RTX2080S graphics card and 8 GB video memory.

3.1.1. Loss Function and Model Training Settings

Denote the labels in a sample batch as $\{\boldsymbol{G}_i(i = 1, 2, \cdots, N)\}$ and the heatmap output by the convolutional neural network as $\{\boldsymbol{H}_i(i = 1, 2, \cdots, N)\}$, where N represents the sample number in a batch during the training process. The number of channels, height and width of the true heatmap corresponding to a sample image are the same as those of the predicted heatmap, denoted as C, H and W, respectively. The number of channels C equals the keypoint number, i.e., 20, in this paper.

The mean square error was used as the loss function:

$$MSE = \frac{1}{N}\frac{1}{C}\frac{1}{H}\frac{1}{W}\sum_{i=0}^{N-1}\sum_{j=0}^{C-1}\sum_{h=0}^{H-1}\sum_{w=0}^{W-1}[\boldsymbol{G}_i(j,h,w) - \boldsymbol{H}_i(j,h,w)]^2 \quad (5)$$

SHNet [8], HRNet [11] and the proposed SRKDNet were trained with the generated virtual data for the comparison of the keypoint detection performance among these models. The PyTorch library was used to build and train the models. In the training of HRNet and the proposed SRKDNet, the settings in Ref. [11] were adopted: Adam optimizer was used; the initial learning rate was set to 0.001; the total training epoch was 45; the data batch size was 8; the learning rate was reduced once every 15 rounds with a reduction factor of 0.1. The weights of HRNet were obtained from the pre-trained HRNet on the ImageNet [27] dataset. For the backbone network of the proposed SRKDNet, the same initial weights and number of intermediate layers as in Ref. [11] were adopted. For SHNet, two hourglass modules were stacked, and its training followed the settings in Ref. [8]: the Rmsprop optimizer was used, the learning rate was initially set to 0.00025 and the neural network was trained from scratch using Pytorch's default weight initialization.

The resolution of the heatmaps output by both SHNet and HRNet was 64 × 64. The standard deviation of the Gaussian distribution of the weights on the corresponding ground-truth heatmap was set to 1 pixel. The resolution of the heatmaps output by SRKDNet was 256 × 256, and the standard deviation of the Gaussian distribution of the weights on the corresponding ground-truth heatmap was set to 3 pixels. The channel number of the highest-resolution branch of HRNet and SRKDNet was 32. The channel number of the feature maps in the supplementary branch containing shallow image features in SRKDNet was set to 8.

3.1.2. Experimental Results on Full-View SRKDNet

A commonly used metric, namely the percentage of correct keypoints (PCK) [18], was adopted to evaluate the keypoint prediction accuracy of each model. It is defined as shown in Formula (6):

$$PCK = \frac{1}{A}\sum_{i=1}^{A}\delta(\frac{e_i}{e_{norm}}), \delta(x) = \begin{cases} 1, & x \leq \tau \\ 0, & x > \tau \end{cases} \quad (6)$$

where A is the total number of predicted results; e_i is the pixel distance between the predicted and the ground-truth positions; e_{norm} is the standard error distance; τ is a specified threshold to adjust the ratio between the calculated error distance in the experiments and e_{norm}. If the calculated distance error e_i between the predicted and the true positions of the keypoint is less than $e_{norm} \times \tau$, δ equals 1 and the predicted position is considered correct. The keypoint prediction result of our full-view SRKDNet will be compared with that of SHNet and HRNet by using PCK as the metric. In our experiment, e_{norm} was set to 40 pixels and τ was assigned as 0.2, 0.15 or 0.1. Considering that the three neural networks output heatmaps with different resolutions, but the detected keypoint positions need to be mapped back to the original sample images to conduct the subsequent robot arm attitude estimation, we mapped the predicted coordinates of all keypoints to the original resolution 640×640 for comparison. Table 1 lists the PCK values of the three methods, where PCK@0.2, PCK@0.15 and PCK@0.1 represent the prediction accuracy with $\tau = 0.2$, $\tau = 0.15$ and $\tau = 0.1$, respectively.

Table 1. Experimental results of keypoint detection on virtual samples.

Methods	PCK@0.2	PCK@0.15	PCK@0.1
SHNet	93.17%	87.27%	66.06%
HRNet	94.91%	88.79%	68.14%
full-view SRKDNet	96.23%	94.33%	89.07%

The results in Table 1 show that the trained full-view SRKDNet completely outperforms the two comparison models SHNet and HRNet under all three threshold values. The smaller the threshold is, the more obvious the superiority of the full-view SRKDNet over the two comparison models is. The reasons for the superiority may lie in two aspects: (1) Using the heatmaps with a higher resolution (256×256) in the training labels can reduce the negative influence of the downsampling operation. (2) The predictive heatmap with the trained super-resolution layer can express the detected keypoints more accurately.

Figure 10 shows the detection results of the three keypoint detection networks SHNet, HRNet and our full-view SRKDNet for the same test image. The green dots are the real locations of the keypoints, and the blue dots are the predicted locations. The mean error refers to the average of the pixel distances between the predicted locations and the real locations of all the keypoints. The mean prediction error of full-view SRKDNet is significantly lower than that of SHNet and HRNet. We can also intuitively see that most keypoint locations predicted by the full-view SRKDNet are closer to the real location than those predicted by the comparison methods.

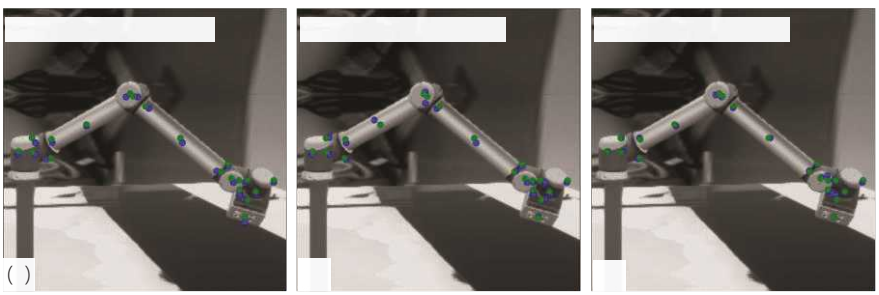


Figure 10. Comparison of keypoint detection results: (**a**) SHNet; (**b**) HRNet; (**c**) full-view SRKDNet.

3.1.3. Experiment on Occlusion Effect

A large number of experiments have shown that the detection confidence of the occluded keypoints in the image is low. In the example shown in Figure 11, keypoint No. 1 on the end working unit is completely blocked by other arm segments. It turns out that the maximum confidence of this keypoint in the heatmap is only 0.116. Meanwhile, keypoint No. 2 on the first arm segment is visible in the image, and its detection confidence is 0.774, which is much higher than that of keypoint No. 1. It can also be clearly observed from Figure 11 that the predicted position of keypoint No. 1 deviates a lot from the real position, while the predicted position of keypoint No. 2 is closely coincident with the real position. The verified negative influence of self-occlusion on keypoint detection motivated the screening scheme proposed in Section 2.3.2. Experiments on the screening scheme will be reported in Section 3.1.5.

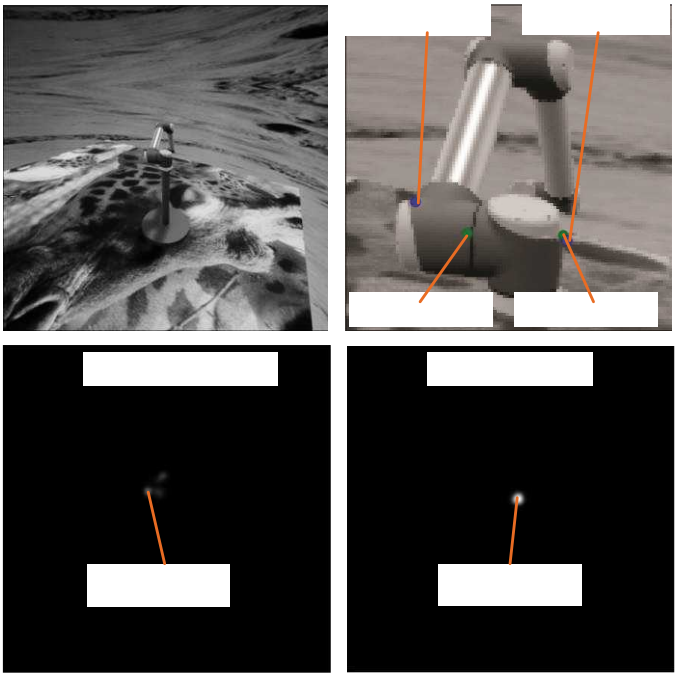


Figure 11. The influence of self-occlusion on keypoint detection.

The GPU (graphics processing unit) memory occupation of the full-view SRKDNet was also compared with that of SHNet and HRNet with the batch size set to 8 in the training. The result is shown in Table 2. The output heatmap resolution of the three convolutional neural networks is shown in parentheses.

Table 2. Comparison of GPU memory occupation (batch size = 8).

Neural Network Model	GPU Occupation
SHNet (64 × 64)	3397 MB
HRNet (64 × 64)	3031 MB
full-view SRKDNet (256 × 256)	3717 MB

Table 2 shows that HRNet occupies the least GPU memory during the training and outputs heatmaps with a resolution of only 64 × 64. The SRKDNet occupies 22.6% more GPU memory resources than HRNet. This demonstrates that the proposed full-view SRKDNet can remarkably improve the detection accuracy (see Table 1) at the expense of a mild increase in GPU occupation.

For further comparison, we canceled the downsampling operations in the preprocessing stage of HRNet so that HRNet can also output heatmaps with a 256 × 256 resolution,

which is the same as that of the full-view SRKDNet. However, the maximum batch size of HRNet can only be set to 2 in this situation, and the experimental results are shown in Table 3. These results demonstrate that to enable HRNet to output heatmaps with the same resolution as that of the full-view SRKDNet, the GPU memory resource consumption will increase sharply. When batch size = 2, the GPU occupation of HRNet exceeds 277.03% compared with our full-view SRKDNet.

Table 3. Comparison of GPU memory occupation (batch size = 2).

Neural Network Model	GPU Occupation
HRNet (256 × 256)	7371 MB
full-view SRKDNet (256 × 256)	1955 MB

3.1.4. Experimental Results on Dual SRDKNets

In this section, we present the results of keypoint detection conducted with dual SRDKNets to verify the effect of the coarse-to-fine detection strategy. Based on the detection results of the full-view SRKDNet in Section 3.1.2 for each virtual sample image, the corresponding region of interest (ROI, i.e., the region within the bounding box of the detected keypoints) of the robot arm in each image was determined. The close-up SRKDNet was then trained using the clipped local ROI images.

In the experiments on dual SRDKNets, for any virtual test samples, the trained full-view SRKDNet was first used to conduct initial keypoint detection. Then, the ROI image was input to the trained close-up SRKDNet to achieve the final detection results. The threshold τ was set to 0.15, 0.1 and 0.05, which were more stringent in order to adapt to the detection accuracy increase. The other settings were consistent with the experiments in Section 3.1.2.

The keypoint detection results of the full-view SRKDNet and dual SRDKNets are shown in Table 4. In the "full-view SRKDNet" method, only the trained full-view SRKDNet was utilized for the keypoint detection. In the "dual SRKDNet" method, the trained close-up SRKDNet was used following the full-view SRKDNet.

Table 4. Comparison of full-view SRKDNet only and dual SRKDNets.

Method	PCK@0.15	PCK@0.1	PCK@0.05
full-view SRKDNet	94.33%	89.07%	62.14%
dual SRKDNet	98.69%	97.66%	93.92%

Table 4 shows that the PCK score of dual SRKDNets is higher than that of the full-view SRKDNet, which means that the use of the close-up SRKDNet can effectively improve the keypoint detection accuracy. When a more stringent threshold is set, a more obvious improvement can be achieved. When τ = 0.05, in other words, when the distance threshold between the detected and the real keypoint positions was set to 2 pixels, the keypoint detection accuracy increased from 62.14% to 93.92%. When the threshold was assigned as 0.1, the PCK score of the close-up SRKDNet increased to 97.66%, compared to 89.07% of the full-view SRKDNet.

The above experimental results demonstrate that the proposed successive working mechanism of the dual SRKDNets is quite effective. The close-up SRKDNet can further improve the keypoint detection accuracy by a large margin.

3.1.5. Robot Arm Attitude Estimation Experiments

We used the dual SRKDNets and virtual multi-view images to verify the effect of robot arm attitude estimation. Specifically, four cameras were arranged in the UE4 virtual environment, and 1000 sets of four-perspective sample images were collected using the method in Section 2.2. The setting values of the six joint rotation angles corresponding to

each of the 1000 sets of images were recorded as the ground-truth values of the 1000 attitude estimation experiments. The full-view SRKDNet trained as described in Section 3.1.2 and the close-up SRKDNet trained as described in Section 3.1.4 were used for the coarse-to-fine keypoint detection.

The comparison experiments of single-view and multi-view attitude estimation, as well as the comparison of using and not using the confidence-based keypoint screening scheme, were conducted. The specific keypoint screening method for the four-perspective sample images adopted in the attitude estimation experiments was as follows: The detection keypoints with the top three highest confidence values were kept. If the fourth detection result had a confidence score greater than 0.9, it would also be retained; otherwise, it would be discarded.

The average error of the estimated rotation angles of the 1000 experiments of each joint is shown in Table 5. "Single view" means the attitude estimation was performed based on the information from one single-perspective image (we randomly selected the 1000 sample images collected by the second camera); "four views" means that image from all four perspectives was used in the attitude estimation; "four views + confidence screening" means the multi-view keypoint screening scheme was utilized on the basis of "four-perspective".

Table 5. Robot arm attitude estimation errors based on virtual images (unit: degree).

Method	Joint-1	Joint-2	Joint-3	Joint-4	Joint-5	Joint-6	Average
Single view	0.24	0.42	1.33	5.55	6.72	12.91	4.53
Four views	0.12	0.17	0.30	1.41	1.42	2.95	1.06
Four views + confidence screening	0.05	0.06	0.13	0.70	0.73	1.72	0.57

The above experimental results demonstrate that the average estimation error of the joint angles using images from four perspectives was reduced by 76.60% compared with that using the information from one perspective only. The confidence-based keypoint screening scheme further reduced the average error of the four-view attitude estimation by 46.23%. The compound accuracy increase reaches nearly an order of magnitude, which proves that the whole set of methods proposed in this paper is very effective.

3.2. Experiments on Real Robot Arm

3.2.1. Real Data Acquisition

The scene of a real robot arm attitude estimation experiment is shown in Figure 12, in which three cameras are distributed around a UR10 robot arm. The intrinsic parameters of the cameras and the transformation matrix of each camera coordinate system relative to the base coordinate system of the robot arm were calibrated in advance by using well-studied methods [23–25].



Figure 12. Experiment scene of real robot arm attitude estimation.

We planned 648 positions for the flange endpoint of the robot arm in its working space as the sample positions, as shown in Figure 13. Each sample position corresponded to a set of six joint angles. After the robot arm reached each sample position, the three cameras collected images synchronously and automatically recorded the current joint angles and the 3D coordinates of the center endpoint of the flange in the base coordinate system. This process was repeated until the real sample data collection was completed. A total of 1944 images were captured by the three real cameras.

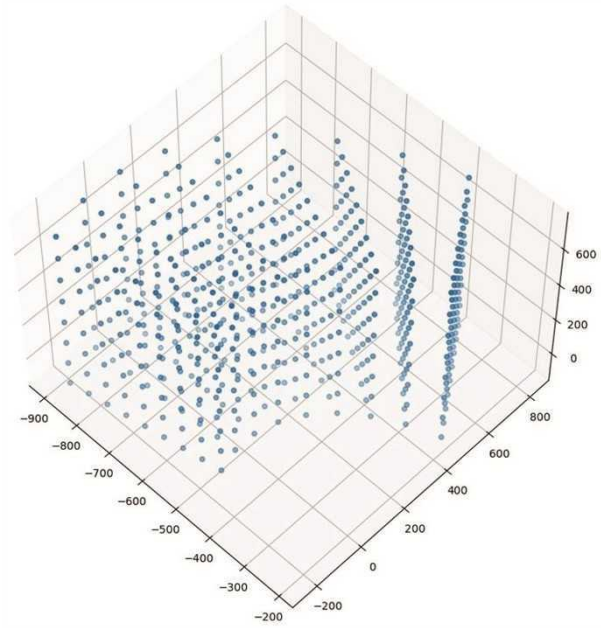


Figure 13. Sample planning of real robot arm in working space.

The resolution of the industrial camera used in the experiment was 5120 × 5120. To facilitate the training and prediction of the keypoint detection networks, and to unify the experimental standards, the resolution of the collected real images was reduced to 640 × 640, which was the same as the resolution of the virtually synthesized images. The detected keypoint positions were mapped back to the initial images for the robot arm attitude estimation.

3.2.2. Keypoint Detection Experiment on Real Robot Arm

The real UR10 robot arm is consistent with the digital model in the virtual sampling platform. Therefore, all the settings in the experiments in Section 3.1, the geometric parameters of the robot arm, the 3D coordinates of the keypoints in the arm segment coordinate system and the kinematic model of the robot arm were also applied to the real robot arm attitude estimation experiments.

When the full-view SRKDNet trained using the virtual samples as described in Section 3.1.2 was used to detect the keypoints in the real images, its detection accuracy on real images was only 34.11%, 20.86% and 7.06% when the threshold τ was set to 0.15, 0.1 and 0.05, respectively. Therefore, we considered using the real sample data to fine-tune the trained model.

From the 1944 images (648 sets of triple-view real data) obtained in Section 3.2.1, 99 real sample data (33 sets) were randomly selected as training sets. Another randomly selected 99 real sample data (33 sets) served as the validation sets. The remaining 1746 samples (582 sets) were used as the test sets to evaluate the performance of keypoint detection and attitude estimation.

The full-view SRKDNet and the close-up SRKDNet pre-trained with virtual sample data were both fine-tuned with the training sets. For comparison, we also tried the method in which the full-view SRKDNet and the close-up SRKDNet were trained not with virtual

sample data but directly with the real training sets. Since the number of real samples used for training was very small, the training epoch was set to 300. The learning rate was reduced once for every 100 epochs with a reduction coefficient of 0.1. The other settings were consistent with those in Section 3.1.1. The keypoint detection results of the 1746 real test samples with the model trained with these methods are shown in Table 6. The comparison of all the experimental results was still evaluated at a resolution of 640×640. The first row displays the result of applying only the full-view SRKDNet trained with the 99 real sample data (training sets) to detect the keypoints. The second row displays the result of applying only the full-view SRKDNet trained with our virtual datasets and fine-tuned with the 99 real sample data (training sets) to detect the keypoints. The keypoint detection results of dual SRKDNets trained with the 99 real sample data (training sets) are shown in the third row. The last row displays the result of dual SRKDNets trained with our virtual datasets and fine-tuned with the 99 real sample data (training sets). The initial weights of the backbone network of these models were obtained from HRNet pre-trained with the ImageNet dataset.

Table 6. Experimental results of keypoint detection for real robot arm.

Method	Virtual Sample Data Used in Training	PCK@0.15	PCK@0.1	PCK@0.05
Full-view SRKDNet	No	92.01%	83.38%	47.27%
Full-view SRKDNet	Yes	95.35%	88.48%	56.26%
Dual SRKDNets	No	91.68%	85.72%	66.50%
Dual SRKDNets	Yes	96.07%	92.51%	78.07%

Table 6 shows that no matter whether full-view SRKDNet only or dual SRKDNets were used for the keypoint detection, the model trained with the virtual samples generated in Section 3.1.2 and fine-tuned with 99 real sample data (training sets) demonstrates much better detection accuracy on the real test. When the full-view SRKDNet and close-up SRKDNet were trained with no virtual data but only the 99 real sample data (training sets), the keypoint detection accuracy values on the real test images were obviously lower. The results of this experiment verify the following: (1) the virtual samples generated with the proposed method in Section 2.2 have a significant positive effect on the keypoint detection of the robot arm in real scenes; (2) small amounts of real samples can efficiently re-train the model having been trained with virtual samples and achieve high generalization on real robot arms.

An example of keypoint detection in a realistic scenario is shown in Figure 14, where the first row shows the situation of using the full-view SRKDNet only, and the second row shows the situation of using the full-view SRKDNet and the close-up SRKDNet. The first column in Figure 14 shows the input real images, the second column shows the achieved heatmaps and the third column illustrates the keypoint detection results.

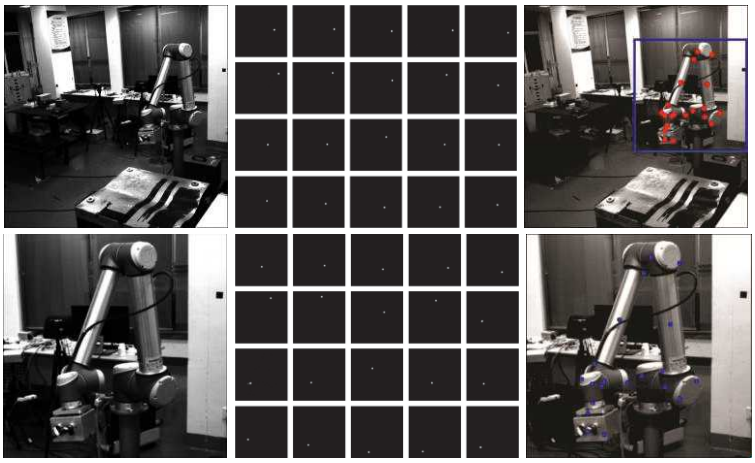


Figure 14. Keypoint detection in realistic scenario.

3.2.3. Attitude Estimation Experiment on Real Robot Arm

In this section, we report the real robot arm attitude estimation experiment, which was carried out based on the keypoint detection method with the highest detection accuracy in Section 3.2.2, i.e., the last method in Table 6. The confidence-based keypoint screening scheme was employed in the triple-view attitude estimation process. For each keypoint, the results with the first and the second highest confidence scores were retained. If the confidence score of the lowest results was larger than 0.9, it was also retained; otherwise, it was discarded. The distortion compensation was conducted according to the calibrated camera parameters to rectify the detected keypoint position.

The average estimation errors of each joint angle on the 582 real test sets collected from three perspectives are shown in Table 7. The ground-truth values of each joint angle were obtained from the control center of the robot arm when the sample data were collected. The error of each joint in the table is the average of 582 estimated results.

Table 7. Experimental results of real robot arm attitude estimation (unit: degree).

Joint No.	Joint-1	Joint-2	Joint-3	Joint-4	Joint-5	Joint-6	Average
Average error	0.15	0.10	0.15	0.55	0.78	1.47	0.53

The experimental results in Table 7 show that the whole set of methods proposed in this paper can achieve high-precision attitude estimation for the robot arm under realistic scenarios. The total average error of all of the six joints is only 0.53°, which is even slightly better than the average error of 0.57° in the attitude estimation for the virtual test samples in Table 5. Here, we briefly analyze the reasons. When sampling in the virtual environment, each joint angle and the camera poses were set at random. Therefore, a certain number of virtual samples are unrealistic, such as interference existing between the arm segments, and positions of the arm segments being too tight, which results in self-occlusion of some keypoints in all perspectives. It is hard to estimate the attitude of the robot arm with these virtual samples. However, in the real scenario, only the normal working attitudes appear in the sample data, with no extremely strange attitudes for the sake of safety. This may explain the reason why the attitude estimation accuracy using three-view information in real scenes is slightly better than that using four-view information in a virtual dataset.

The average joint angle estimation errors of a four-joint robot arm reported in Ref. [17] and Ref. [18] are 4.81 degrees and 5.49 degrees, respectively, while the average error of the estimated joint angle of the six-joint UR10 robot arm with our method is only 0.53 degrees, which is significantly lower. The reasons may lie in three aspects: (1) The SRKDNet proposed in this paper learns the heatmaps with higher resolution by adding a subpixel

convolutional layer. In addition, the combination detection scheme based on the full-view and close-up dual SRKDNets significantly improves the detection accuracy of the keypoints. (2) The existing methods only use images from one single view, while our method uses multi-view images, which can effectively alleviate the problems of self-occlusion and depth ambiguity exhibited in single-view images. Moreover, the negative influence of improper keypoint detection results can be greatly reduced by using the information redundancy of the multi-view images and the confidence-based keypoint screening scheme. (3) The existing methods not only estimate the joint angles of the robot arm but also estimate the position and attitude of the robot arm relative to the camera, so there are 10 unknowns to be estimated. Considering that the base of the robot arm is usually fixed in the industrial scenes, we determine the relative geometry relationship between the robot arm and the cameras through a well-established off-line calibration process to simplify the problem to six unknowns. The above aspects of our method together contribute to the significant accuracy improvements.

We also counted the time consumed in the two stages of keypoint detection and attitude estimation in the real scene, as shown in Table 8. The total time used for keypoint detection in the three-perspective images using the dual SRKDNets was 0.28 s. The resolution of the sample images used for the keypoint detection here was still 640×640. The time required for solving the joint angles of the real robot arm was 0.09 s.

Table 8. Time consumption of each stage (unit: second).

Stage	Triple-View Keypoint Detection	Robot Arm Attitude Estimation
Time consumption	0.28	0.09

4. Conclusions

We have proposed a set of methods for accurately estimating the robot arm attitude based on multi-view images. By incorporating a subpixel convolution layer into the backbone neural network, we put forward the SRKDNet to output high-resolution heatmaps without significantly increasing the computational resource consumption. A virtual sample generation platform and a keypoint detection mechanism based on dual SRKDNets were proposed to improve the keypoint detection accuracy. The keypoint prediction accuracy for the real robot arm is up to 96.07% for PCK@0.15 (i.e., the position deviation between the predicted and the real keypoints is within 6 pixels). An equation system, involving the camera imaging model, the robot arm kinematic model, and the keypoints detected with confidence values, was established and solved to finally obtain the rotation angles of the joints. The confidence-based keypoint screening scheme makes full use of the information redundancy of the multi-view images and is proven to be effective in ensuring attitude estimation. Plenty of experiments on virtual and real robot arm samples were conducted, and the results show that the proposed method can significantly improve the robot arm attitude estimation accuracy. The average estimation error of the joint angles of the real six-joint UR10 robot arm under three views is as low as 0.53 degrees, which is much higher than that of the comparison methods. The entire proposed method is more suitable for industrial applications with high precision requirements for robot arm attitude estimation.

In the real triple-view monitoring scenario, a total of 0.37 s was required for the keypoint detection stage and the attitude-solving stage. The keypoint detection accounted for the most time. The reason lies in that our method needs to detect keypoints in multi-view images with dual SRKDNets. Therefore, the efficiency of the proposed method is lower than that of the single-view-based method.

In this study, we only conducted experiments on one U10 robot arm. In the future, we will try to extend our method to real industrial scenes with more types of robot arms.

Author Contributions: The work described in this article is the collaborative development of all authors. Conceptualization, L.Z. (Liyan Zhang); methodology; L.Z. (Ling Zhou) and R.W.; software,

R.W.; validation, L.Z. (Ling Zhou) and R.W.; formal analysis, L.Z. (Ling Zhou); investigation, L.Z. (Ling Zhou) and R.W.; resources, L.Z. (Liyan Zhang); data curation, R.W.; writing—original draft preparation, L.Z. (Ling Zhou); writing—review and editing, R.W. and L.Z. (Liyan Zhang); visualization, R.W. and L.Z. (Ling Zhou); supervision, (Liyan Zhang); project administration, L.Z. (Liyan Zhang); funding acquisition, L.Z. (Liyan Zhang). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation of China (Grant number 52075260) and the Key Research and Development Program of Jiangsu Province, China (Grant number BE2023086).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lin, L.; Yang, Y.; Song, Y.; Nemec, B.; Ude, A.; Rytz, J.A.; Buch, A.G.; Krüger, N.; Savarimuthu, T.R. Peg-in-Hole assembly under uncertain pose estimation. In Proceedings of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014; pp. 2842–2847.
2. Smys, S.; Ranganathan, G. Robot assisted sensing, control and manufacture in automobile industry. *J. ISMAC* **2019**, *1*, 180–187.
3. Bu, L.; Chen, C.; Hu, G.; Sugirbay, A.; Sun, H.; Chen, J. Design and evaluation of a robotic apple harvester using optimized picking patterns. *Comput. Electron. Agric.* **2022**, *198*, 107092. [CrossRef]
4. Lu, G.; Li, Y.; Jin, S.; Zheng, Y.; Chen, W.; Zheng, X. A realtime motion capture framework for synchronized neural decoding. In Proceedings of the 2011 IEEE International Symposium on VR Innovation, Singapore, 19–20 March 2011. [CrossRef]
5. Verma, A.; Kofman, J.; Wu, X. Application of Markerless Image-Based Arm Tracking to Robot-Manipulator Teleoperation. In Proceedings of the 2004 First Canadian Conference on Computer and Robot Vision, 2004, Proceedings, London, ON, Canada, 17–19 May 2004; pp. 201–208.
6. Liang, C.J.; Lundeen, K.M.; McGee, W.; Menassa, C.C.; Lee, S.; Kamat, V.R. A vision-based marker-less pose estimation system for articulated construction robots. *Autom. Constr.* **2019**, *104*, 80–94. [CrossRef]
7. Toshev, A.; Szegedy, C. DeepPose: Human pose estimation via deep neural networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1653–1660.
8. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 483–499.
9. Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; Sun, J. Cascaded pyramid network for multi-person pose estimation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7103–7112.
10. Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; Bao, H. PVNet: Pixel-wise voting network for 6DoF pose estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4556–4565.
11. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5686–5696.
12. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 295–307. [CrossRef] [PubMed]
13. Kim, J.; Lee, J.; Lee, K. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
14. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 1132–1140.
15. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
16. Widmaier, F.; Kappler, D.; Schaal, S.; Bohg, J. Robot arm pose estimation by pixel-wise regression of joint angles. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 616–623.
17. Labbé, Y.; Carpentier, J.; Aubry, M.; Sivic, J. Single-view robot pose and joint angle estimation via render & compare. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1654–1663.

18. Zuo, Y.; Qiu, W.; Xie, L.; Zhong, F.; Wang, Y.; Yuille, A.L. CRAVES: Controlling robotic arm with a vision-based economic system. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4209–4218.
19. Liu, Q.; Yang, D.; Hao, W.; Wei, Y. Research on Kinematic Modeling and Analysis Methods of UR Robot. In Proceedings of the 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 14–16 December 2018; pp. 159–164.
20. Sanders, A. *An Introduction to Unreal Engine 4*; CRC Press: Boca Raton, FL, USA; Taylor & Francis Group: Abingdon, UK, 2017.
21. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
22. Qiu, W.; Zhong, F.; Zhang, Y.; Qiao, S.; Xiao, Z.; Kim, T.S.; Wang, Y. UnrealCV: Virtual worlds for computer vision. In Proceedings of the 2017 ACM, Tacoma, WA, USA, 18–20 August 2017; pp. 1221–1224.
23. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [CrossRef]
24. Strobl, K.; Hirzinger, G. Optimal hand-eye calibration. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–13 October 2006; pp. 4647–4653.
25. Park, F.; Martin, B. Robot sensor calibration: Solving AX=XB on the euclidean group. *IEEE Trans. Robot. Autom.* **1994**, *10*, 717–721. [CrossRef]
26. Levenberg, K. A method for the solution of certain problems in least squares. *Quart. Appl. Mach.* **1944**, *2*, 164–168. [CrossRef]
27. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Semi-Supervised Learning-Enhanced Fingerprint Indoor Positioning by Exploiting an Adapted Mean Teacher Model

Peng Chen [1,†], Yingzhi Liu [2,*,†], Wei Li [1], Jingyi Wang [1], Jianxiu Wang [1], Bei Yang [1] and Gang Feng [3]

[1] China Telecom Research Institute, Beijing 102209, China; chenpeng11@chinatelecom.cn (P.C.); liw40@chinatelecom.cn (W.L.); wangjy74@chinatelecom.cn (J.W.); wangjianxiu@chinatelecom.cn (J.W.); yangbei1@chinatelecom.cn (B.Y.)
[2] Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518110, China
[3] National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China; fenggang@uestc.edu.cn
* Correspondence: liuyingzhi@std.uestc.edu.cn
† These authors contributed equally to this work.

Abstract: Location awareness is crucial for numerous emerging wireless indoor applications. Deep learning algorithms have demonstrated the potential for achieving the required level of positioning accuracy in indoor environments. However, obtaining abundant labels for data-driven machine learning is costly in practical situations. As an effective solution to alleviating the insufficiency of labeled data for deep learning-based indoor positioning, deep semi-supervised learning (DSSL) can be employed to lessen the dependency on labeled data by exploiting potential patterns in unlabeled samples. In this paper, we propose an Adapted Mean Teacher (AMT) model within the DSSL paradigm for indoor fingerprint positioning by using a channel impulse response. To enhance the generalization of the trained model, we design an efficient implicit augmentation scheme for the training process in the AMT model. Furthermore, we develop a tailored residual network to efficiently extract location characteristics in the AMT framework. We conduct extensive simulation experiments for indoor scenarios with heavy non-line-of-sight conditions based on open datasets to demonstrate the effectiveness of our proposed AMT model. Numerical results indicate that the AMT model outperforms several consistency regularization methods and the pseudo-label method in terms of positioning accuracy and lower positioning latency, achieving a mean error of 90 cm when using a small number of labels.

Keywords: indoor positioning; channel impulse response; heavy non-line-of-sight; deep learning; semi-supervised learning

1. Introduction

With the emergence of mobile device location-based services, such as the Internet of Things (IoT) and Machine Type Communication (MTC), location-based services (LBS) have been attracting intensive attention recently. As one of the most popular wireless positioning technologies, the Global Navigation Satellite System (GNSS) has achieved great success in outdoor open-scene positioning. However, GNSS becomes infeasible for indoor scenarios due to the signal strength attenuation and multi-path effects [1]. Thus, it is imperative to develop efficient indoor positioning schemes to meet the requirements of numerous and booming indoor location-aware applications, such as indoor emergency rescue, smart factory asset management and tracking, mobile medical services, virtual reality games, etc.

Traditional positioning methods can be roughly categorized as geometry-based and feature-matching-based methods [2]. The geometry-based methods, such as Angle of Arrival (AOA), Angle of Departure (AOD), Time Difference of Arrival (TDOA), and Multi-Round Trip Time (Multi-RTT), rely on the measurement of positioning information and

estimation of the target location. Feature matching-based methods are mainly regarded as fingerprint recognition methods, which have also received widespread attention in positioning technology in the era of 5G and beyond [3]. Specifically, the primary approach of fingerprint recognition is based on the Received Signal Strength (RSS) or Channel State Information (CSI) [4].

Recently, the 3rd Generation Partnership Project (3GPP) emphasized the importance of the LBS in 5G networks, and in 3GPP Rel-16, it ultimately established the direction of 5G positioning enhancement [5]. Specifically, the New Radio (NR) specification includes reference signals introduced for positioning. These signals include the Positioning Reference Signal (PRS) for the downlink and the Sounding Reference Signal (SRS) for the uplink. Based on these signals, 3GPP Rel-16 has introduced a number of advanced positioning schemes suitable for 5G NR, including angle-based positioning schemes based on downlink AOD or uplink AOA, downlink TDOA, uplink TDOA, and Multi-RTT.

Although these methods can provide high positioning accuracy, they heavily depend on the availability of line-of-sight (LOS) components [6]. Unfortunately, in many indoor scenarios, such as the industrial environment, there could be many obstacles that can cause signal refraction, reflection, and diffraction, which leads to poor performance under some specific scenarios [7]. Hence, it is imperative to explore efficient techniques to improve the positioning performance in heavy non-line-of-sight (NLOS) scenarios for the 5G system and beyond.

In order to achieve higher positioning accuracy, the standardization work of 3GPP Rel-18 is introducing Carrier Phase Positioning (CPP) technology. The carrier phase information of a signal contains distance information between the signal receiver and transmitter, which can be used to accurately calculate the user's position. CPP technology has been widely applied in GNSS, enabling centimeter or even millimeter-level positioning accuracy [8]. However, the low power of the GNSS signals can be blocked and discontinuous in indoor scenarios. Due to the high power of cellular network signals and their resistance to environmental interference, carrier phase positioning based on cellular signals is not limited to outdoor environments, and because the carrier phase contains the distance between the signal receiver and transmitter, it can be used to precisely calculate targets' position [1]. Compared to satellite-based CPP, using CPP in indoor scenarios can achieve similar positioning accuracy and lower positioning latency. In the measurement of the carrier phase from the Positioning Reference Signal (PRS), the estimated value of the Channel Frequency Response (CFR) is first used to obtain the Channel Impulse Response (CIR) by an Inverse Discrete Fourier Transform (IDFT). Then, based on certain criteria, the first path of arrival is determined from the CIR, and the phase is calculated to obtain the carrier phase measurement value. Therefore, the CIR signal can be considered as the original information of the carrier phase to distinguish multipath characteristics and can potentially be used for accurate and pervasive indoor positioning [9]. In addition, CSI can be also obtained from CFR, which is the sampled version of CFR at the granularity of the subcarrier level [9].

Meanwhile, in recent years, Artificial Intelligence (AI) has experienced rapid development and widespread applications in positioning fields due to their outstanding performance [10]. 3GPP also studied AI-based positioning enhancement for indoor scenarios [11]. The AI-based solutions can potentially overcome the limitations and difficulties of traditional positioning methods, and numerical results show that deep supervised learning with CIR information can greatly improve positioning accuracy compared with traditional methods [12].

Motivated by the high performance of AI and its wide application, some research has regarded CSI as image information and finds a mapping from CSI measurements to the coordinates of the target terminals by using deep learning; these learning-based methods achieved higher positioning accuracy than traditional positioning methods [13,14]. Meanwhile, In CSI-based or CIR-based fingerprinting approaches, AI models are able to learn the knowledge of fingerprint features offline based on the dataset of labeled

fingerprints [4]. However, obtaining a large amount of labeled data is difficult and rather costly due to the need for experts' time and experience. Moreover, low-quality labeled data can adversely affect the performance of deep models. To address these issues, Deep Semi-Supervised Learning (DSSL) has been recently considered to improve learning performance by exploring potential patterns from unlabeled samples.

When carefully examining the similarity of image processing and the underlying indoor location, we can find that both image processing and CIR positioning are based on feature recognition, thereby realizing the perception and understanding of user location, and obtaining position features of User Equipments (UEs) in a specific area. They also have similar feedback mechanisms in the process of model training through generating loss values when using a neural network. These observations inspire us to exploit an adapted DSSL method to handle indoor positioning tasks. Although various methods, such as the ΠΠ Model, Temporal Ensembling [15], and Mean Teacher [16], have shown advantages for image classification, some modifications have to be made based on the data type and target format for positioning tasks. In this paper, we develop an Adapted Mean Teacher (AMT) model under the DSSL paradigm for indoor positioning using CIR fingerprints, which is inspired by the inherent similarity between image processing and indoor positioning, and the efficiency of the consistency regularization method. Additionally, the 5G Advanced has set high requirements for future positioning accuracy, aiming to achieve centimeter-level precision. Based on the scenario of a 5G new radio, we aim to apply the machine learning method to indoor positioning to meet the development needs of 5G Advanced. The main contributions of this paper can be summarized as follows:

- We mathematically present the CIR estimation for building a CIR-based fingerprint dataset according to the 5G NR standard.
- A tailored neural network based on Residual Network (ResNet) is designed to extract position features of CIR fingerprints to predict the position of users. In a supervised learning manner with abundant label data, it achieves sub-meter level accuracy with a mean error of 31 cm.
- We propose efficient implicit random augmentation methods for CIR data by borrowing the idea of data augmentation in image processing tasks. Experiments on adding augmentation methods in the training process show that our proposed method can achieve higher accuracy in both supervised and semi-supervised learning methods.
- We propose an AMT model to handle fingerprint indoor positioning tasks and possess a superior positioning performance than reference algorithms, achieving a sub-meter level accuracy.

The rest of this paper is organized as follows. Section 2 presents the related works for positioning methods and existing DSSL methods. Section 3 presents the scenario and system model. Section 4 elaborates the proposed AMT model, and in Section 5, we provide a detailed description of the CNN structure. We present simulation results as well as discussions in Section 6 and finally conclude the paper in Section 7.

2. Related Works

Positioning technology has been developing for decades. During this time, various location technologies have emerged. To summarize the previous technical work, in this section, we start by summarizing positioning techniques based on the common metrics of positioning. We then review recent research advancements in wireless positioning systems, which contain a detailed explanation of positioning technology for 5G cellular networks related to the paper's topic. Additionally, we provide an overview of AI-based indoor positioning methods and DSSL-based indoor positioning schemes.

2.1. Positioning Techniques Based on Common Measurements

Indoor positioning is a challenging problem that has been extensively investigated, resulting in the development of various technologies such as WiFi, Bluetooth, Ultra-Wideband (UWB), geomagnetism, sound/ultrasound, or Pedestrian Dead Reckoning (PDR) [8,17].

Though a number of facilities can be used for positioning, in traditional positioning schemes for both cellular and non-cellular positioning systems, some universal signal measurements are used. In this section, we introduce positioning methods based on common measurements such as RSS, AOA, and TDOA.

The RSS-based approach [18] is a simple and commonly used method for indoor positioning, which involves measuring the strength of the received signal. By using signal propagation models with the knowledge of the transmission power or power at a reference point, it is possible to estimate the absolute distance between the two devices based on the RSS value. In the device-based positioning, RSS positioning requires the use of trilateration or N-point lateration [19]. This involves using the RSS at the UE to estimate the precise distance between a UE and three or more signal sources. Subsequently, basic geometry and trigonometry are applied to determine the location of the device relative to the reference points.

The AOA-based methods [20] make use of antenna arrays on the receiver side to determine the angle that the transmitted signal arrives at the receiver. This is achieved by calculating the TDOA at each element of the antenna array. While AOA can provide an accurate estimation for short distances between the transmitter and receiver, it requires more complex hardware and precise calibration compared with RSS techniques. Additionally, the accuracy of the AOA-based positioning decreases as the distance between the transmitter and receiver increases, as even a small error in the angle calculation can result in a significant error in the actual location estimation. Furthermore, in an indoor environment with multipath effects, obtaining the LOS condition for AOA-based positioning can be challenging.

The TDOA-based methods [21] use the differences in signal propagation time measured at the receivers from different transmitters. To accurately determine the location of the receiver, the TDOAs from at least three transmitters are required. This allows for the calculation of the receiver's position as the intersection of three or more hyperboloids [21]. Solving the system of hyperbola equations can be achieved through methods such as linear regression or by linearizing the equation using Taylor-series expansion.

However, all of the aforementioned existing methods heavily depend on LOS scenarios, and in other scenarios, such as urban streets and indoor scenarios, the complex signal propagation paths will lead to unreliable positioning results.

2.2. Wireless Positioning Systems

Wireless positioning technology can be categorized based on the scope of service, including positioning systems for Wireless Wide Area Network (WWAN) and Wireless Local Area Network (WLAN)/Personal Area Network (PAN). The WLAN/PAN includes WiFi positioning systems, Bluetooth positioning systems, and UWB positioning systems, while the WWAN includes GNSS and cellular network positioning systems.

WiFi-based positioning technologies mainly consist of four types, including positioning methods based on RSS, fingerprinting, AOA, and TOA. Different from traditional RSS-based WiFi positioning systems, the use of deep learning methods has been widely applied to explore the numerical features of signals in RSSI positioning. Dai et al. [22] used a multi-layer neural network (MLNN) to provide localization services in RSS-based indoor localization, which combined the RSS signal-transforming section, raw data-denoising section, and node-locating section to form a deep architecture. By using the deep architecture, the predicted locations of UE can be attained without using a radio pathloss model or comparing with a radio map. Hoang et al. [23] emphasized the superiority of RNN in dealing with location nonlinearly because the mapping from RSS to UE's location is nonlinear. In this work, the authors provided a complete study of several RNN architectures for WiFi RSS fingerprint positioning. Research on WiFi fingerprint positioning typically utilizes the CSI or RSS signals obtained from WiFi signals [24]. The CSI-based method provides more detailed signal propagation characteristics, resulting in better positioning accuracy compared to RSS [25]. However, the acquisition of CSI requires the cooperation

of WiFi access points (APs), which is limited by the practical deployment of devices. To overcome this obstacle, Gao et al. [26] proposed a CSI fingerprinting-based positioning approach named CRISLoc, which obtains the packets in the air passively, while a joint clustering and outlier detection method is used to find altered APs. By applying CRISLoc, the accuracy of CSI fingerprinting-based localization can reach a sub-meter level. The RSS-based WiFi fingerprint positioning technology is often heavily influenced by noisy environments; recent studies have begun to utilize advanced deep-learning models to address these issues. Chen et al. [27] proposed an LF-DLSTM framework to alleviate the noise effect and attain stable features from the raw noisy RSS data. Additionally, in traditional AOA WiFi positioning systems, a limited number of antennas in WiFi devices can result in limited AOA resolution. In order to achieve more accurate and robust positioning, recent research has focused on developing new methods. Yang et al. [28] worked out the relationships among different AoAs of different APs, and proposed a novel co-localization method between multiple APs to achieve a real-time and accurate localization system. For TOF-based WiFi positioning, the positioning performance of the TOF-based WiFi system is largely limited by the WiFi channel bandwidth because of the low resolution of TOF, and recent research has utilized multipath to increase time resolution [29]. While the use of WIFI technology for positioning facilities can achieve centimeter-level accuracy in many studies, the coverage range is limited to a 10 m level, which results in extremely high deployment costs when attempting to cover large areas.

The Bluetooth positioning system is a common short-distance wireless communication technology primarily used for PAN. In the latest release of Bluetooth 5.1, the version has added measurements of AOA and AOD, integrating the results with RSS to provide sub-meter positioning accuracy [30]. However, Bluetooth positioning faces serious multipath interference issues. The accuracy of positioning is difficult to further improve, and there are limitations on coverage range, making it challenging to deploy over large areas [31].

UWB positioning technology is characterized by high positioning accuracy, high rating, and strong resistance to multipath interference [32]. Current recent research has mainly focused on how to reduce the impact of non-line-of-sight (NLOS) paths in high NLOS scenarios when applying UWB positioning to decrease positioning errors. Poulose et al. [33] applied LSTM networks in UWB localization in indoor scenarios to migrate the negative effects from both NLOS conditions and TOA errors. Compared to the conventional method, it can reduce the mean position error to 7 cm. Although UWB has a high positioning accuracy, the high cost of base stations and tags for UWB positioning makes it not a universally applicable positioning solution.

For positioning systems in WWAN, Assist-Global Positioning System (A-GPS) technology is widely used in the location services of smartphones. It leverages cellular mobile communication networks to broadcast GNSS information and its auxiliary data, thereby assisting UEs in shortening the satellite's initial search time and improving location accuracy during satellite navigation [34]. The GNSS signal can be easily blocked, and recent research has attempted to enhance GPS using the UWB systems. Gao et al. [35] proposed an RCP scheme that evaluates the positioning performance by generating a dataset in real urban scenarios. Experimental results show that this scheme can robustly resist adverse effects on positioning performance.

Positioning technologies in cellular networks have evolved from 2G to 5G, and now to 5G NR and 5G-Advanced. In 5G NR and 5G-Advanced, the requirement for positioning accuracy has reached to centimeter level, leading to a significant focus on research based on CSI and CPP. Meanwhile, some studies have emphasized machine learning and fingerprint recognition. Zhang et al. [36] developed a novel Attention-Aided Residual Convolutional Neural Network (AAresCNN) for CSI-based indoor positioning, achieving a state-of-the-art performance on public datasets. Ruan et al. [37] proposed a novel positioning system, iPOS, using commercial 5G-NR CSI fingerprints for indoor positioning, incorporating CSI pre-processing and feature reconstruction modules. In the current development of 5G-Advanced, Tedeschini et al. [38] utilized CIR to extract position-related features and

enhance positioning accuracy through cooperative deep learning, combined with NLOS recognition. Unlike other positioning systems, cellular network positioning does not require additional infrastructures and can achieve centimeter-level accuracy, which significantly reduces positioning costs. The use of CSI and CPP as measurement values, along with machine learning positioning strategies, demonstrates advancements in both 3GPP standards and recent research.

2.3. DSSL Methods for Indoor Positioning

In recent years, deep learning algorithms have shown great potential in solving complex positioning problems, and DSSL methods have been emerging as a promising approach to deal with the challenges of limited labeled data in positioning problems by leveraging both labeled and unlabeled data to train deep learning models.

The branches of DSSL mainly include pseudo-label methods [39], deep generative methods [40], graph-based methods [41], consistency regularize methods [15,16] and hybrid methods [42]. Currently, research on DSSL mainly focuses on the image classification task.

For DSSL methods applied in image classification tasks, each branch has advanced algorithms capable of achieving high accuracy. Regarding the pseudo-labeling method, using the high-confidence model's predictions as pseudo labels for unlabeled samples is a common approach known as self-training. Ref. [39] proposed a simple and efficient training framework for neural networks. The model is first trained in a usual supervised manner; this trained model is used to attain predictions from unlabeled data. The crossentropy loss is used in the process of obtaining predictions from unlabeled samples, and when we obtain soft labels, the model's highest confidence predictions are viewed as pseudo labels.

Consistency regularization is a technique that typically involves using a single model to make multiple predictions with different input noise or model parameters each time. It aims to obtain a similar prediction result under different noisy inputs and parameters, thereby improving the generalization of the model. From certain perspectives, using consistency regularization can also be observed as generating pseudo-labels. However, it focuses on obtaining accurate labels by regularizing the distance between outputs and supervised training, which is fundamentally different from the pseudo-labeling method. Ref. [15] proposed two training frameworks, named Π Model and Temporal Ensembling, respectively. During each epoch of training with the Π Model, the same batch of unlabeled samples is processed by the same model twice after adding random perturbations. Some inconsistencies in the two predictions will exist because of the different perturbations, so the Π Model uses a consistency loss function to minimize the disparity between the two predicted outputs. Temporal Ensembling makes some improvements to the Π Model. Due to the fact that the Π Model requires two rounds of inference at each step, it slows down the inference speed. To overcome this defect, the Temporal Ensembling model only needs one round of inference, while one prediction is obtained by calculating the moving average of a historical output over a certain period of time. Another prediction is generated by the current output. By combining the loss items of two predictions as a loss function, the inconsistency in the two predictions decreases.

Mean Teacher [16] is an improved method for Temporal Ensembling, and it consists of a teacher model and a student model. The student model resembles the Π Model, while the teacher model shares the same structure as the student model but incorporates Exponential Moving Average (EMA) of the student weights. This allows Mean Teacher to enforce a consistency constraint between the predictions of the student and teacher models. Results from [16] indicate that Mean Teacher performs superiorly in test accuracy compared to Temporal Ensembling, while it also allows for training with fewer labels.

DSSL-based indoor positioning is often considered as a regression problem rather than a classification problem, and previous research on DSSL for indoor positioning mainly focuses on pseudo-label methods [43], deep generative methods [44], and graph-based methods [45]. The idea of the pseudo-label method in [43] is to pretrain the initial model with labeled data and use the trained model to predict unlabeled data, treating the predic-

tions as pseudo-labels. The advantage of this method is its simplicity and strong operability, which is needless to deploy additional models. Whereas in actual positioning scenarios, the target distributions of labeled data and unlabeled data are often inconsistent, and using the same model parameters directly on different distributed data will reduce the positioning accuracy. In [44], the author discussed a semi-supervised indoor positioning scenario based on the Generative Adversarial Network (GAN) by using CSI data, which consists of a generator and a discriminator and aims to generate new CSI that similar to labeled data. Results of using GAN for semi-supervised positioning have confirmed its effectiveness under a few labeled input. However, using explicit data augmentation methods to improve the performance of positioning may increase the computing power burden of the device and consume a large amount of memory when facing the massive data demand. Moreover, when the labeled data are highly resembled, overfitting can be caused by a deep generative method. Moreover, the consistency regularization method as an effective DSSL method achieves a good balance between accuracy and memory occupation. In [46], the ladder network, the first attempt at consistency regularization, was used for indoor positioning using CSI, which is inspired by a deep denoising AutoEncoder. It predicts whether each input has noise, using denoising functions and the unsupervised denoising square to generate consistency loss, and aligns with the supervised learning loss to obtain more accurate user coordinates.

The advanced consistency regularization methods after the Ladder Network, such as the Π Model, Temporal Ensembling, and Mean Teacher, have been proposed to improve accuracy without requiring additional data by regulating the consistency loss between noisy inputs to reduce overfitting and enhance the generalization of neural networks. However, these methods have not been really well used yet, and how to efficiently apply them to the positioning field still remains outstanding.

3. Scenario and System Model

3.1. Scenario Description

We consider the positioning in a general indoor scenario, such as a factory, as shown in Figure 1. A number of Base Stations (BSs) are deployed on the ceiling of the factory with a certain area, and each BS consists of one microcell with an omni-directional antenna towards the ground. The dense clutters are randomly distributed, and heavy NLOS propagations exist in such complex scenarios. The clutter in the scenario can be considered as machinery, assembly lines, and storage shelves. UEs are randomly distributed in the factory while receiving the reference signal, e.g., PRS sent by BSs and process channel estimation for deriving CIR.

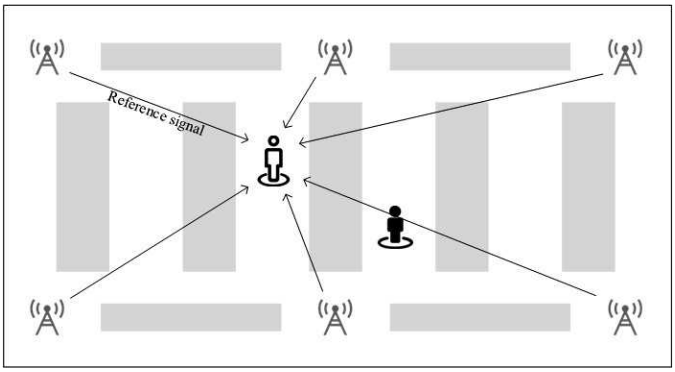


Figure 1. Schematic diagram of indoor positioning scenario.

3.2. Channel Model

We assume a massive multiple-input multiple-output (MIMO) system for presenting CIR, where each BS is equipped with N_t transmission antennas (Tx), and each UE has N_r receiving antennas (Rx). In a different environment, the mobile radio propagation link between BS and UE could be LOS or NLOS. To establish CIR, we use 3D channel models to

express the channel coefficients between each Tx and Rx according to 3GPP TR 38.901 [47]. For each link, let there be Z clusters, each with M rays. We use $m \in \{1, 2, \ldots, M\}$ to denote the index of a specific ray. For the NLOS case, when cluster ζ belongs to the $Z-2$ weakest clusters represented as $\zeta \in \{3, 4, \ldots, Z\}$, the power of different rays in each cluster is equal. The channel coefficient of ray m in cluster ζ for the receiver and transmitter antenna element pair u, s can be expressed as

$$\begin{aligned} H_{u,s,\zeta,m}^{NLOS}(t) = \sqrt{\frac{P_\zeta}{M}} \begin{bmatrix} F_{rx,u,\theta} \\ F_{rx,u,\phi} \end{bmatrix}^T \begin{bmatrix} e^{j\Phi_{\zeta,m}^{\theta\theta}} & \sqrt{\kappa_{\zeta,m}^{-1}} e^{j\Phi_{\zeta,m}^{\theta\phi}} \\ \sqrt{\kappa_{\zeta,m}^{-1}} e^{j\Phi_{\zeta,m}^{\phi\theta}} & e^{j\Phi_{\zeta,m}^{\phi\phi}} \end{bmatrix} \\ \begin{bmatrix} F_{tx,s,\theta} \\ F_{tx,s,\phi} \end{bmatrix} \exp\left(\frac{j2\pi(\hat{r}_{rx,\zeta,m}^T \bar{d}_{rx,u} + \hat{r}_{tx,\zeta,m}^T \bar{d}_{tx,s} + \hat{r}_{rx,\zeta,m}^T \bar{v} t)}{\lambda_0}\right), \end{aligned} \tag{1}$$

where P_ζ stands for the power of the cluster, $F_{rx,u,\theta}$, and $F_{rx,u,\phi}$ is the field pattern of u in the direction of the spherical basis vectors, θ and ϕ, respectively. $F_{tx,s,\theta}$ and $F_{tx,s,\phi}$ are the field patterns of s in the direction of θ and ϕ, respectively. $\hat{r}_{tx,\zeta,m}^T$ is the spherical unit vector with an azimuth arrival angle and elevation arrival angle, and $\hat{r}_{rx,\zeta,m}^T$ is the spherical unit vector with an azimuth departure angle and elevation departure angle. ($\bar{d}_{rx,u}$, $\bar{d}_{tx,s}$) is the location vector of u and s, $\kappa_{\zeta,m}$ is the cross-polarization power ratio in a linear scale, and λ_0 is the wavelength of the carrier frequency. Φ represents the random initial phase for different polarization combinations, and $\bar{v}$ is the velocity vector of the UE.

For the LOS case, the channel coefficient is given by:

$$\begin{aligned} H_{u,s,1}^{LoS}(t) = \begin{bmatrix} F_{rx,u,\theta} \\ F_{rx,u,\phi} \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} F_{tx,s,\theta} \\ F_{tx,s,\phi} \end{bmatrix} \exp\left(-j2\pi \frac{d_{3D}}{\lambda_0}\right) \\ \exp\left(\frac{-j2\pi(\hat{r}_{rx,LoS}^T \bar{d}_{rx,u} + \hat{r}_{tx,LoS}^T \bar{d}_{tx,s} + \hat{r}_{rx,LoS}^T \bar{v} t)}{\lambda_0}\right), \end{aligned} \tag{2}$$

where d_{3D} represents the 3D distance between Tx and Rx.

3.3. Estimation of Channel Impulse Response

In order to better capture the characteristics of the natural channel environment, the two strongest clusters $\zeta \in \{1, 2\}$ are spread to three different sub-clusters with fixed delay offsets. The M rays within a cluster are mapped to sub-clusters $\xi \in \{1, 2, 3\}$, and are divided into three groups R_ξ, each with a power R_ξ / M and delay offset $\tau_{\zeta,\xi} - \tau_\zeta$ for a NLOS channel. Then, the CIR from s to u in the NLOS case can be expressed as

$$H_{u,s}^{NLoS}(\tau, t) = \sum_{\zeta=1}^{2} \sum_{\xi=1}^{3} \sum_{m \in R_\xi} H_{u,s,\zeta,m}^{NLoS}(t) \delta(\tau - \tau_{\zeta,\xi}) + \sum_{\zeta=3}^{Z} H_{u,s,\zeta}^{NLoS}(t) \delta(\tau - \tau_\zeta), \tag{3}$$

where $H_{u,s,\zeta}^{NLoS}(t)$ represents the channel coefficient of cluster ζ for the antenna pair u, s, which is formulated as $H_{u,s,\zeta}^{NLoS}(t) = \sum_{m=1}^{M} H_{u,s,\zeta,m}^{NLoS}(t)$. In the LOS case, by adding the LOS channel coefficient to the CIR in the NLOS case and adjusting the scaling factor γ, we obtain the CIR, as follows:

$$H_{u,s}^{LoS}(\tau, t) = \sqrt{\frac{1}{\gamma + 1}} H_{u,s}^{NLoS}(\tau, t) + \sqrt{\frac{\gamma}{\gamma + 1}} H_{u,s,1}^{LoS}(t) \delta(\tau - \tau_1), \tag{4}$$

where $H_{u,s,1}^{LoS}(t)$ represents the LOS channel coefficient with the strongest power transmission when cluster $\zeta = 1$, and τ_1 symbolizes the minimum delay of arrival. Note that positioning mainly depends on the CIR in the NLOS case because a heavy NLOS environment exists in the scenario.

3.4. Problem Formulation

In the indoor positioning scenario, we represent the UE set as $X = \{x_n\}_{n=1,\ldots,N}$. The real global coordinate of UE n in the indoor area is denoted as $P^i_{true} = [P^n_h, P^n_v]$, where P^n_h, P^n_v represent the horizontal coordinate and vertical coordinate respectively. Assuming that the coordinate of each UE predicted by the neural network $f(\cdot)$ is valid, the problem can be restated as finding a way to utilize CIR information, the neural network $f(\cdot)$, and the true position P_{true} of each UE to approximate the global coordinate.

Let the predicted coordinate for UE n with the position error be denoted as $f(\theta, x_n)$, where θ represents the weights of the neural network. The position error is defined as $|f(\theta, x_n) - P^n_{true}|$. Our optimization objective for a set of N training samples is to minimize the Mean Squared Error (MSE) value between the predictions and true labels, which can be expressed as:

$$min \frac{1}{N} \sum_{n=1}^{N} (f(\theta, x_n) - P^n_{true})^2. \tag{5}$$

4. Semi-Supervised Learning Based on Mean Teacher Model

The concept of consistency regularization is that even if the input is perturbed, the network can still generate an output consistent with the output before perturbating and punishing inconsistent items. Specifically, consistency is based on the comparison of output space distributions, which is referred to as an approximate result or an output vector with a small distance from distribution. Consistency regularization is mainly applied to the teacher–student structure, with a consistency constraint defined as:

$$\mathbb{E}_{x \in X} \mathcal{D}(f_s(\theta_s, \eta_s, x_s), f_t(\theta_t, \eta_t, x_t)), \tag{6}$$

where $f_s(\theta_s, \eta_s, x_s)$ is the student's prediction of input x_s, $f_t(\theta_t, \eta_t, x_t)$ is the teacher's prediction of input x_t. $\mathcal{D}(\cdot)$ is the distance function between two vectors. Different consistency regularization methods differ in the way they generate consistency constraints. For example, the Π Model [15] generates consistency a constraintbetween two predictions of the same model by adding different noises to inputs, and the Temporal Ensembling [15] generates constraint between the training prediction of the current epoch and EMA prediction from the last epoch. As for the Mean Teacher, it averages model weights instead of predictions. Specifically, the teacher model uses the EMA weights of the student model and then generates a constraint between the teacher model's prediction and the student model's prediction. The consistency constraint of the Mean Teacher can be defined as

$$\mathbb{E}_{x, \eta_s, \eta_t} \mathcal{D}(f_s(\theta, \eta_s, x), f_t(EMA(\theta), \eta_t, x)), \tag{7}$$

where η_s, η_t represent different perturbations for input.

There are several techniques to improve the performance of the consistency regularization method. One strategy is to carefully select input perturbations instead of adding additive or multiplicative noises. Another one is to carefully consider the teacher model instead of copying the student model [16].

4.1. AMT for Indoor Positioning

The AMT ensembles the teacher and student model, aiming to train a better teacher model from the student model without additional training. In this paper, the teacher and student models use the same network structure, which can be considered as a self-ensemble method. The framework of the proposed AMT is shown in Figure 2.

The labeled data are denoted as $X_L = \{(x_l, y_l)\}^L_{l=1}$, where L represents the total number of labeled samples. The unlabeled data are expressed as $X_U = \{(x_u)\}^N_{u=L}$, where U represents the total number of unlabeled samples. The total dataset is presented as $X = X_L \cup X_U$, and $X = \{x_n\}^N_{n=1}$. The random signal perturbations (data augmentation) for the teacher and student models are denoted as η_t and η_s, respectively, and the weights

of the two neural networks are denoted as θ_t and θ_s, respectively. Then, the predictions of the teacher and student models are expressed as $f(\theta_t, \eta_t, x)$ and $f(\theta_s, \eta_s, x)$, respectively, where they use the identical input x with a fixed proportion of labeled and unlabeled data, and both teacher and student models use the same network structure $f(\cdot)$. We first input two CIR data streams, and use different random augmentations for each stream, and then predict UEs' position using the two models. During training steps, the two models interact, with the teacher model θ_t^k using the EMA weights of the student model. In more detail, at the end of the kth step, the weights of the teacher model θ_t^k are updated using the EMA weights of the student model, and the weight update function of the teacher model is given by

$$\theta_t^k = \alpha\theta_t^{k-1} + (1-\alpha)\theta_s^k, \tag{8}$$

where α represents the smoothing coefficient hyper-parameter.

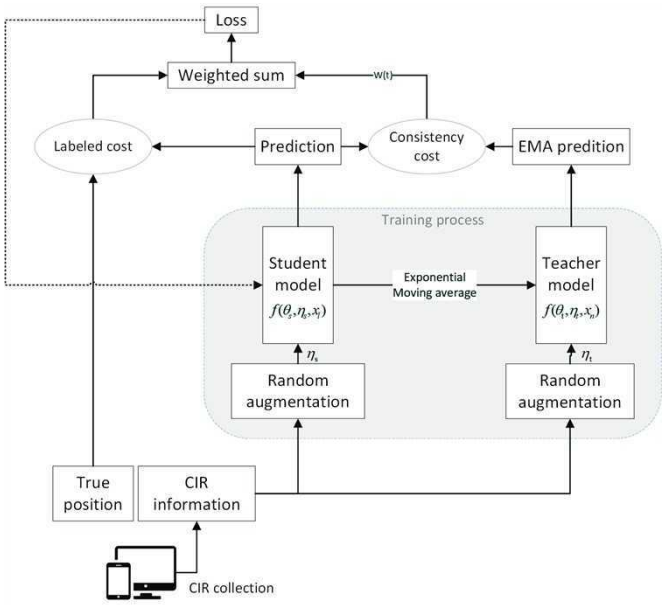


Figure 2. The framework of the proposed Adapted Mean Teacher(AMT).

Instead of regulating the consistency loss of the image classification task in the original Mean Teacher method, we measure the consistency of the users' predicted coordinates. Therefore, we set the distance function $\mathcal{D}(\cdot)$ as the smooth L1 loss between the batch outputs of data streams instead of the cross-entropy loss. The advantage of using the smooth L1 loss is that the loss can be updated more smoothly, and it is the combination of the L1 and L2 loss with the benefits of both approaches. The specific formula can be expressed as

$$SmoothL_1(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}. \tag{9}$$

Using the distance Function (9), we define the loss function for the student model, updating in a minibatch as the weighted sum of the labeled loss of the student model and the consistency loss, which is namely the distance between student and teacher model's prediction. The loss function is

$$\begin{aligned} Loss_s = &\sum_{l=1}^{L} SmoothL_1(f(\theta_s, \eta_s, x_l) - y_l) + \\ &\epsilon_t w(\frac{T}{T_{max}}) \sum_{n=1}^{N} SmoothL_1(f(\theta_t, \eta_t, x_n) - f(\theta_s, \eta_s, x_n)), \end{aligned} \tag{10}$$

where $T \in [0, T_{max}]$ represents the current training epoch, T_{max} is a coefficient that represents the maximum ramp-up length, T/T_{max} linearly increases from 0 to 1, and $w(T/T_{max})$

is the unsupervised weight ramp-up function that controls the weight of the unsupervised loss, which increases linearly to 1 over a certain number of epochs. The mild increase in training is important to assist the model in adapting to the increased training interference caused by the unlabeled data and prevent any degradation in performance. In this paper, we use a Gaussian ramp-up function with $w(T/T_{max}) = e^{-5(1-T/T_{max})^2}$. Additionally, ϵ_t is the constant that controls the maximum loss for unsupervised training and is calculated as $\epsilon_t = w_{max} \times (L/N)$, where w_{max} is a coefficient that represents a maximum weight value for unsupervised training.

At each training step, the student model learns from the teacher model by minimizing the $Loss_s$. Through this approach, we can achieve consistency in regularization.

4.2. Data Augmentation

To help models learn abstract patterns in data without being affected by minor changes, the concept of implicit data augmentation has been proposed. The model should tend to provide consistent output for similar data points. In classification tasks, the consistent output refers to the same classification, while in regression tasks, consistent output refers to output vectors that are close in distance. To achieve this goal, minor changes are typically implemented by adding noise or data perturbation. Many regularization techniques rely on this concept, such as the dropout used in neural network models.

In image classification tasks, consistency regularization methods often add random noise to the data in the data augmentation process. Some techniques, such as flipping, resizing, and random cropping can be used to increase the variety of images. The original Mean Teacher method used random translations and horizontal flips as part of its data augmentation strategy [16]. The rationale behind these approaches is that the model's softmax output usually cannot provide accurate predictions beyond the training data. To alleviate this problem, noise can be added to the model during the inference time to generate more accurate predictions. This method is used in the Pseudo-Ensemble Agreement [48] and has demonstrated excellent performance. Thus, a teacher model injected with noise can be inferred to generate more precise targets than that not injected with noise. Therefore, implicit data augmentation, namely data perturbation, aims to provide accurate predictions by generating new predictions beyond labeled data and adding randomness to prevent overfitting.

4.2.1. Implicit Data Augmentation for CIR

General positioning methods map geometric information to user positions using measurement quantities such as power, time, and angle, then estimate user position through geometric estimation methods. Power, time, and angle features are common physical measurements, each with varying accessibility, complexity, and accuracy. As shown in the estimation of Formulas (2) and (3) for CIR, the three types of information can be well reflected in CIR. Therefore, we infer that power, time, and angle should be extracted as the main useful positioning features as AI positioning methods for using CIR. However, precise angle-based positioning typically relies on the angle difference between multiple antennas on the same device in MIMO communication. In our settings, the number of sampled antennas is insufficient, so we will mainly focus on the power and time of arrival features in AI positioning for using CIR. Inspired by the concept of data augmentation in image classification, we perturb CIR input by adding random noise to critical positioning features, namely the power and time of arrival.

4.2.2. Random Amplitude Scaling

As RSS-based fingerprint positioning systems are commonly used, their fundamental limitation is their inability to capture multipath effects [9]. To fully characterize each path, the wireless communication propagation channel is modeled as a time-linear filter called CIR. CIR is similar to the RSS sequence, but it has a finer frequency resolution and equally higher time resolution to distinguish multipath components. Therefore, we can reasonably

infer that CIR has a high power feature for effective positioning, and augmentation can be effectively performed by perturbing the power characteristics of CIR.

The received power measured at a fixed frequency is proportional to the amplitude of the channel frequency response (CFR) [9]. Similarly, in CIR, we infer that the amplitude information is highly relevant to the received power, and different amplitude represents different LOS/NLOS environment distributions. Thus, we propose a random amplitude scaling in training steps to perturb partial LOS/NLOS distribution in the indoor environment, aiming to provide accurate predictions outside the training data. The amplitude of the CIR received by antenna u of UE n from transmitter antenna s can be expressed as:

$$Amp^n = \sqrt{Re\{H^n_{u,s}(\tau,t)\}^2 + Im\{H^n_{u,s}(\tau,t)\}^2}. \tag{11}$$

The scaling size is represented as a positive random number μ, where $\mu \sim U(0,a)$, with a being the maximum scaling size. Then, the random scaling amplitude can be expressed as $A^n_{scale} = \mu Amp^n$. Note that the same scaling on the time series of CIR should be performed to ensure that the shape of the amplitude remains unchanged and to avoid destroying effective positioning features.

4.2.3. Random Temporal Shifting

The time feature is a conventional positioning physical measurement. It can obtain a highly accurate position under LOS conditions. In the positioning situations, there are two conventional time-of-arrival estimation techniques based on CIR; one method is to convert CFR to CIR through the inverse Fourier transform and select the index time of the first peak as the estimated time of arrival. A series of super-resolution techniques are used for estimation; the most commonly used technique is MUSIC algorithm [49]. The other method is based on cross-correlation techniques such as matched filtering [50]. Therefore, we believe that adding perturbations to the time characteristics can randomly shift the overall multipath information of CIR forward or backward, thereby perturbing the index of the first peak. Setting the random perturbation constant as λ, where $\lambda \sim U(-b,b)$, and for the UE n, the continuous CIR information affected by a delay perturbation can be written as:

$$H^n_{shift}(\tau,t) = H^n_{u,s}(\tau,t) * \delta(t-\lambda) \tag{12}$$

where $*$ represents the convolution operation, and λ is the translation coefficient with a maximum translation size of b.

5. Convolutional Neural Network for Indoor Positioning

For CIR samples, the structure and dimension of the CIR input are similar to image input, so AI methods used for image processing are considered as our positioning schemes. The most commonly used neural network models for image processing are based on the CNN and self-attention mechanism. Although the concept of deep neural networks is stacking neural networks together, which is observed as a simple process, the performance of these networks can vary greatly due to different network architectures and choices of hyperparameters.

With regard to the models based on CNN, there are several mainstream network structures. AlexNet [51] introduced the concept of deep CNNs and addressed the vanishing gradient problem by utilizing the Rectified Linear Unit (ReLU) activation function while it employed a dropout regularization to prevent overfitting. GoogLeNet [52] further advances the field with its inception module architecture, which allows for the simultaneous use of different-sized convolutional kernels and pooling layers, enabling the extraction of features at multiple scales. On the other hand, ResNet [53] introduces the concept of residual learning. It addresses the degradation problem that arises when deep neural networks suffer from a diminishing performance with increasing depth. By employing skip connections, ResNet allows for the direct flow of input to the output layer, facilitating the learning of residual mappings.

The self-attention mechanism [54] focuses on the correlation of vectors in input sequences, and it was first used for semantic comprehension. When applying the self-attention to image processing tasks, images are divided into small parts and input as sequences. The most typically used model is the Vision Transformer (ViT) [55]. ViT is an effective tool for handling large images and complex visual scenes. This model can also enhance its performance through pre-training.

To effectively extract features for positioning, we consider a ResNet structure as the basic model for AMT. It is applied in both student and teacher models. The ResNet has several advantages, including its extremely deep network structure, which enhances the network's ability for feature extraction. Additionally, it introduces the residual blocks, which prevent network degradation, gradient vanishing, or gradient explosion in deeper networks.

5.1. Residual Network

ResNet features two main types of residual blocks: the basic block and the bottleneck block. The basic block consists of two 3×3 convolutional layers and a residual connection with a stride of 1, while the bottleneck block includes a 1×1 convolutional layer, a 3×3 convolutional layer, another 1×1 convolutional layer, and a residual connection. The 1×1 convolutional layer in the bottleneck block is primarily used to decrease the dimension of the feature map, thereby reducing the computation and parameter count.

In this paper, we consider a basic block of residual blocks. The structure of residual blocks is shown in Figure 3. The core of ResNet's residual blocks lies in its residual connection, which adds the output of the previous layer to the current layer's output, enabling the current layer to incorporate information from the previous layer. This design effectively alleviates the problem of gradients vanishing, making the model easier to train and optimize.

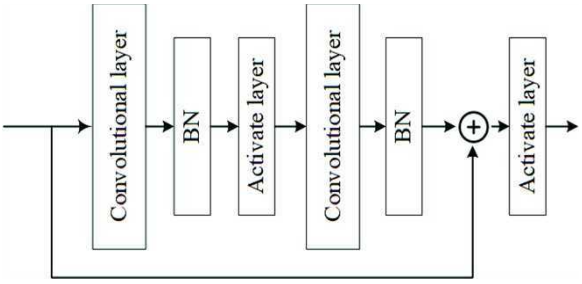


Figure 3. Structure of a residual block.

The ResNet consists of residual blocks that are based on the convolutional layer. For the purpose of accelerating the model convergence speed and improving model performance, we added BN and activation function layers between convolutional layers in the residual structure of ResNet. The Batch Normalization (BN) layer normalizes the input of each layer in a deep neural network for each batch, which stabilizes the input distribution of each layer and accelerates model convergence speed. Assuming the input after augmentation is x, for a layer with multiple input dimensions, the BN operation [56] for each dimension can be expressed as:

$$\hat{x} = \frac{x - E(x)}{\sqrt{Var(x) + \varepsilon}} \tag{13}$$

where $E(x)$ and $\sqrt{Var(x)}$ present the expectation and variance of the input mini-batch respectively, and ε is a small coefficient to prevent the denominator from being zero, which approximates to 0.

Alternatively, the activation function layer, ReLU function, is used in the structure due to its simplicity and non-linearity. This function is added after the BN operation to

effectively mitigate the gradient vanishing problem and enhances the model's performance. The activation function can be mathematically presented as

$$\sigma(\cdot) = ReLU(\hat{x}) = max(\hat{x}, 0), \tag{14}$$

5.2. CNN-Based Regression Positioning Method

In the previous research about classification-based fingerprint positioning, the area was divided into small grids, and the fingerprints were mapped to the reference point (RP) in the grid [57,58]; however, this approach is not suitable for large areas because of the extensive number of classes it needs to be divided into, and this will greatly increase the model complexity in deep learning-based fingerprint positioning. Meanwhile, a straightforward classification method of fingerprint positioning first defines RPs by collecting features at different points and then finding the similarity between the target feature and different RPs. This heavily depends on the number and density of RPs because of the limited training space; it is hard to reach sub-meter level accuracy in large areas. In contrast, the regression method can overcome the discontinuity of RPs and has the potential to reach higher accuracy, while it is also insensitive to the size of areas. From these concerns, we use the regression method rather than the classification.

6. Simulations

In this section, we conduct simulation experiments based on the open-source dataset in order to evaluate the positioning performance of our proposed method.

6.1. Simulation Settings

3GPP Technical Report 38.901 [47] has outlined an indoor factory with dense clutter and a high base station height (InF-DH) scenario with its lower probability of LOS conditions. We evaluate the performance of our proposed AMT method based on the InF-DH scenario specified by 3GPP.

The indoor factory scenario is a highly intricate industrial environment characterized by obstacles. As shown in Figure 4, 18 BSs are deployed on the ceiling of the factory with a length of 120 m and a width of 60 m. The clutters in the scene can be considered as machinery, assembly lines, and storage shelves, and UEs are uniformly distributed in the area.

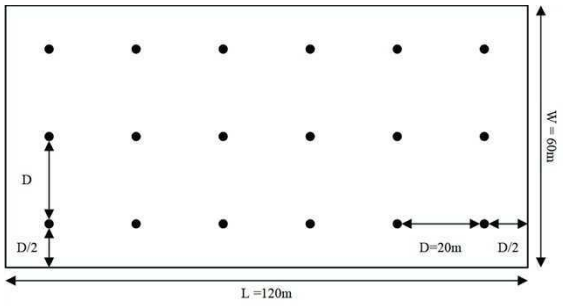


Figure 4. Indoor positioning scenario.

We use an open-source indoor measurement dataset WLI_3_ 1015_InF_DH662_FR1 _drop1_ cir_ rsrp _toa_tdoa from [59], the full parameter settings of the dataset simulation can be found in the introduction_positioning file, and we list partial parameter setting of system-level simulation in Table 1. The dataset generated by the system-level simulator is generated according to the 5G-NR standard of 3GPP [47]. In the system-level simulation, the clutter's density, height, and diameter are 60%, 6 m, and 2 m, respectively, which are used to generate a heavy NLOS indoor scenario. The BS antenna configuration is denoted by (M^a, N^a, P, M_g^a, N_g^a), and M^a and N^a indicate that a single panel array antenna has an $M^a \times N^a$ uniform planar array, where N^a is the antenna in each row and M^a is the number of rows in the vertical dimension. P represents the number of polarization dimensions, M_g^a

and N_g^a illustrate that the antenna system consists of M_g^a vertical panels and N_g^a horizontal panels, and in the parameter setting, we only use one antenna in the receiver/UE side. Spatial consistency is used to keep the various channel generation steps spatially consistent for a drop-based simulation, and this process is not used in the simulator. The UE is moving in a random direction with a fixed speed of 3 km/h, which means the orientation of the antenna is random.

Table 1. Parameter settings for generating datasets.

Parameters	Values
Clutter density, height, size	0.6, 6 m, 2 m
Bandwidth	100 M
TX power of total base station(BS)	24 dBm
BS antenna configuration	$(M^a, N^a, P, M_g^a, N_g^a) = (4, 4, 2, 1, 1)$
Antenna height of user equipment(UE)	1.5 m
BS height	8 m
Carrier frequency	3.5 GHz
Subcarrier spacing	30 kHz for 100 MHz
Spatial Consistency	No
Synchronization between BS and UE	Ideal
Penetration loss	0 dB
Number of floors	1
UE mobility	3 km/h
Min BS-UE distance (two dimensional)	0 m

From the settings of the system-level simulation, we can infer that there may exist some errors in AOA because of the randomness of UEs' orientation. These errors will directly affect the phase of CIR by introducing errors in field patterns, which will influence the measurement of AOA. However, because the useful angle information for the position estimation contained in CIR exits in the angle difference between antennas and we only use one antenna for receiving signals, it will not result in many errors in our positioning scheme. However, when facing the effects of antenna orientation, improvements in the estimation method of AoA can help increase the estimation accuracy of AoA [60], and using machine learning methods to assist position calculation using AoA estimation can also help enhance the positioning accuracy [61]. Also, the number of antennas is proven to affect the positioning performance by improving the resolution in AoA estimation [62]; therefore, in practical situations, the user devices with more antennas will also benefit the positioning performance.

We sample CIR data by truncating the first 256 time-domain points based on the first Tx antenna element and the first Rx antenna element from CIR, and the corresponding multipath characteristic of CIR is represented in Figure 5. In the dataset, the CIR received by each UE is represented as a three-dimensional vector of "$18 \times 256 \times 2$", which represents the UE received CIRs from 18 BSs, and each CIR contains 256 complex-valued sampling points.

The dataset consists of 80,000 CIRs corresponding to true UE coordinates. A total of 78,400 data samples are used as the training set while 1600 are used for a test set. Ignoring the inter-UE correlation changes caused by the reduction in the number of UEs, we randomly select 3000 or fewer UEs from the training set as labeled data to simulate the limited labeled data scenario for indoor positioning. We assign a label of 0 to the remaining data, treating them as unlabeled data. The student and teacher models are trained with a fixed ratio of 1:1 for unlabeled and labeled data. To maintain the fixed ratio of labeled and unlabeled data, a two-stream batch sampler is used to obtain a training batch, and a batch size of 256 is used for each stream. In the augmentation process, we select 5% of labeled data samples to operate amplitude scaling or temporal shifting and set the maximum scaling size a as 3 while the maximum translation size b is set to 5. We should keep the perturbed data as a small portion of the labeled data to ensure that critical positioning features are not completely destroyed. It should be noted that the perturbation here should operate in

the training steps of each mini-batch instead of scaling the amplitude for the overall data first. For the reason that a large learning rate is prone to causing non-convergence or a gradient explosion phenomenon in the simulation, we carefully select an optimizer and a small learning rate.

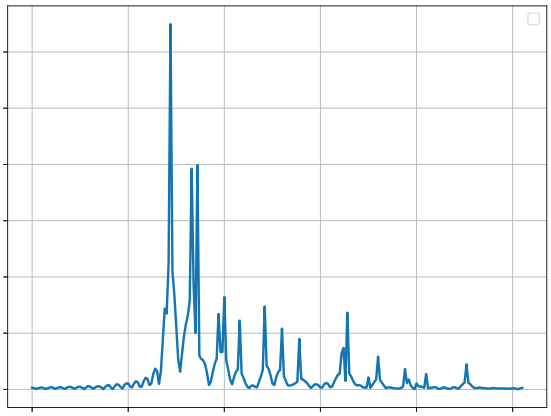


Figure 5. The received channel impulse response(CIR) of a UE from a single BS.

For the parameter setting of AMT, we set the ramp-up length T_{max} to 80, which means that $w(T/T_{max})$ linearly increases to 1 within 80 epochs. To prevent training from degrading, $w(T/T_{max})$ is set to 0 when the epoch starts from 0. We set the maximum value of the unsupervised weight w_{max} to 70. In addition, the smoothing coefficient α of EMA is fixed at 0.97. The value of α is set to ensure that the teacher model keeps up with the rapidly evolving student model by promptly disregarding erroneous prior values of the student model. The performance of the teacher model is considered to be the final result. The detailed parameter settings of AMT are concluded in Table 2.

Table 2. Parameter settings for AMT training.

Parameters	Values
Number of samples	80,000
Number of samples for training	78,400
Number of samples for testing	1600
Network structure $f(\cdot)$	Resnet
Batch sampler	Two-stream batch sampler
Batch size	512 (256 for each stream)
w_{max}	80
T_{max}	80
α	0.97
Training epoch	200
Optimizer	Stochastic Gradient Descent
Learning rate	0.001
Perturbed proportion of labeled data	5%

In our simulation experiments, positioning accuracy and convergence are used as the performance metrics of the proposed AMT model. In particular, we adopt 90%, 80%, 67%, and 50% points of the Cumulative Distribution Function (CDF) of the positioning accuracy to illustrate the performance gain in accordance with the evaluation methodology in 3GPP TR38.857.

6.2. The ResNet Model for Processing CIR

For the model basic structure $f(\cdot)$, we design a ResNet to extract effective positioning features from CIR information, which is originally used for processing multi-channel image information in the field of image processing [53].

A ResNet structure for feature extraction is shown in Figure 6. It consists of four reshape layers and six residual blocks, while the reshape operation serves to preprocess the data and adapt it to the input dimension of the residual block.

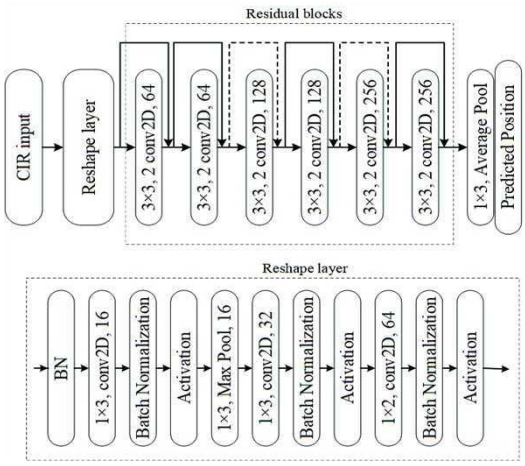


Figure 6. Structure of Resnet.

Specifically, in the design, the input data dimension is considered as a 2-channel image data of size 18×256. In the first four layers of the network, the dimension of the input information increases to 18×18 with 64 channels so that it is able to be processed by 3×3 convolutional kernels in the residual blocks. Then, the residual blocks are used for training, with the input being directly added to the output of a residual block, which is known as a shortcut connection. In terms of a shortcut operation, when a dimension change occurs between residual blocks, the input propagation between blocks needs to be dimensionally changed to adapt to the next residual block. This procedure is marked by the dotted line in Figure 6. After the pooling layer, a linear layer with an output dimension of two is used to obtain the UE's two-dimensional coordinates.

6.3. Results and Discussions

6.3.1. Performance Evaluation of the Supervised Learning-Based Method

Before conducting the simulation of the semi-supervised and enhanced indoor positioning under limited labels, we first evaluate the performance of positioning under a supervised learning situation, which uses the full labels in the fingerprint dataset and an original CIR input without augmentation. For an objective comparison, we also use other fingerprinting methods based on deep learning and a traditional estimation method to test their positioning accuracy under the heavy NLOS scenario.

The full number of labeled data for input is 80,000. In the supervised learning trial, we also use the rate of 2% to divide the training set and test set. In CIR-based fingerprint indoor positioning, we use the ResNet structure mentioned in 5.1 to conduct the simulation. In the training process, the batch size and learning rate are set to 512 and 0.1 respectively; the adam optimizer and cosine-annealing scheduler are also used in the training.

Meanwhile, we also process the TOA, TDOA, and RSS fingerprints that are collected in the same scenario with the same collecting steps to deal with the positioning task using a learning-based method. The structure of a single datum is 1×18, which stands for the TOA/TDOA/RSS signal received from 18 BSs. These fingerprint datasets are also listed in WLI_3_1015_InF_DH662_FR1_drop1_ cir_rsrp_toa _tdoa from [59]. To avoid destroying the dimension of these fingerprints, we applied a Multi-layer Perception(MLP) network to recognize the features. The MLP structure consists of 4 linear layers with an activation function ReLU connected to each linear layer and a dropout operation before the last linear layer.

Additionally, we also apply CHAN, a traditional method that is widely used in TDOA-based positioning [21]. By utilizing the TDOA fingerprint values and the known

BS positions to apply the Weighted Least Squares (WLS) algorithm, we can obtain the calculated position of UEs, noting that the algorithm is under the case of more than three reference nodes.

The position performance can be reflected by the CDF of positioning errors and Mean Error (ME), which are shown in Table 3. By comparing the performance of the learning-based fingerprint positioning and CHAN, we can conclude that AI positioning methods are superior to the traditional TDOA-based estimation algorithm. In the performance of learning-based fingerprint positioning, all displayed methods reach the accuracy of sub-meter level, while the CIR-based method has the best positioning accuracy, which reaches an ME of 0.31 m.

Table 3. Comparison of position accuracy based on supervised learning

Method for Indoor Positioning Using Deep Learning	CDF Percentile of Position Errors (m)				ME (m)
	50%	67%	80%	90%	
CIR-based fingerprint positioning	0.28	0.35	0.45	0.54	0.31
TOA-based fingerprint positioning	0.36	0.46	0.57	0.69	0.40
TDOA-based fingerprint positioning	0.38	0.49	0.60	0.74	0.41
RSS-based fingerprint positioning	0.30	0.41	0.52	0.68	0.38
CHAN	10.35	16.01	22.08	31.14	14.72

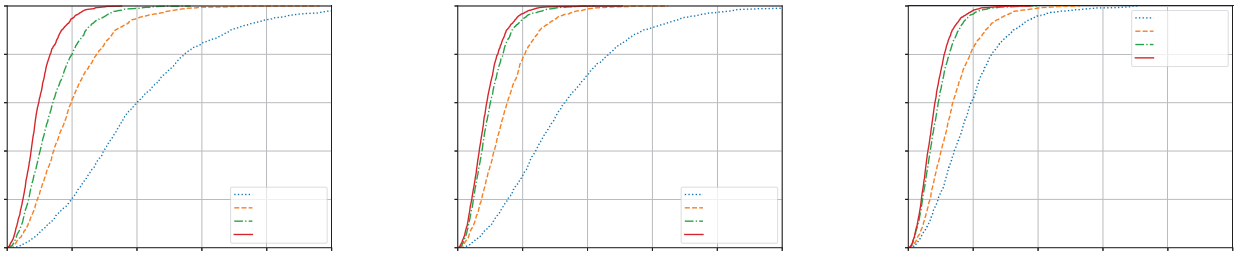
After the performance evaluation of CIR-based fingerprint positioning, we can observe that it outperforms positioning methods based on other measurements collected from the same scenario; thus, it is more effective to use it as the basic module in our proposed training framework.

6.3.2. Numerical Results of Semi-Supervised Learning-Based Method

In this simulation, we conducted tests with a very small sample size, specifically using 3000 or fewer samples to evaluate the performance.

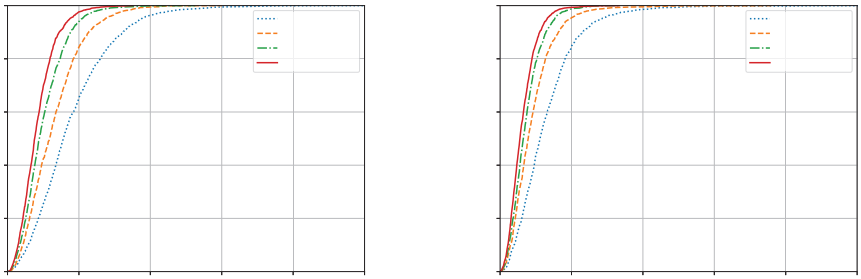
On the basis of supervised learning, we first verified the effectiveness of the proposed data augmentation methods. From Figure 7, it can be observed from four performance curves with different numbers of labels that the number of samples is positively correlated with the positioning accuracy. Additionally, from the comparison of Figure 7a–c, it is clear that both proposed data augmentation methods can effectively improve the positioning accuracy for all test samples in supervised learning (SL) methods, with the most significant improvement observed for sample sizes ranging from 500 to 2000. This demonstrates that our proposed data augmentation methods can adapt to scenarios with very few labeled data and meet practical needs. In terms of the effectiveness of the two data augmentation methods, amplitude scaling is significantly better than temporal shifting when the number of labels is 500, and it also performs slightly better than the temporal shifting for other sample sizes.

To judge whether the unlabeled data are effective for a performance improvement in DSSL methods, we obtain the positioning accuracy of AMT to compare with the performance of purely supervised learning using a few labeled data to train the model. After verifying the effectiveness of two data augmentation methods in the SL method, we conduct the AMT model by using the two proposed augmentation methods. By comparing Figure 8a,b to Figure 7b,c), we observe that the performance of AMT significantly improves in four cases compared to purely supervised learning methods, while it applies two proposed data augmentation techniques. Additionally, similar to purely supervised learning, the use of the random amplitude scaling augmentation method attains better performances than random temporal shifting in four cases. Furthermore, the AMT method in the case of small sample sizes demonstrates more significant improvement, specifically when the number of labeled samples is less than 2000. A visualization example to show the performance of AMT using 1000 labeled samples is present in Figure 9.



(**a**) (**b**) (**c**)

Figure 7. Supervised positioning performance under different samples and augmentation methods. (**a**) Performance of supervised learning(SL) method only. (**b**) Performance of SL method with temporal-shifting augmentation. (**c**) Performance of SL method with amplitude-scaling augmentation.



(**a**) (**b**)

Figure 8. Performance of AMT with different augmentations. (**a**) Performance of AMT with temporal-shifting augmentation. (**b**) Performance of AMT with amplitude-scaling augmentation.

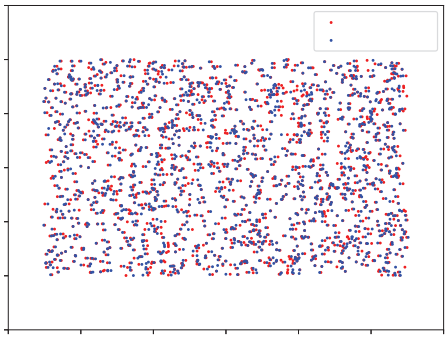


Figure 9. A visualization form to show the distribution of UEs. The red dots represent the actual user locations, while the blue dots represent the predicted user locations.

Furthermore, we verify the effectiveness of the proposed AMT model by comparing it with several reference algorithms under 1000 labeled samples. In order to facilitate better horizontal comparison, we will use some DSSL methods that have similar training frameworks and steps to AMT as benchmark algorithms. One reference algorithm is the Π Model and the other is the Temporal Ensembling. Both two algorithms are advanced consistency regularization methods, and the same data augmentations are used for the three approaches. In addition, we also compare the performance of the pseudo label to verify the effectiveness of our method because it is widely used for positioning. As numerical results are shown in Table 4, we can observe that the three consistency regularization methods significantly outperform the pseudo label in positioning accuracy. Moreover, among the three consistency regularization methods, the AMT model achieves the highest accuracy.

Meanwhile, we compare the convergence of the four methods and record the tendency of the positioning error corresponding to 90 percentile CDF. During the training process, we can observe from Figure 10 that although there is a small difference in the positioning error of AMT, Π Model and Temporal Ensembling, the convergence of AMT is significantly faster than the other two methods, while the convergence curve seems to be smoother. This implies that under the condition of limited training time, the AMT model demonstrates obvious advantages over the other two referenced consistency regularization methods.

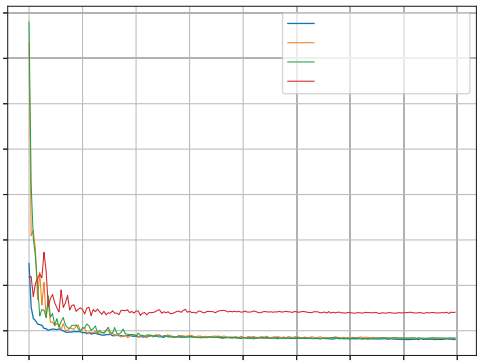


Figure 10. Tendency of position errors.

Table 4. Comparison of position accuracy under 1000 labeled samples.

DSSL Method for Indoor Positioning	CDF Percentile of Position Errors (m)				ME(m)
	50%	67%	80%	90%	
Proposed AMT with amplitude scaling	0.79	1.03	1.27	1.63	0.90
Proposed AMT with temporal shifting	1.18	1.53	1.85	2.27	1.29
Temporal Ensembling with amplitude scaling	0.82	1.06	1.33	1.65	0.93
Temporal Ensembling with temporal shifting	1.20	1.54	1.88	2.33	1.31
Π Model with amplitude scaling	0.84	1.10	1.36	1.74	0.96
Π Model with temporal shifting	1.20	1.54	1.89	2.33	1.32
Pseudo label	1.45	1.87	2.29	2.93	1.65
Supervised only	1.71	2.19	2.76	3.33	1.91

During the training process, we also recorded the tendency of positioning error in Figure 11. The utilization of AMT can be understood through the training curves. The models with EMA weighting (represented by the orange curves in the bottom row) exhibit more accurate predictions compared to the student models (represented by the blue curves) after an initial period.

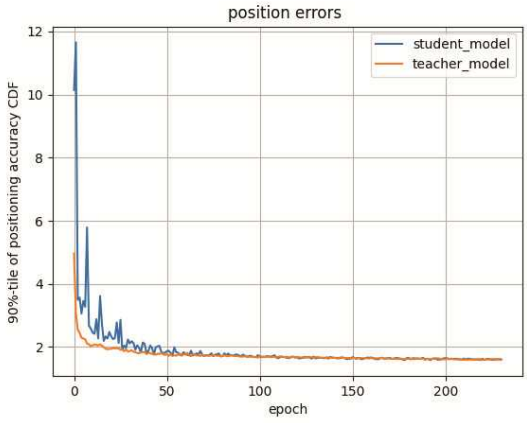


Figure 11. Positioning errors of teacher and student models under 1000 labeled samples.

When the EMA-weighted model is employed by the teacher, it leads to enhanced outcomes in semi-supervised scenarios. A virtuous feedback cycle appears where the teacher model (orange curve) improves the student model (blue curve) through the application of consistency cost. Simultaneously, the student model contributes to the improvement of the teacher model via EMA.

6.3.3. Discussions

To analyze the performance differences between random amplitude scaling and random temporal shifting, the amplitude scaling technique performs better than temporal shifting. In this paper, we consider several possible reasons. First, time-based ranging typically uses an external signal source and heavily relies on complex signal processing methods, which means it generally relies on super-resolution methods like the MUSIC algorithm to obtain the first arrival peak as an arrival time. However, in our CIR training sets, CIR information is not continuous; it is selected as time intervals, which makes the time sequence discrete. This decreases the resolution of the time sequence in CIR and possibly leads to an incorrect recognition for the model. The power-based features vary over time, but the shape of the peak does not change rapidly between a few sampling points. Moreover, the entire CIRs' peaks can be utilized to estimate the location, rather than solely relying on the arrival time of the first peak. This may be a reason why the power-based feature is more effective than the time-based feature, and power scaling exhibits a better performance in augmentation. Meanwhile, the time-based ranging relies highly on the LOS condition, a positive bias could be induced in NLOS propagation and a larger variance tends to exist in the NLOS condition [63].

To analyze the performance of methods in Table 4, we re-examine the fundamental principles of these four methods. The Π Model and Temporal Ensembling only use a consistency loss for learning while the AMT uses both the weights of the student model and consistency loss for learning. Interactions in weights can help learn the performance of supervised training rapidly from the beginning instead of receiving feedback at the end of each epoch to punish inconsistent items, which greatly reduces time costs. Similarly, like AMT, the pseudo label also uses the training weights of supervised training, manifesting as no significant positioning error at the beginning of training. However, due to the small amount and sparse distribution of labeled data, the pseudo-label method cannot learn information beyond this sparse distribution, which leads to a limited performance during training with dense but unlabeled data, and results in a relatively low positioning accuracy.

7. Conclusions

In this paper, we proposed an effective DSSL framework for InF-DH scenarios named AMT, which solves the problem of inaccurate positioning caused by inadequately labeled samples for fingerprint positioning. In the DSSL framework, we operate AMT by assigning the EMA weights of the student model to the teacher model and regularizing the consistency loss of two models. Additionally, we have also proposed novel implicit random augmentation methods in terms of amplitude and temporal features of CIR data in AMT to enhance the performance of neural networks. By adding random perturbation to these critical positioning features in the training process, continuous new data can be generated artificially from existing data.

From conducting the simulation, we conclude that the deep-learning method has a superior performance compared with the estimation method in the heavy NLOS scenario, which can achieve a sub-meter level accuracy, and using CIR for deep learning-based positioning can achieve a higher positioning accuracy than using other measurements proposed in the 5G NR standard. By analyzing the performances of the DSSL methods, the numerical results show that using regularization in consistency loss improves the neural network's ability to resist perturbation, which helps learn effective features in unlabeled data, especially in the case of small samples. Meanwhile, the interaction in model weights for consistency regularization methods improves the convergence of neural networks.

Additionally, results also show the robustness of AMT by using different numbers of labeled data, and its performance gain in the smaller samples is more obvious.

By using AMT, we find the inherent similarity between image processing and indoor positioning, and we also verify the effectiveness of image processing methods when applying them to the positioning field. However, the work does not involve using the angle feature for positioning because of the single receiving antenna set in the dataset. Since multiple receiving antennas can bring more channel information for positioning, we plan to explore the generalization of our proposed method in multiple-receiver situations in future studies.

Author Contributions: Conceptualization, P.C.; Methodology, P.C. and Y.L.; Resources, B.Y.; Data curation, W.L. and J.W. (Jingyi Wang); Writing—original draft preparation, P.C., Y.L., W.L., J.W. (Jingyi Wang) and B.Y.; Writing—review and editing, P.C., Y.L. and J.W. (Jianxiu Wang); Supervision, J.W. (Jianxiu Wang), B.Y. and G.F.; Project administration, B.Y.; Funding acquisition, J.W. (Jianxiu Wang) and P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by China Telecom Research Institute.

Data Availability Statement: Restrictions apply to the availability of these data. Data were obtained from OPPO Research Institute and are available at wireless-intelligence.com (accessed on 6 November 2023) with the permission of OPPO Research Institute.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kaplan, E.D.; Hegarty, C. *Understanding GPS/GNSS: Principles and Applications*, 3rd ed.; Artech House: New York, NY, USA, 2017.
2. del Peral-Rosado, J.A.; Raulefs, R.; López-Salcedo, J.A.; Seco-Granados, G. Survey of cellular mobile radio localization methods: From 1G to 5G. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1124–1148. [CrossRef]
3. Mogyorósi, F.; Revisnyei, P.; Pašić, A.; Papp, Z.; Törös, I.; Varga, P.; Pašić, A. Positioning in 5G and 6G networks—A survey. *Sensors* **2022**, *22*, 4757. [CrossRef] [PubMed]
4. Alhomayani, F.; Mahoor, M.H. Deep learning methods for fingerprint-based indoor positioning: A review. *J. Locat. Based Serv.* **2020**, *14*, 129–200. [CrossRef]
5. 3GPP TS 38.300. NR; NR and NG-RAN Overall Description. 2021. Available online: https://www.3gpp.org/ftp/Specs/archive/38_series/38.300 (accessed on 6 November 2023).
6. Miao, H.; Yu, K.; Juntti, M.J. Positioning for NLOS Propagation: Algorithm Derivations and Cramer–Rao Bounds. *IEEE Trans. Veh. Technol.* **2006**, *4*, IV. [CrossRef]
7. 3GPP TR 38.857. Study on NR Positioning Enhancements. 2021. Available online: https://www.3gpp.org/ftp/Specs/archive/38_series/38.857 (accessed on 6 November 2023).
8. Zafari, F.; Gkelias, A.; Leung, K.K. A survey of indoor localization systems and technologies. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2568–2599. [CrossRef]
9. Yang, Z.; Zhou, Z.; Liu, Y. From RSSI to CSI: Indoor Localization via Channel Response. *ACM Comput. Surv.* **2013**, *46*, 25. [CrossRef]
10. Zheng, Y.; Liu, J.; Sheng, M.; Zhou, C. Exploiting fingerprint correlation for fingerprint-based indoor localization: A deep learning-based approach. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5762–5774. [CrossRef]
11. 3GPP TSG-RAN Meeting #96. Revised SID: Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface. 2022. Available online: https://www.3gpp.org/ftp/TSG_RAN/TSG_RAN/TSGR_96/Docs/RP-221348.zip (accessed on 6 November 2023).
12. 3GPP TSG-RAN WG1 #112bis-e. Summary of Evaluation on AI/ML for Positioning Accuracy Enhancement. 2023. Available online: https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_112b-e/Docs/R1-2304106.zip (accessed on 6 November 2023).
13. Cerar, G.; Švigelj, A.; Mohorčič, M.; Fortuna, C.; Javornik, T. Improving CSI-based Massive MIMO Indoor Positioning using Convolutional Neural Network. In Proceedings of the 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Porto, Portugal, 8–11 June 2021; pp. 276–281. [CrossRef]
14. Chin, W.L.; Hsieh, C.C.; Shiung, D.; Jiang, T. Intelligent Indoor Positioning Based on Artificial Neural Networks. *IEEE Netw.* **2020**, *34*, 164–170. [CrossRef]
15. Laine, S.; Aila, T. Temporal Ensembling for Semi-Supervised Learning. *arXiv* **2017**, arXiv:cs.NE/1610.02242.
16. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.

17. Brena, R.F.; García-Vázquez, J.P.; Galván-Tejada, C.E.; Muñoz-Rodriguez, D.; Vargas-Rosales, C.; Fangmeyer, J. Evolution of indoor positioning technologies: A survey. *J. Sens.* **2017**, *2017*, e2630413. [CrossRef]
18. Haeberlen, A.; Flannery, E.; Ladd, A.M.; Rudys, A.; Wallach, D.S.; Kavraki, L.E. Practical Robust Localization over Large-Scale 802.11 Wireless Networks. In Proceedings of the MobiCom'04, 10th Annual International Conference on Mobile Computing and Networking, Philadelphia, PA, USA, 26 September–1 October 2004; pp. 70–84. [CrossRef]
19. Sadowski, S.; Spachos, P. RSSI-Based Indoor Localization with the Internet of Things. *IEEE Access* **2018**, *6*, 30149–30161. [CrossRef]
20. Xiong, J.; Jamieson, K. Arraytrack: A fine-grained indoor location system. In Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13), Usenix, Lombard, IL, USA, 2–5 April 2013; pp. 71–84.
21. Chan, Y.T.; Ho, K. A simple and efficient estimator for hyperbolic location. *IEEE Trans. Signal Process.* **1994**, *42*, 1905–1915. [CrossRef]
22. Dai, H.; Ying, W.h.; Xu, J. Multi-layer neural network for received signal strength-based indoor localisation. *IET Commun.* **2016**, *10*, 717–723. [CrossRef]
23. Hoang, M.T.; Yuen, B.; Dong, X.; Lu, T.; Westendorp, R.; Reddy, K. Recurrent Neural Networks for Accurate RSSI Indoor Localization. *IEEE Internet Things J.* **2019**, *6*, 10639–10651. [CrossRef]
24. Wu, K.; Xiao, J.; Yi, Y.; Chen, D.; Luo, X.; Ni, L.M. CSI-based indoor localization. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *24*, 1300–1309. [CrossRef]
25. Han, S.; Li, Y.; Meng, W.; Li, C.; Liu, T.; Zhang, Y. Indoor localization with a single Wi-Fi access point based on OFDM-MIMO. *IEEE Syst. J.* **2019**, *13*, 964–972. [CrossRef]
26. Gao, Z.; Gao, Y.; Wang, S.; Li, D.; Xu, Y. CRISLoc: Reconstructable CSI Fingerprinting for Indoor Smartphone Localization. *IEEE Internet Things J.* **2021**, *8*, 3422–3437. [CrossRef]
27. Chen, Z.; Zou, H.; Yang, J.; Jiang, H.; Xie, L. WiFi Fingerprinting Indoor Localization Using Local Feature-Based Deep LSTM. *IEEE Syst. J.* **2020**, *14*, 3001–3010. [CrossRef]
28. Yang, S.; Zhang, D.; Song, R.; Yin, P.; Chen, Y. Multiple WiFi Access Points Co-Localization through Joint AoA Estimation. *IEEE Trans. Mob. Comput.* **2023**, 1–16. [CrossRef]
29. Chen, Z.; Zhu, G.; Wang, S.; Xu, Y.; Xiong, J.; Zhao, J.; Luo, J.; Wang, X. M^3M3: Multipath Assisted Wi-Fi Localization with a Single Access Point. *IEEE Trans. Mob. Comput.* **2021**, *20*, 588–602. [CrossRef]
30. Zhuang, Y.; Zhang, C.; Huai, J.; Li, Y.; Chen, L.; Chen, R. Bluetooth localization technology: Principles, applications, and future trends. *IEEE Internet Things J.* **2022**, *9*, 23506–23524. [CrossRef]
31. Pei, L.; Liu, J.; Guinness, R.; Chen, Y.; Kröger, T.; Chen, R.; Chen, L. The evaluation of WiFi positioning in a Bluetooth and WiFi coexistence environment. In Proceedings of the 2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), Helsinki, Finland, 3–4 October 2012; pp. 1–6. [CrossRef]
32. Alarifi, A.; Al-Salman, A.; Alsaleh, M.; Alnafessah, A.; Al-Hadhrami, S.; Al-Ammar, M.A.; Al-Khalifa, H.S. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors* **2016**, *16*, 707. [CrossRef]
33. Poulose, A.; Han, D.S. UWB indoor localization using deep learning LSTM networks. *Appl. Sci.* **2020**, *10*, 6290. [CrossRef]
34. Van Diggelen, F.S.T. *A-GPS: Assisted GPS, GNSS, and SBAS*; Artech House: New York, NY, USA, 2009.
35. Gao, Y.; Jing, H.; Dianati, M.; Hancock, C.M.; Meng, X. Performance Analysis of Robust Cooperative Positioning Based on GPS/UWB Integration for Connected Autonomous Vehicles. *IEEE Trans. Intell. Veh.* **2023**, *8*, 790–802. [CrossRef]
36. Zhang, B.; Sifaou, H.; Li, G.Y. CSI-Fingerprinting Indoor Localization via Attention-Augmented Residual Convolutional Neural Network. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 5583–5597. [CrossRef]
37. Ruan, Y.; Chen, L.; Zhou, X.; Liu, Z.; Liu, X.; Guo, G.; Chen, R. iPos-5G: Indoor Positioning via Commercial 5G NR CSI. *IEEE Internet Things J.* **2023**, *10*, 8718–8733. [CrossRef]
38. Tedeschini, B.C.; Nicoli, M. Cooperative Deep-Learning Positioning in mmWave 5G-Advanced Networks. *IEEE J. Sel. Areas. Commun.* **2023**, *41*, 3799–3815. [CrossRef]
39. Lee, D.H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Proceedings of the Workshop on Challenges in Representation Learning, ICML, Atlanta, GA, USA, 16–21 June 2013; Volume 3, p. 896.
40. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
41. Kipf, T.N.; Welling, M. Variational Graph Auto-Encoders. *arXiv* **2016**, arXiv:stat.ML/1611.07308.
42. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In Proceedings of the 34th Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 596–608.
43. Li, W.; Zhang, C.; Tanaka, Y. Pseudo label-driven federated learning-based decentralized indoor localization via mobile crowdsourcing. *IEEE Sens. J.* **2020**, *20*, 11556–11565. [CrossRef]
44. Chen, K.M.; Chang, R.Y. Semi-supervised learning with GANs for device-free fingerprinting indoor localization. In Proceedings of the GLOBECOM 2020–2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
45. Zhang, M.; Fan, Z.; Shibasaki, R.; Song, X. Domain Adversarial Graph Convolutional Network Based on RSSI and Crowdsensing for Indoor Localization. *IEEE Internet Things J.* **2023**, *10*, 13662–13672. [CrossRef]

46. He, Y.W.; Hsu, T.T.; Tseng, P.H. A Semi-Supervised Ladder Network-Based Indoor Localization Using Channel State Information. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–13. [CrossRef]
47. 3GPP TR 38.901. Study on Channel Model for Frequencies from 0.5 to 100 GHz. 2020. Available online: https://www.3gpp.org/ftp/Specs/archive/38_series/38.901 (accessed on 6 November 2023).
48. Bachman, P.; Alsharif, O.; Precup, D. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems 27*; Curran Associates, Inc.: Red Hook, NY, USA, 2014.
49. Li, X.; Pahlavan, K. Super-resolution TOA estimation with diversity for indoor geolocation. *IEEE Trans. Wirel. Commun.* **2004**, *3*, 224–234. [CrossRef]
50. Golden, S.A.; Bateman, S.S. Sensor Measurements for Wi-Fi Location with Emphasis on Time-of-Arrival Ranging. *IEEE Trans. Mob. Comput.* **2007**, *6*, 1185–1198. [CrossRef]
51. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Pereira, F., Burges, C.J., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
52. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
54. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
55. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:cs.CV/2010.11929.
56. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; Volume 37, pp. 448–456.
57. Hsieh, C.H.; Chen, J.Y.; Nien, B.H. Deep learning-based indoor localization using received signal strength and channel state information. *IEEE Access* **2019**, *7*, 33256–33267. [CrossRef]
58. Iqbal, Z.; Luo, D.; Henry, P.; Kazemifar, S.; Rozario, T.; Yan, Y.; Westover, K.; Lu, W.; Nguyen, D.; Long, T.; et al. Accurate real time localization tracking in a clinical environment using Bluetooth Low Energy and deep learning. *PLoS ONE* **2018**, *13*, e0205392. [CrossRef]
59. OPPO Research Institute. Wireless Intelligence. Available online: https://wireless-intelligence.com (accessed on 6 November 2023).
60. Ledergerber, A.; Hamer, M.; D'Andrea, R. Angle of Arrival Estimation based on Channel Impulse Response Measurements. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2018; pp. 6686–6692. [CrossRef]
61. Comiter, M.; Kung, H.T. Localization Convolutional Neural Networks Using Angle of Arrival Images. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–7. [CrossRef]
62. Bast, S.D.; Guevara, A.P.; Pollin, S. CSI-based Positioning in Massive MIMO systems using Convolutional Neural Networks. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–5. [CrossRef]
63. Gezici, S.; Tian, Z.; Giannakis, G.B.; Kobayashi, H.; Molisch, A.F.; Poor, H.V.; Sahinoglu, Z. Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks. *IEEE Signal Process. Mag.* **2005**, *22*, 70–84. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

mathematics

MDPI

Article

Exploiting Cross-Scale Attention Transformer and Progressive Edge Refinement for Retinal Vessel Segmentation

Yunyi Yuan [1], Yingkui Zhang [2], Lei Zhu [3], Li Cai [4,*] and Yinling Qian [5,*]

[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[2] Schole of Nursing, The Hong Kong Polytechnic University, Hong Kong SAR 999077, China
[3] ROAS Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511400, China
[4] Department of Ophthalmology, Shenzhen University, Shenzhen 518055, China
[5] Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
* Correspondence: caili@szu.edu.cn (L.C.); yl.qian@siat.ac.cn (Y.Q.)

Abstract: Accurate retinal vessel segmentation is a crucial step in the clinical diagnosis and treatment of fundus diseases. Although many efforts have been presented to address the task, the segmentation performance in challenging regions (e.g., collateral vessels) is still not satisfactory, due to their thin morphology or the low contrast between foreground and background. In this work, we observe that an intrinsic appearance exists in the retinal image: among the dendritic vessels there are generous similar structures, e.g., the main and collateral vessels are all curvilinear, but they have noticeable scale differences. Based on this observation, we propose a novel cross-scale attention transformer (CAT) to encourage the segmentation effects in challenging regions. Specifically, CAT consumes features with different scales to produce their shared attention matrix, and then fully integrates the beneficial information between them. Such new attention architecture could explore the multi-scale idea more efficiently, thus realizing mutual learning of similar structures. In addition, a progressive edge refinement module (ERM) is designed to refine the edges of foreground and background in the segmentation results. Through the idea of edge decoupling, ERM could suppress the background feature near the blood vessels while enhancing the foreground feature, so as to segment vessels accurately. We conduct extensive experiments and discussions on DRIVE and CHASE_DB1 datasets to verify the proposed framework. Experimental results show that our method has great advantages in the Se metric, which are 0.88–7.26% and 0.81–7.11% higher than the state-of-the-art methods on DRIVE and CHASE_DB1, respectively. In addition, the proposed method also outperforms other methods with 0.17–2.06% in terms of the Dice metric on DRIVE.

Keywords: retinal vessel segmentation; cross-scale attention transformer; progressive edge refinement

MSC: 00A06

1. Introduction

A retinal image is a direct assistance medium for non-invasive diagnosis, and retinal vessel segmentation is an important basis in the clinical diagnosis of retinal degenerative disease. The morphological variations in retinal vessels are closely related to various diseases, including diabetes, hypertension, and so on [1–3]. Manual vessel segmentation is time-consuming and laborious, and the results are also easily affected by expert experience. Hence, it is required but challenging for developing automatic vessel segmentation in clinical medical diagnosis to reduce human labor.

In the past decades, automatic retinal vessel segmentation has become a hot research task, and numerous methods have been proposed for addressing this problem with promising performances. These methods can be roughly divided into traditional methods and

deep-learning-based methods. Traditional methods have been investigated for a long time, including filtering-based methods [4,5], model-based methods [6,7], tracking-based methods [8,9], and machine-learning-based methods [10,11]. Mendonca and Campilho [4] first extract vascular centerlines and apply vessel width related filters for vessel filling. Similarly, Zhang et al. [5] use hand-crafted filters for vessel enhancement and then perform threshold processing to segment vessels. Al-Diri et al. [6] also employ a generalized morphological order filter on vessel centerlines and utilize an active contour model to capture vessel edges. And Yang et al. [7] propose an improved region-based level set model to segment vessels, which consumes gray-scale retinal image and vascular-enhanced image. With the development of machine learning, support vector machines [10] and the AdaBoost classifier [11] have been applied in some early supervised methods to achieve better performance. These methods generally utilize shallow fitting from handcrafted features for classification/segmentation.

Recently, Convolutional Neural Networks (CNNs) have achieved impressive performance on retinal vessel segmentation. They design diverse and ingenious deep networks to extract high-dimensional features, which further regress the probability that pixels belong to vessels. These methods either emphasize the discrimination and generalization ability of the deep model or focus on the morphological characteristics of retinal vessels.

Alom et al. [12] design a recurrent and residual convolutional unit to serve as the basic block to improve the discrimination of UNet [13], which is one of the most widely used deep learning frameworks in medical image analysis. To heighten important vascular features and suppress meaningless background features, Guo et al. [14] replace the deepest convolution operation of UNet with a spatial attention module for adaptive feature refinement. Additionally, Li et al. [15] leverage a similar idea to strengthen the input features. Considering the effective performance of spatial and channel attention, Wang et al. [16] and Mou et al. [17] integrate the dual attention module into their network to extract attention-aware features. For the morphological analysis, Mou et al. [18] consider the connectivity of vessels and propose a post-processing algorithm to obtain more complete vascular tree. To achieve accurate segmentation in the presence of noise, Nazir et al. [19] take advantage of the deep learning architecture to promote the cervical cytoplasm and nucleus segmentation. And a denoising variational auto-encoder network is proposed by Araújo et al. [20] to refine the segmentation results. Furthermore, Shin et al. [21] propose a vessel graph network, which combines graph neural network and CNN for joint learning to extract both global and local features.

In addition to the above methods, transformer-based neural networks have recently attracted great interest in diverse medical imaging tasks. Transformers first achieved huge success in natural language processing by capturing the long-range dependence of an input sequence via a self-attention mechanism. Dosovitskiy et al. [22] make it effectively adapt to vision tasks, which further expand the development of medical image analysis. Chen et al. [23] propose Trans-UNet to integrate a transformer and U-shaped network for medical image segmentation. To reduce the computational complexity and improve the applicability of processing high-resolution images, Liu et al. [24] utilize the shifted window technique to improve the generalization of transformer. Furthermore, Cao et al. [25] combine shifted window with UNet and construct a pure transformer-based network (i.e., Swin-UNet). Huang et al. [26] design a global transformer block to preserve detailed information and a relation transformer block to explore dependencies among lesions and other fundus tissues. To combine the advantages of the general UNet architecture and transformer, the transformer-UNet framework is popular in Li et al. [27], Shen et al. [28], Lin et al. [29]. Li et al. [27] significantly reduce the computational cost in vessel segmentation by proposing a grouping structure of convolution and transformer. Shen et al. [28] introduce the squeeze-excitation transformer into UNet for retinal vessel segmentation. Furthermore, Lin et al. [29] adopt a dual-branch Swin transformer with UNet architecture to capture the feature representations of different semantic scales. The great potential of transformer and the advantages of the transformer-based UNet framework have been presented in these

methods. By taking these into account, we propose a cross-scale attention transformer into the Swin-UNet framework. Our network retains the superior feature learning ability of transformer and promotes the feature fusion of different scales in retinal vessels.

Given an input image, the transformer network first divides it into patches by a predefined scale, e.g., 4×4 pixel grid. Then all patches are flattened into a sequence and embedded into high-dimensional features with shared weights. Therefore, the self-attention block can consume the sequence features for correlation learning. In retinal vessel segmentation, the goal is to accurately segment the foreground, including main and collateral vessels. Due to their thin morphology or the low contrast between foreground and background, most approaches are limited in their ability to segment vessels in challenging regions. We observe a momentous situation that the dendritic vessels have a great number of similar structures but with different scales. For example, the main and collateral vessels are both curvilinear, but their widths are significantly different. Therefore, transformer with one fixed patch scale has natural limitations for retinal vessel segmentation.

In order to better exploit the attention mechanism to aggregate vessel features with similar structures but different scales, a simple idea is to adopt a multi-scale scheme. Lin et al. [29] propose a dual-scale transformer UNet for medical image segmentation. They split the input image into different patches with two sizes, and apply two transformer encoders to the different scales, respectively. Nevertheless, such a dual-scale scheme with different encoders is too complex to efficiently fuse features of different scales, resulting in the inability to fully benefit from similar structures.

Based on the above observations, we propose a Cross-scale Attention Transformer, noted CAT, to serve as the basis block in the UNet architecture. CAT consumes features with different patch scales to produce their shared attention matrix and then integrates the beneficial information between them. Such a design makes full use of the characteristics of the dendritic structure in retinal images, encouraging the features with different scales to benefit from each other. Consequently, the proposed CAT can locate and separate more collateral vessels while maintaining the segmentation effect of main vessels. In addition, CAT is more lightweight than the dual-scale scheme.

Moreover, there are typically misclassifications in the edge of foreground and background as the scale differences between main and collateral vessels. To alleviate this problem, we further propose a progressive edge refinement module, called ERM. We adopt the deep-supervision scheme in the proposed network and apply ERM in the multiple supervision layers. Inspired by gradient-based edge detection, we can generate an edge map from the prediction map in each ERM. And we utilize a simple but effective operation to decouple the edge map into a foreground edge map and a background edge map. The decoupled edge maps are then used to suppress background features and enhance vessel features in subsequent layers. Therefore, the ultimate prediction map can be refined progressively to improve the segmentation effect.

In summary, the main contributions of this work are as follows:

1. Cross-scale Attention Transformer (CAT) is proposed as the basic unit in the transformer UNet. CAT fully exploits the dense similar structures with different scales in a retinal image and effectively fuses the features to improve the vessel segmentation performance.
2. A progressive edge refinement module (ERM) is designed to refine the edges of foreground and background in the segmentation results. ERM can suppress non-vessel features and enhance vessel features, and then progressively refine the prediction mask.
3. Comprehensive experiments and discussions are carried out to verify the effectiveness of the proposed method. The vessel segmentation performance outperforms the state-of-the-art methods on public retinal datasets.

The rest of this paper is as follows. We detail our methods in Section 2 and evaluate the experimental results in Section 3. In Section 4, we further discuss our innovations. Finally, the conclusion is arranged in Section 5.

2. Materials and Methods

In this section, we first overview our framework and then expatiate the proposed cross-scale attention transformer (CAT) and progressive edge refinement module (ERM). Finally, the detailed configurations of the network are given.

2.1. Framework Overview

The diagram of our method is shown in Figure 1. We denote the input image size as $H \times W$. As the patch embedding [22] with size 4×4 in transformer UNet downsamples the image size to $\frac{H}{4} \times \frac{W}{4}$, directly using a $4\times$ upsampling operator in the decoder will lose a lot of shallow features [29]. So we first perform two layers of CBR blocks (i.e., Convolution, Batch Normalization, and ReLU activation) and one downsampling layer to obtain the low-level features $F_1 \in R^{H\times W\times 32}$ and $F_2 \in R^{\frac{H}{2}\times\frac{W}{2}\times 64}$ (32 and 64 are the feature dimensions). In this way, we can utilize CBR blocks also in the decoder to gradually generate the prediction maps.

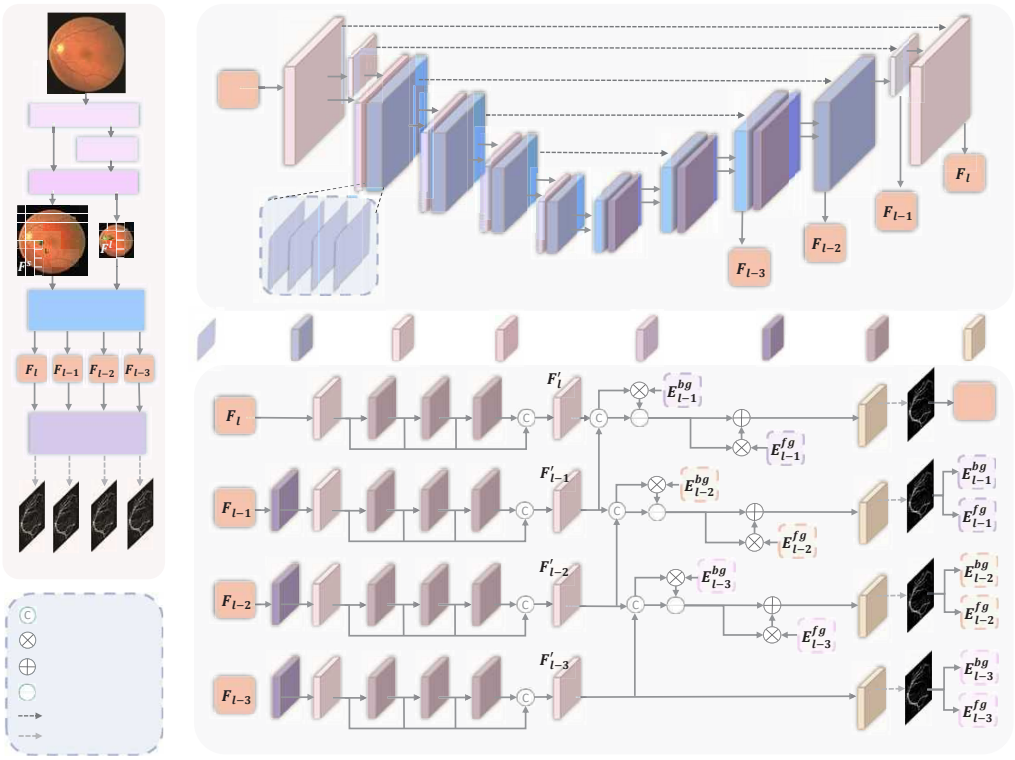


Figure 1. Overview of the proposed method. (**a**) The detailed network structure, mainly used for the visualization of cross-scale feature construction. (**b**) The architecture of transformer UNet, in which Cross-scale Attention Transformer (CAT) serves as the basic unit. (**c**) Illustration of the Edge Refinement Module (ERM), which integrates with the deep-supervision scheme.

For the multi-scale features construction, we find using the outputs of the former CBR layers can smooth the transition from convolutional layers to transformer layers. Therefore, we adopt patch embedding with size 4×4 on both the features F_1 and F_2 to produce different scale features F^s and F^l (the index is omitted here for a better representation). And we visualize them in the detailed network structure of Figure 1a for intuitive understanding. Such two scale features are fed into the subsequent CAT blocks for cross-scale feature learning and interaction.

The proposed transformer UNet framework in Figure 1b is mainly composed of CAT blocks. In each block, there are multiple CAT layers cascaded together. Maxpooling and bilinear interpolation serve as the downsampling and upsampling operations, respectively. The details of CAT are given in a later subsection.

In the decoder, we can obtain the output features before regressing the prediction maps. A deep-supervision scheme is adopted here for better semantic learning. Specifically, the features of the last four layers with sizes of $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{4} \times \frac{W}{4}$, $\frac{H}{2} \times \frac{W}{2}$, and $H \times W$ are selected for multiple supervisions. As shown in Figure 1a, we introduce ERM into the deep-supervision layers to refine the edges in the prediction maps. ERM consumes the prediction map of the higher layer to suppress background features and enhance vessel features to purify each output representation. So the ultimate output can be refined progressively to improve the segmentation effect. The details of ERM are given in a later subsection.

2.2. Cross-Scale Attention Transformer

In this subsection, we will elaborate on the proposed CAT for retinal vessel segmentation. For a clear comparison, we first briefly introduce the native self-attention transformer, which is referred to as Single-scale Attention Transformer (SAT), and Dual-scale Attention transformer (DAT).

SAT is widely utilized in the early transformer UNet frameworks [23,24], which consist of several SAT blocks. In each block, the feature input to i_{th} layer can be denoted as $F_i \in R^{L \times C}$ (C is the feature dimension). SAT is mainly composed of Norm, W-MSA, and MLP with GELU activation. As shown in Figure 2a, SAT can be formalized as

$$\begin{aligned} \hat{F}_i &= W - MSA(Norm(F_i)) + F_i, \\ F_{i+1} &= MLP(Norm(\hat{F}_i)) + \hat{F}_i. \end{aligned} \tag{1}$$

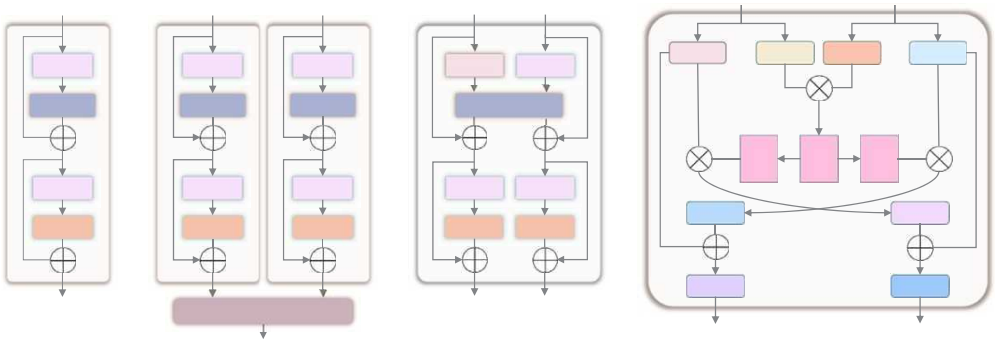


Figure 2. Comparisons of the proposed CAT with other transformer architectures. SAT, DAT, and CAT mean Single-scale Attention Transformer, Dual-scale Attention Transformer, and Cross-scale Attention Transformer, respectively. SAT consists of Layer Normalization (Norm), Window-based Multi-head Self-attention (W-MSA), and MLP with GELU activation. DAT consists of two SAT layers and a Cross-Attention Fusion module. CAT is similar to DAT but introduces a newly designed Window-based Multi-head Cross-scale Attention (W-MCSA) to replace W-MSA and Cross-Attention Fusion module.

Based on the multi-scale idea, Lin et al. [29] propose DAT to extract the coarse and fine-grained feature representations of different semantic scales. They first perform patch embedding with two sizes to yield different scale features, which can be denoted as $F^s \in R^{L^s \times C}$ (small-scale) and $F^l \in R^{L^l \times C}$ (large-scale). They further apply two naive transformer encoders on the features, i.e., F^s and F^l are fed into different SAT layers, respectively. And a cross-attention fusion (CAF) module is followed to aggregate the features. As shown in Figure 2b, DAT can be written as

$$F_{i+1} = CAF(SAT(F_i^s), SAT(F_i^L)). \tag{2}$$

According to previous observation that there are dense similar structures with different scales in retinal images, multi-scale learning is a natural idea to exploit this characteristic.

DAT is a simple and feasible scheme for cross-scale learning. However, the scheme that only performing cross-attention after independent single-scale attention is too complex to efficiently integrate features of different scales, resulting in the inability to fully benefit from similar structures. Therefore, we propose CAT in the transformer UNet framework for retinal vessel segmentation. CAT also consumes features F^s and F^l but achieves attentional feature aggregation in each transformer layer. This is possible due to the newly designed W-MCSA, which is utilized to replace the W-MSA in SAT for cross-scale feature interaction. As shown in Figure 2c, CAT can be written as

$$\begin{aligned} \hat{F}_i^s, \hat{F}_i^l &= W - MCSA(Norm(F_i^s), Norm(F_i^l)), \\ F_{i+1}^s &= MLP(Norm(\hat{F}_i^s)) + \hat{F}_i^s, \\ F_{i+1}^l &= MLP(Norm(\hat{F}_i^l)) + \hat{F}_i^l. \end{aligned} \tag{3}$$

Apparently, CAT can conduct cross-scale attention interactions and produce fused dual-scale features F_{i+1}^s and F_{i+1}^l in only one transformer layer. So we can cascade multiple CAT layers to gradually extract high-dimensional representations to take full advantage of similar structure information at different scales.

The details of W-MCSA are also shown in Figure 2d. For the input dual-scale features $F_{in}^s \in R^{L^s \times C}$ and $F_{in}^l \in R^{L^l \times C}$, the traditional way is applying several weight matrices $W \in R^{C \times D}$ to transform F_{in}^s and F_{in}^l to query, key and value features, respectively. In our W-MCSA, it only calculates the query and value vectors of the dual-scale features. The form can be written as

$$\begin{aligned} Q_s, V_s &= F_{in}^s W_Q^s, F_{in}^s W_V^s, \\ Q_l, V_l &= F_{in}^l W_Q^l, F_{in}^l W_V^l. \end{aligned} \tag{4}$$

Then we skillfully utilize a single attention matrix $A \in L^s \times L^l$ to formulate the cross-scale feature similarity. The form can be written as

$$A = \frac{Q_s {Q_l}^\top}{\sqrt{D}} + RPB, \tag{5}$$

where RPB means relative position bias proposed in [24]. Then we perform Softmax normalization along different dimensions of A to generate attention weights of F_{in}^s to F_{in}^l or F_{in}^l to F_{in}^s. Therefore, the calculation of key features can be economized but efficient cross-attention is also realized. The attentional features can be formalized as

$$\begin{aligned} V_s' &= Softmax(A, 1) V_l + V_s, \\ V_l' &= Softmax(A, 0)^\top V_s + V_l, \end{aligned} \tag{6}$$

where the numbers in Softmax mean the dimensions for normalization. And we also include their respective value features to keep feature consistency. Finally, projection matrices are applied to produce the interacted features F_{out}^s and F_{out}^l.

It is worth noting that the above cross-attention is applied in a window to balance the receptive field and efficiency, and the attention calculation adopts the multi-head scheme. The architectures of W-MCSA and CAT are ingenious and such design cleverly combines the dual-scale features and attention mechanism. So compared to SAT and DAT, the proposed CAT is more robust and efficient in fusing cross-scale similar structure information. We will discuss the impacts in Section 3.5.

2.3. Edge Refinement Module

In this section, we will introduce the proposed ERM for edge refinement. As the fine-grained foreground of the vessel mask, there is typically misclassifications in the edges of the segmentation results. Based on the idea of edge decoupling, we further propose ERM integrated with the deep-supervision scheme.

As shown in Figure 1c, we select the last four layers in the decoder for multiple supervisions. The features are $F_{l-3} \in R^{\frac{H}{8} \times \frac{W}{8} \times 8C}$, $F_{l-2} \in R^{\frac{H}{4} \times \frac{W}{4} \times 4C}$, $F_{l-1} \in R^{\frac{H}{2} \times \frac{W}{2} \times 2C}$ and $F_l \in R^{H \times W \times C}$, respectively, and we upsample them to the size of $H \times W$. In each layer, several blocks of strip convolutions, e.g., horizontal, vertical, left diagonal, and right diagonal, are applied to extract shallow morphology features for better vessel prediction. And dense connections are performed to aggregate these features to enrich the representations. The connected and transformed features can be denoted as $F'_{l-3}, F'_{l-2}, F'_{l-1}, F'_l \in R^{H \times W \times C}$. Edge refinement is expected before regressing feature F' to prediction map $M \in R^{H \times W}$ (index is omitted here).

Therefore, for the feature F'_l of the l_{th} layer, ERM first generates a soft edge map $E_{l-1} \in R^{H \times W}$ based on the prediction map M_{l-1} of the adjacent layer. This operation is implemented through gradient calculation in a 3×3 kernel inspired by edge detection. By analyzing the edge map E_{l-1}, we find that the salient pixels in E_{l-1} include not only vessel pixels but also background pixels. Thus, we further design a simple but effective approach to excavating the edge map. Specifically, we decouple the edge map E_{l-1} to a foreground edge map E^{fg}_{l-1} and a background edge map E^{bg}_{l-1} through element-wise multiplication based on M_{l-1}

$$\begin{aligned} E^{fg}_{l-1} &= M_{l-1} \cdot E_{l-1}, \\ E^{bg}_{l-1} &= (1 - M_{l-1}) \cdot E_{l-1}. \end{aligned} \tag{7}$$

Such that E^{fg}_{l-1} expresses the vessel pixels near the background, and E^{bg}_{l-1} expresses the non-vessel pixels near the foreground. An illustration is shown in Figure 3 to clarify the decoupled edge maps. By doing this, ERM can perform feature repressing and enhancement accordingly. The form can be expressed as

$$\begin{aligned} \hat{F}'_l &= BR(Cat(F'_l, F'_{l-1})(1 - E^{bg}_{l-1})), \\ F''_l &= BR(\hat{F}'_l(1 + E^{fg}_{l-1})), \end{aligned} \tag{8}$$

where F''_l is the refined feature for subsequent vessel prediction, BR means Batch Normalization and ReLU activation. We also concatenate the feature F'_{l-1} of the adjacent layer. After these attentive operations, ERM suppresses the unexpected background edge features by element-wise subtraction and enhances the foreground edge features by element-wise addition. The module is cascaded to the adjacent layers in the deep-supervision. So the ultimate prediction map can be refined progressively to improve the segmentation result.

2.4. Network Details

The proposed CAT and ERM are integrated into the transformer UNet framework. We implement the network with two CBR blocks and four CAT blocks for feature extraction in different resolutions. Each CAT block is composed of four CAT layers. The window size of W-MCSA in CAT is set to 32×32 to balance the receptive field and efficiency. As dual-scale features are produced in the last CAT layer of the decoder, we upsample them to the same resolution and concatenate them as the input of the latter CNN layers. The last four layers are conducted for deep supervision with ERM, and the final output prediction map serves as the vessel segmentation result in the inference stage.

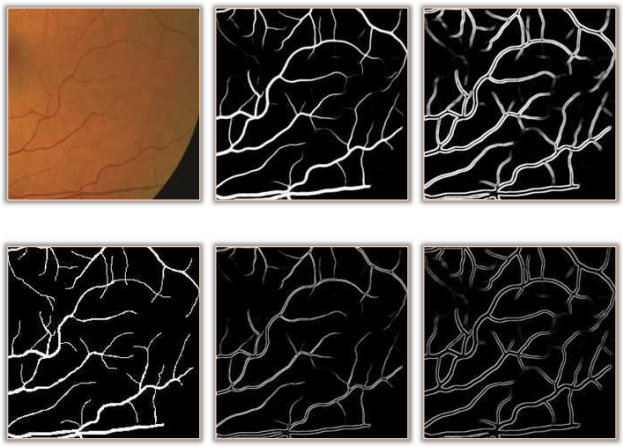


Figure 3. An illustration to represent edge decoupling on the prediction map M. E, E^{fg}, E^{bg} are the edge map, foreground edge map, and background edge map, respectively. Given an input image patch, the proposed network will generate the prediction maps for multiple supervision. Based on the prediction map M of the adjacent layer, ERM first generates a soft edge map E. After that, the edge map E is decoupled into the foreground edge map E^{fg} and the background edge map E^{bg}. E^{fg} expresses the vessel pixels near the background, and E^{bg} expresses the non-vessel pixels near the foreground.

3. Results

In this section, we perform comprehensive experiments on public datasets to verify the effectiveness of our method. Then quantitative and qualitative results are presented to evaluate the segmentation performance of our framework. Finally, we further explore the effect of our innovations through an ablation study.

3.1. Datasets

This work is evaluated on two widely-used datasets, including DRIVE [30] and CHASE_DB1 [31]. DRIVE dataset has 40 color retinal images, including seven abnormal pathology cases. And it is fixedly split into two sets for training and testing, respectively. The training set contains only one manual segmentation of an ophthalmologist, while two manual annotations have been applied by two different observers in the testing set. Following many existing methods [32–34], the manual annotation of the first observer is used as the ground truth for training and evaluation. The spatial resolution of each image in the dataset is 584×565. And we uniformly crop the training images into patches with a size of 256×256. In addition, the dataset also provides the Field Of View (FOV) masks.

The CHASE_DB1 dataset contains 28 color fundus images from both the left and right eyes of 14 children. To be consistent with previous methods, the first 20 images are used for training and the remaining are for testing. Each image has different manual annotations from two experts. For a fair comparison with other methods, manual annotations of the first expert are taken as ground truth in the experiments. The spatial resolution of each image is 999×960. Due to the high resolution of the original images in the dataset, we crop them into patches with a size of 384×384 as the network input. The FOV masks of CHASE_DB1 dataset are not provided, we generate them by a simple threshold filtering according to Boudegga et al. [35].

3.2. Implementation Details

Regarding the data preprocessing, we first convert color fundus images to gray images. Then we normalize the gray images to a standard normal distribution and re-scale them to image space. In addition, Contrast Limited Adaptive Histogram Equalization (CLAHE) [36] and gamma correction are also employed to enhance image contrast. Finally, we perform

data augmentation, e.g., image flip, random rotation, and random zoom, to improve the robustness and prevent overfitting during training.

We adopt the PyTorch framework to conduct all the experiments on a PC with Intel Xeon E5 CPU, 64GB RAM, and an NVIDIA RTX 3090 GPU. During the training, the initial learning rate is set to 0.0005 for both DRIVE and CHASE_DB1. To dynamically adjust the training process, we utilize the multi-step decay strategy to update the learning rate. The batch size is set to 2, and the maximal epoch is 150. Moreover, we use the Adam algorithm as the training optimizer.

The loss function of our network is the class-weighted binary cross-entropy loss $\mathcal{L}_{BCE}$, which is defined as

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{j=0}^{N}(y_j log y'_j + (1-y_j) log(1-y'_j)) \tag{9}$$

where N denotes the number of pixels of each image. y denotes the ground truth label, and y' represent the prediction value. Class weights of 1.5 and 1 are found appropriate for vessel and non-vessel in our experiments. Since we employ a deep supervision scheme in the last four layers, $\mathcal{L}_{total}$ can be written as:

$$\mathcal{L}_{total} = \mathcal{L}(G, M_{l-3}) + \mathcal{L}(G, M_{l-2}) + \mathcal{L}(G, M_{l-1}) + \mathcal{L}(G, M_l) \tag{10}$$

where $\mathcal{L}$ denotes the BCE loss $\mathcal{L}_{BCE}$, G is the ground truth mask, and M_k is the prediction map of k_{th} layer.

3.3. Evaluation Metrics

To quantitatively compare with the state-of-the-art methods, we use the standard evaluation metrics, including Accuracy (Acc), Dice, the area under curve (AUC), Sensitivity (Se), and Specificity (Sp). Acc measures the overall performance of the segmentation results. Dice (also called F1-score) is to evaluate the network performance in an imbalance binary segmentation task. For binary segmentation of retinal vessels, positive and negative are classified as vessel pixels and background pixels, respectively. Therefore, Se and Sp represent the segmentation accuracy of the vessel and background pixels, respectively. Furthermore, the receiver operating characteristic curve (ROC) reflects the trade-off between sensitivity and specificity, and the area under the curve (AUC) is used to comprehensively assess the capability of network classification.

3.4. Results and Comparisons

Compared Methods: To demonstrate the superiority of the proposed method, we collect many state-of-the-art approaches as competitors, including CNN-based methods (e.g., SA-UNet [14], CTF-Net [33]) and transformer-based methods (e.g., GT U-Net [27], GT-DLA-dsHFF [37]). Moreover, we also compare our network against two methods based on the multi-scale features (i.e., Dual E-Unet [38] and SCS-Net [39]), and two edge-aware methods (i.e., BEFD-UNet [40], DE-DCGCN-EE [41]). For SA-UNet [14], GT U-Net [27], Iter-Net [32], CTF-Net [33], CAR-UNet [34], and Genetic U-Net [42], we generate their segmentation results based on the released codes by the authors. And the results of other compared methods are directly from the corresponding papers.

Quantitative Comparisons: Tables 1 and 2 report the quantitative results of our network and compared methods. For the DRIVE dataset, our quantitative results of Acc, Dice, AUC, Se, and Sp are 97.05%, 83.33%, 98.87%, 84.68%, and 98.26%, respectively. Compared with all other methods, the proposed model achieves the best results in Dice, AUC and Se, while producing a comparable score in other metrics. Specifically, SGL [43] achieves the highest Se among the competitors, but our method is still 0.88% higher than it. And our result is also 0.17% higher than it in Dice, while maintaining almost the same Acc, AUC, and Sp. In terms of the metric Sp, although CAR-UNet [34] achieves the highest score among all the compared methods, its other metrics are all lower than our method, especially

for Dice and Se, which are 0.8% and 3.33% lower, respectively. Overall, our method has great advantages in Dice and Se, which are 0.17–2.06% and 0.88–7.26% higher than other methods, respectively. As described in the definitions of Dice and Se, these improvements indicate that our method is more capable of segmenting vessels in challenging regions. More importantly, our proposed model fulfills such superiority while achieving the best balance among the evaluation metrics compared to the state-of-the-art methods.

Table 1. Quantitative comparisons on the DRIVE dataset. The bold items are our results and the underlined items are the best results.

Method	Year	Acc (%)	Dice (%)	AUC (%)	Se (%)	Sp (%)
Dual E-Unet [38]	2019	95.67	82.70	97.72	79.40	98.16
Iter-Net [32]	2020	95.74	82.18	98.13	77.91	98.31
CTF-Net [33]	2020	95.67	82.41	97.88	78.49	98.13
BEFD-UNet [40]	2020	97.01	82.67	98.67	82.15	98.45
HANet [44]	2020	95.81	82.93	98.23	79.91	98.13
SA-UNet [14]	2021	96.98	82.63	98.64	82.12	98.40
GT U-Net [27]	2021	95.46	81.27	96.96	77.42	98.09
SCS-Net [39]	2021	96.97	81.89	98.37	82.89	98.38
CAR-UNet [34]	2021	96.99	82.53	98.52	81.35	98.49
SGL [43]	2021	97.05	83.16	98.86	83.80	98.34
GT-DLA-dsHFF [37]	2022	97.03	82.57	98.63	83.55	98.27
DE-DCGCN-EE [41]	2022	97.05	82.88	98.66	83.59	98.26
Genetic U-Net [42]	2022	97.07	83.14	98.85	83.00	98.43
Ours	2023	**97.05**	**83.33**	**98.87**	**84.68**	**98.26**

Table 2. Quantitative comparisons on the CHASE_DB1 dataset. The bold items are our results and the underlined items are the best results.

Method	Year	Acc (%)	Dice (%)	AUC (%)	Se (%)	Sp (%)
Dual E-Unet [38]	2019	96.61	80.37	98.12	80.74	98.21
Iter-Net [32]	2020	96.55	80.73	98.51	79.70	98.23
CTF-Net [33]	2020	96.48	82.20	98.47	79.48	98.42
HANet [44]	2020	96.73	81.91	98.81	81.86	98.44
SA-UNet [14]	2021	97.55	81.53	99.05	85.73	98.35
CAR-UNet [34]	2021	97.51	80.98	98.98	84.39	98.39
SCS-Net [39]	2021	97.44	-	98.67	83.65	98.39
GT-DLA-dsHFF [37]	2022	97.60	-	98.92	84.40	98.58
DE-DCGCN-EE [41]	2022	97.62	82.61	98.98	84.00	98.56
Ours	2023	**97.66**	**82.36**	**98.77**	**86.59**	**98.42**

Regarding the CHASE_DB1 dataset in Table 2, we can find that our method achieves the best Acc score of 97.66% and the best Se score of 86.59% among all methods. The most significant improvement is Se, which measures the accuracy of vessel segmentation, outperforming other methods by at least 0.86%. Thus, it means that our model can correctly localize and segment more vessel pixels. Since the vessel region occupied less than 10% of the whole image, Dice which denotes the degree of overlap between the predictions and the ground truth is more susceptible to the background. Although DE-DCGCN-EE [41] obtains the optimal Dice, our method takes the second rank and our Dice score (82.36%) is very close to the best one (82.61%). In contrast, our method has a Se improvement of 2.59% over DE-DCGCN-EE, demonstrating that our model has more ideal performance in the classification of vessel pixels. SA-UNet [14] adopts a spatial attention module for adaptive feature refinement, thereby achieving the highest AUC. Nevertheless, compared

to SA-UNet, our model achieves improvements of 0.86%, 0.83%, 0.11%, 0.07% on Se, Dice, Acc, and Sp.

Visual Comparisons: Apart from the quantitative results, we also present the visual comparisons against other methods in Figures 4 and 5. Specifically, we visualize the retinal vessel segmentation results produced by our network and state-of-the-art methods in terms of two test cases. As shown in Figure 4, the collateral vessels in the results of Iter-Net [32], GT U-Net and Genetic U-Net are seriously missing or misclassified. CTF-Net [33] utilizes deep coarse-to-fine supervision network to refine segmentation and segments more collateral vessels than others. However, there are still broken vessels in CTF-Net. Therefore, the existing methods have difficulty segmenting the collateral vessels in challenging regions. The results are either completely without collateral vessels or with segmented broken vessels. Nevertheless, our predicted segmentation map has the same complete collateral vessels as the Ground Truth. As visualized in Figure 5, Iter-Net, CTF-Net, and CAR-UNet have poor segmentation performance at collateral vessels or the intersection of vessels. Although SA-UNet obtains better vessel probability in the prediction map than Iter-Net, CTF-Net, and CAR-UNet, it still has the problem of misclassification. However, our proposed model can effectively alleviate this limitation. It means that our model can exactly locate and segment more collateral vessels while maintaining the segmentation effect of the main vessels.

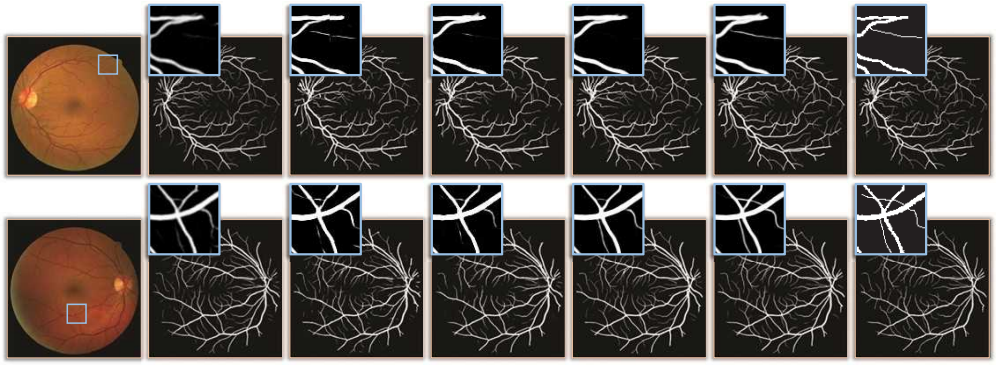


Figure 4. Segmentation results of the DRIVE dataset.

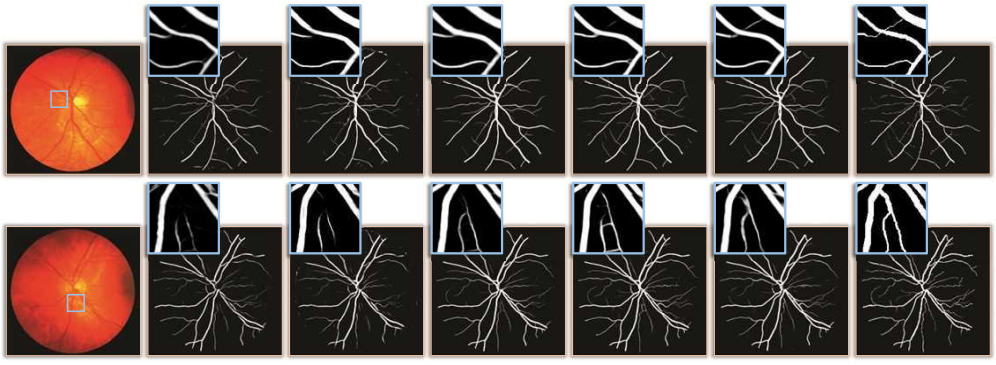


Figure 5. Segmentation results of the CHASE_DB1 dataset.

Parameters and FLOPs: For a fair comparison, we also count the network parameters and the FLOPs when inferring a test retinal image. The results are listed in Table 3. The method utilizing CNN is lighter, with fewer parameters and FLOPs. Transformer networks typically contain more parameters than CNNs. Especially since our method introduces cross-scale learning, we have achieved the maximum number of parameters. However, thanks to the window-based attention structure [25], our network has FLOPs that are lower than the other two transformer-based methods.

Table 3. The network parameters and the FLOPs of the compared methods and ours. The FLOPs are calculated on a 256 × 256 image patch.

Method	Parameters	FLOPs
CAR-UNet [34]	1.05 M	2.10 M
Iter-Net [32]	8.24 M	16.48 M
SA-UNet [14]	538.71 K	1.07 G
CTF-Net [33]	1.38 M	3.43 G
Genetic U-Net [42]	272.42 K	8.23 G
GT U-Net [27]	25.89 M	55.32 G
DE-DCGCN-EE [41]	14.11 M	73.62 G
GT-DLA-dsHFF [37]	26.09 M	118.62 G
Ours	291.56 M	71.15 G

3.5. Ablation Study

To further verify the effectiveness of major components of our network, we perform ablation experiments on the DRIVE dataset. To do so, we first take a U-shaped framework (denoted as transformer UNet) based on the transformer block as the basic architecture in this experiment. Then, SAT, DAT, and CAT are respectively used as the transformer block to compare their segmentation capabilities. Then we add ERM to these three variants to evaluate the effect of ERM. The experimental results have been summarized in Table 4. And the visual comparisons of the segmentation results are presented in Figure 6.

Table 4. Ablation results on DRIVE dataset. The underlined items are the best results.

Framework	Transformer Block			ERM	Metrics				
	SAT	DAT	CAT		Acc (%)	Dice (%)	AUC (%)	Se (%)	Sp (%)
Transformer UNet	✓				96.90	82.18	98.60	82.09	98.32
		✓			96.93	82.44	98.66	82.86	98.30
			✓		97.01	82.99	98.79	83.73	98.30
	✓			✓	97.00	82.74	98.74	82.50	98.31
		✓		✓	97.01	82.91	98.77	83.39	<u>98.34</u>
			✓	✓	<u>97.05</u>	<u>83.33</u>	<u>98.87</u>	<u>84.68</u>	98.26

As shown in Table 4, compared to SAT and DAT, CAT achieves 97.01% on Acc, 82.99% on Dice, 98.79% on AUC, and 83.73% on Se, which outperforms SAT by an improvement of 0.11%, 0.81%, 0.19%, and 1.64%, respectively. The most remarkable improvement is Se, indicating that the CAT can significantly improve the sensitivity of the network on multi-scale features. Although the Sp of 98.30% in CAT is lower, the disadvantage is negligible. And in Figure 6c, we can further compare and analyze the difference between the segmentation results of CAT and SAT through the red part. We find that the segmentation effect of main vessels in CAT is more complete than in SAT, while more collateral vessels are segmented. Furthermore, CAT is 0.08%, 0.55%, 0.13%, and 0.87% higher than DAT in terms of Acc, Dice, AUC, and Se, respectively. The overall improvement means that compared with DAT, the cross-scale scheme designed in CAT can efficiently fuse the features with similar structures to improve vessel segmentation performance in challenging regions. As visualized in Figure 6e, the patches at the top and the bottom prove that CAT is able to better identify collateral vessels or microvessels in challenging regions. In conclusion, CAT can segment more collateral vessels or microvessels than SAT and DAT in challenging regions while maintaining the segmentation effect of the main vessels.

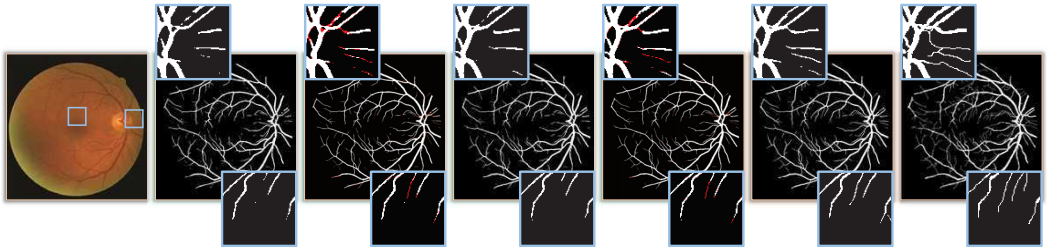


Figure 6. Segmentation results of different transformer blocks. SAT-CAT and DAT-CAT mark the differences between their results and ours. The differences are highlighted in red.

To explore the effect of ERM, we conduct the experiments with three different configurations, which are the SAT with ERM, the DAT with ERM, and the CAT with ERM. It can be observed in Table 4 that SAT with ERM achieves improvements of 0.1%, 0.56%, 0.14%, 0.41% on Acc, Dice, AUC, and Se compared to SAT while showing a comparable score on Sp. After adding ERM based on the previous experiment of DAT variation, all of the five performance indicators have been increased. The overall improvement in performance metrics proves that the progressive refinement of the prediction map is beneficial to alleviate the edge-blurring problem. In particular, CAT with ERM produces SOTA results on almost all metrics. Specifically, it outperforms CAT by an improvement of 0.04%, 0.34%, 0.08%, and 0.95% in terms of Acc, Dice, and Se. Although it is not as good as CAT in Sp, it still produces close and comparable scores. More importantly, the performance of three experiments of SAT, DAT, and CAT are improved after adding ERM respectively. It demonstrates that the proposed ERM can provide supplementary information for high-level features, thereby further promoting retinal vessel segmentation.

4. Discussion

4.1. The Effect of Input Patch Size

It is a common operation to crop the input image into square patches in a transformer network. A large patch size leads to faster inference time but more memory consumption. We adjust the input patch to multiple sizes to explore the performance of our method under different hardware and demand scenarios. Specifically, we consider five different patch sizes, and they are 384×384, 256×256, 192×192, 128×128, and 96×96. The results on DRIVE dataset are shown in Figure 7.

We can see that the results of Acc and Sp are almost unchanged for different input patch sizes on the DRIVE dataset. As the patch size becomes larger, Dice and Se gradually increase. The reason behind this is that small patches ignore rich cross-scale similar structures. For the Se metric, the highest score is obtained when the patch size is 256×256. Therefore, we empirically select 256×256 as the input patch size for the DRIVE dataset.

4.2. The Effect of CAT Layers

In order to verify the ability of cross-scale learning of the proposed CAT, we conduct experiment with different numbers of CAT layers in each transformer block. Specifically, our network has four transformer blocks in the encoder, and we consider four settings on the number of CAT layers in the transformer blocks. The four settings are [4, 4, 8, 8], [4, 4, 4, 4], [2, 2, 4, 4], and [2, 2, 2, 2].

Table 5 reports the quantitative results of our network with different numbers of CAT layers in the transformer blocks. Apparently, the number of network parameters and the training time is increased when we use a large number of CAT layers. From these quantitative results, we can find that the setting with [4, 4, 4, 4] has the largest Acc, Dice, and Se scores, and its Sp score (98.26%) is also close to the best one (98.27%). Hence, we empirically set the numbers of CAT layers in the transformer blocks as [4, 4, 4, 4].

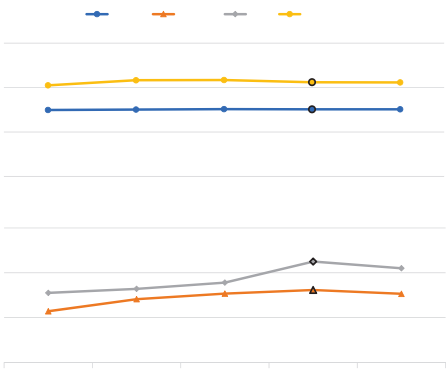


Figure 7. The results of five patch sizes for training and testing. The horizontal axis means the side length of a square image patch. When the patch size is set to 256 × 256, the best overall results are obtained.

Table 5. Comparisons with different numbers of CAT layers. The parameters and training time are also provided. The underlined items are the best results.

CAT Layers	Param.	Acc (%)	Dice (%)	Se (%)	Sp (%)	Training Time
[4, 4, 8, 8]	522.54 M	97.04	83.28	84.45	<u>98.27</u>	10.13 h
[4, 4, 4, 4]	291.56 M	<u>97.05</u>	<u>83.33</u>	<u>84.68</u>	98.26	7.78 h
[2, 2, 4, 4]	284.32 M	97.03	83.23	84.51	98.26	6.93 h
[2, 2, 2, 2]	168.83 M	97.03	83.20	84.41	98.26	6.03 h

4.3. The Effect of Deep-Supervision Layers

Deep-supervision schemes [45] have been widely used to boost the training of intermediate layers and mitigate gradient disappearance. Motivated by this, we adopt the deep-supervision scheme and further integrate ERM into the multiple supervision layers to progressively refine the prediction map. To evaluate the performance of ERM more thoroughly, we adjust the number of deep-supervision layers for training to analyze the effects. Note that the number of ERMs will also change when selecting a different number of deep-supervision layers.

As shown in Figure 8, there are no obvious performance differences in terms of the Acc, Dice, and Sp scores when the numbers of deep supervision layers are three, four, five, and six. Only for the Se metric, we find that the best effect is obtained when the number of supervision layers is four. Hence, we empirically design four deep-supervision layers with ERMs in our method.

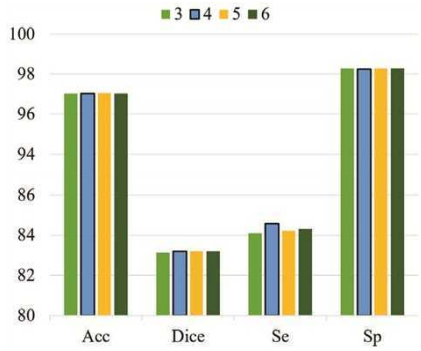


Figure 8. Comparisons with different deep-supervision layers. The best performance is observed when the number of deep-supervision layers is four.

5. Conclusions

The method proposed in this paper has two major innovative components: Cross-scale Attention Transformer (CAT) and progressive Edge Refinement Module (ERM). CAT explores a novel cross-scale attention mechanism in transformer UNet to integrate multi-scale features with similar structures in a retinal image. ERM is designed to decouple the edge map to enhance vessel features and suppress background features in multiple supervision layers. These two novel structures encourage the network to enhance the cross-scale learning ability and refine the vessel edges progressively. Experimental results on two public datasets have shown the effectiveness of the proposed method and the superior segmentation performance on collateral vessels.

Author Contributions: Conceptualization, Y.Y.; methodology, software and visualization, Y.Y. and Y.Z.; validation, Y.Y., Y.Z. and L.Z.; investigation, Y.Q.; resources, L.C.; writing—original draft preparation, Y.Y.; writing—review and editing, Y.Z. and L.Z.; supervision and funding acquisition, L.C. and Y.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program under grant number 2022YFB4703500; National Natural Science Foundation of China (NSFC) General Project grant number 62072452; Shenzhen Science and Technology Program grant number JCYJ20200109115627045 and JCYJ20220818101401003; International Scientific and Technological Cooperation Foundation of Shenzhen, GJHZ20200731095005016; and in part by the Regional Joint Fund of Guangdong grant number 2021B1515120011.

Data Availability Statement: The data presented in this study are openly available in [DRIVE] at [10.1109/TMI.2004.825627] [30], and [CHASE_DB1] at [10.1109/TBME.2012.2205687] [31]. The code of this work is released on https://github.com/Yuanggyy/Retinal-vessel-segmentation (accessed on 13 December 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wong, T.Y.; Mitchell, P. Hypertensive retinopathy. *N. Engl. J. Med.* **2004**, *351*, 2310–2317. [CrossRef] [PubMed]
2. Patton, N.; Aslam, T.; MacGillivray, T.; Pattie, A.; Deary, I.J.; Dhillon, B. Retinal vascular image analysis as a potential screening tool for cerebrovascular disease: A rationale based on homology between cerebral and retinal microvasculatures. *J. Anat.* **2005**, *206*, 319–348. [CrossRef] [PubMed]
3. Zhu, C.; Zou, B.; Zhao, R.; Cui, J.; Duan, X.; Chen, Z.; Liang, Y. Retinal vessel segmentation in colour fundus images using extreme learning machine. *Comput. Med. Imaging Graph.* **2017**, *55*, 68–77. [CrossRef] [PubMed]
4. Mendonca, A.; Campilho, A. Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *IEEE Trans. Med. Imaging* **2006**, *25*, 1200–1213. [CrossRef]
5. Zhang, J.; Dashtbozorg, B.; Bekkers, E.; Pluim, J.P.W.; Duits, R.; ter Haar Romeny, B.M. Robust retinal vessel segmentation via locally adaptive derivative frames in orientation scores. *IEEE Trans. Med. Imaging* **2016**, *35*, 2631–2644. [CrossRef]
6. Al-Diri, B.; Hunter, A.; Steel, D. An active contour model for segmenting and measuring retinal vessels. *IEEE Trans. Med. Imaging* **2009**, *28*, 1488–1497. [CrossRef]
7. Yang, J.; Lou, C.; Fu, J.; Feng, C. Vessel segmentation using multiscale vessel enhancement and a region based level set model. *Comput. Med. Imaging Graph.* **2020**, *85*, 101783. [CrossRef]
8. Yin, Y.; Adel, M.; Bourennane, S. Retinal vessel segmentation using a probabilistic tracking method. *Pattern Recognit.* **2012**, *45*, 1235–1244. [CrossRef]
9. Roychowdhury, S.; Koozekanani, D.D.; Parhi, K.K. Iterative vessel segmentation of fundus images. *IEEE Trans. Biomed. Eng.* **2015**, *62*, 1738–1749. [CrossRef]
10. Ricci, E.; Perfetti, R. Retinal blood vessel segmentation using line operators and support vector classification. *IEEE Trans. Med. Imaging* **2007**, *26*, 1357–1365. [CrossRef]
11. Lupascu, C.A.; Tegolo, D.; Trucco, E. FABC: Retinal vessel segmentation using AdaBoost. *IEEE Trans. Inf. Technol. Biomed.* **2010**, *14*, 1267–1274. [CrossRef]
12. Alom, M.Z.; Yakopcic, C.; Hasan, M.; Taha, T.M.; Asari, V.K. Recurrent residual U-Net for medical image segmentation. *J. Med. Imaging* **2019**, *6*, 014006. [CrossRef] [PubMed]
13. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.

14. Guo, C.; Szemenyei, M.; Yi, Y.; Wang, W.; Chen, B.; Fan, C. Sa-unet: Spatial attention u-net for retinal vessel segmentation. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1236–1242.
15. Li, X.; Jiang, Y.; Li, M.; Yin, S. Lightweight attention convolutional neural network for retinal vessel image segmentation. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1958–1967. [CrossRef]
16. Wang, C.; Xu, R.; Zhang, Y.; Xu, S.; Zhang, X. Retinal vessel segmentation via context guide attention net with joint hard sample mining strategy. In Proceedings of the 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), Nice, France, 13–16 April 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1319–1323.
17. Mou, L.; Zhao, Y.; Fu, H.; Liu, Y.; Cheng, J.; Zheng, Y.; Su, P.; Yang, J.; Chen, L.; Frangi, A.F.; et al. CS2-Net: Deep learning segmentation of curvilinear structures in medical imaging. *Med. Image Anal.* **2021**, *67*, 101874. [CrossRef] [PubMed]
18. Mou, L.; Chen, L.; Cheng, J.; Gu, Z.; Zhao, Y.; Liu, J. Dense dilated network with probability regularized walk for vessel detection. *IEEE Trans. Med. Imaging* **2019**, *39*, 1392–1403. [CrossRef] [PubMed]
19. Nazir, N.; Sarwar, A.; Saini, B.S.; Shams, R. A robust deep learning approach for accurate segmentation of cytoplasm and nucleus in noisy pap smear images. *Computation* **2023**, *11*, 195. [CrossRef]
20. Araújo, R.J.; Cardoso, J.S.; Oliveira, H.P. A deep learning design for improving topology coherence in blood vessel segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Shenzhen, China, 13–17 October 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 93–101.
21. Shin, S.Y.; Lee, S.; Yun, I.D.; Lee, K.M. Deep vessel segmentation by learning graphical connectivity. *Med. Image Anal.* **2019**, *58*, 101556. [CrossRef]
22. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
23. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv* **2021**, arXiv:2102.04306.
24. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
25. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-unet: Unet-like pure transformer for medical image segmentation. *arXiv* **2021**, arXiv:2105.05537.
26. Huang, S.; Li, J.; Xiao, Y.; Shen, N.; Xu, T. RTNet: Relation transformer network for diabetic retinopathy multi-lesion segmentation. *IEEE Trans. Med. Imaging* **2022**, *41*, 1596–1607. [CrossRef] [PubMed]
27. Li, Y.; Wang, S.; Wang, J.; Zeng, G.; Liu, W.; Zhang, Q.; Jin, Q.; Wang, Y. Gt u-net: A u-net like group transformer network for tooth root segmentation. In Proceedings of the International Workshop on Machine Learning in Medical Imaging, Strasbourg, France, 27 September 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 386–395.
28. Shen, X.; Xu, J.; Jia, H.; Fan, P.; Dong, F.; Yu, B.; Ren, S. Self-attentional microvessel segmentation via squeeze-excitation transformer Unet. *Comput. Med. Imaging Graph.* **2022**, *97*, 102055. [CrossRef] [PubMed]
29. Lin, A.; Chen, B.; Xu, J.; Zhang, Z.; Lu, G.; Zhang, D. Ds-transunet: Dual swin transformer u-net for medical image segmentation. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–15. [CrossRef]
30. Staal, J.; Abràmoff, M.D.; Niemeijer, M.; Viergever, M.A.; Van Ginneken, B. Ridge-based vessel segmentation in color images of the retina. *IEEE Trans. Med. Imaging* **2004**, *23*, 501–509. [CrossRef] [PubMed]
31. Fraz, M.M.; Remagnino, P.; Hoppe, A.; Uyyanonvara, B.; Rudnicka, A.R.; Owen, C.G.; Barman, S.A. An ensemble classification-based approach applied to retinal blood vessel segmentation. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 2538–2548. [CrossRef] [PubMed]
32. Li, L.; Verma, M.; Nakashima, Y.; Nagahara, H.; Kawasaki, R. Iternet: Retinal image segmentation utilizing structural redundancy in vessel networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 3656–3665.
33. Wang, K.; Zhang, X.; Huang, S.; Wang, Q.; Chen, F. Ctf-net: Retinal vessel segmentation via deep coarse-to-fine supervision network. In Proceedings of the 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), Iowa City, IA, USA, 3–7 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1237–1241.
34. Guo, C.; Szemenyei, M.; Hu, Y.; Wang, W.; Zhou, W.; Yi, Y. Channel attention residual u-net for retinal vessel segmentation. In Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1185–1189.
35. Boudegga, H.; Elloumi, Y.; Akil, M.; Bedoui, M.H.; Kachouri, R.; Abdallah, A.B. Fast and efficient retinal blood vessel segmentation method based on deep learning network. *Comput. Med. Imaging Graph.* **2021**, *90*, 101902. [CrossRef]
36. Setiawan, A.W.; Mengko, T.R.; Santoso, O.S.; Suksmono, A.B. Color retinal image enhancement using CLAHE. In Proceedings of the International Conference on ICT for Smart Society, Orlando, FL, USA, 10–12 October 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–3.
37. Li, Y.; Zhang, Y.; Liu, J.Y.; Wang, K.; Zhang, K.; Zhang, G.S.; Liao, X.F.; Yang, G. Global transformer and dual local attention network via deep-shallow hierarchical feature fusion for retinal vessel segmentation. *IEEE Trans. Cybern.* **2022**, *53*, 5826–5839. [CrossRef]

38. Wang, B.; Qiu, S.; He, H. Dual encoding u-net for retinal vessel segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Shenzhen, China, 13–17 October 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 84–92.
39. Wu, H.; Wang, W.; Zhong, J.; Lei, B.; Wen, Z.; Qin, J. Scs-net: A scale and context sensitive network for retinal vessel segmentation. *Med. Image Anal.* **2021**, *70*, 102025. [CrossRef]
40. Zhang, M.; Yu, F.; Zhao, J.; Zhang, L.; Li, Q. BEFD: Boundary enhancement and feature denoising for vessel segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Lima, Peru, 4–8 October 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 775–785.
41. Li, Y.; Zhang, Y.; Cui, W.; Lei, B.; Kuang, X.; Zhang, T. Dual encoder-based dynamic-channel graph convolutional network with edge enhancement for retinal vessel segmentation. *IEEE Trans. Med. Imaging* **2022**, *41*, 1975–1989. [CrossRef]
42. Wei, J.; Zhu, G.; Fan, Z.; Liu, J.; Rong, Y.; Mo, J.; Li, W.; Chen, X. Genetic u-net: Automatically designed deep networks for retinal vessel segmentation using a genetic algorithm. *IEEE Trans. Med. Imaging* **2022**, *41*, 292–307. [CrossRef]
43. Zhou, Y.; Yu, H.; Shi, H. Study group learning: Improving retinal vessel segmentation trained with noisy labels. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Strasbourg, France, 27 September–1 October 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 57–67.
44. Wang, D.; Haytham, A.; Pottenburgh, J.; Saeedi, O.; Tao, Y. Hard attention net for automatic retinal vessel segmentation. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 3384–3396. [CrossRef] [PubMed]
45. Lee, C.Y.; Xie, S.; Gallagher, P.; Zhang, Z.; Tu, Z. Deeply-supervised nets. In Proceedings of the Artificial Intelligence and Statistics, PMLR, San Diego, CA, USA, 9–12 May 2015; pp. 562–570.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Synthetic Training Data in AI-Driven Quality Inspection: The Significance of Camera, Lighting, and Noise Parameters

Dominik Schraml [1,2,*] and Gunther Notni [1]

[1] Group for Quality Assurance and Industrial Image Processing, Ilmenau University of Technology, 98639 Ilmenau, Germany; gunther.notni@tu-ilmenau.de
[2] Steinbeis Qualitätssicherung und Bildverarbeitung GmbH, 98693 Ilmenau, Germany
* Correspondence: dominik.schraml@sqb-ilmenau.de; Tel.: +49-3677-4690-5916

Abstract: Industrial-quality inspections, particularly those leveraging AI, require significant amounts of training data. In fields like injection molding, producing a multitude of defective parts for such data poses environmental and financial challenges. Synthetic training data emerge as a potential solution to address these concerns. Although the creation of realistic synthetic 2D images from 3D models of injection-molded parts involves numerous rendering parameters, the current literature on the generation and application of synthetic data in industrial-quality inspection scarcely addresses the impact of these parameters on AI efficacy. In this study, we delve into some of these key parameters, such as camera position, lighting, and computational noise, to gauge their effect on AI performance. By utilizing Blender software, we procedurally introduced the "flash" defect on a 3D model sourced from a CAD file of an injection-molded part. Subsequently, with Blender's Cycles rendering engine, we produced datasets for each parameter variation. These datasets were then used to train a pre-trained EfficientNet-V2 for the binary classification of the "flash" defect. Our results indicate that while noise is less critical, using a range of noise levels in training can benefit model adaptability and efficiency. Variability in camera positioning and lighting conditions was found to be more significant, enhancing model performance even when real-world conditions mirror the controlled synthetic environment. These findings suggest that incorporating diverse lighting and camera dynamics is beneficial for AI applications, regardless of the consistency in real-world operational settings.

Keywords: synthetic data; rendering parameter; AI inspection; quality control; defect detection; blender

1. Introduction

Quality assurance in manufacturing, particularly in injection molding, remains a challenge due to a variety of error types stemming from machine parameters, environmental influences, and batch inconsistencies. As these errors can be expensive to produce in real-world settings, synthetic training data offer a compelling solution for machine learning models tasked with defect detection. The use of synthetic data for deep learning has been expanding in various fields in recent years, yet the impact of rendering parameters on the quality of this data and the subsequent performance of AI models is not well understood.

Recent years have seen an increasing number of publications utilizing synthetic training image data for diverse applications. These range from medical domains, such as Schenkenfelder et al.'s [1] generation of virtual images for burn wound detection on human skin, to civil infrastructure inspection, such as Howells et al.'s [2] work that can be used in assessing earthquake damage, and even space technology, where Viggh et al. [3] used Unity to create synthetic data for spacecraft component detection. In industrial contexts, synthetic data have been employed for tasks including segmenting car parts in the automotive industry [4], training robots for bin-picking chicken fillets in the food industry [5], and detecting surface defects on machine parts [6–8]. The generation of the synthetic images from 3D

geometry is often carried out by using Unity [3,4], and sometimes custom solutions are put into place [7,9], whereas Blender is utilized in the majority of the studies [1,2,5,6,10–19].

A common approach, exemplified by Schmedemann et al. [17], is the randomization of numerous parameters, including lighting, camera position, texture, and the defect itself. This is based on the assumption that a greater extent of domain randomization leads to enhanced domain adaptation and, thus, improved model performance for real images. While it's common to alter at least some of the rendering parameters, particularly focusing on camera position and lighting, beyond this assumption, most studies do not delve deeper into the specific impacts of these variations [1–7,10,12–15,17–19].

Only a few studies have addressed the intricate details of render parameters. Ruediger-Flore et al. [16] explored the impact of realistic material properties and background settings, examining objects rendered at varying levels of detail, from "plain rendered CAD" to "full scene". Their findings highlight that increasing scene realism significantly boosts classification performance, especially when the proper material properties are added. They also found that mixing different levels of realism and expanding the volume of training data further improves performance. However, their analysis did not extend to other critical parameters, such as lighting and camera position. In a different vein, Zhang et al. [20] compared various illumination models, discovering that the choice of illumination model substantially affects network performance. Yet, their study did not investigate whether variation in lighting is better than using any single illumination model, nor did it determine the characteristics required for one lighting model to excel over another.

This study aims to fill the existing knowledge gap by utilizing the Cycles rendering engine to procedurally generate a specific type of defect (flash) on 3D models of injection-molded parts. With this setup for generating synthetic training data, we focus specifically on empirically evaluating the influence of key rendering parameters—lighting, camera position, and noise—on the efficacy of an AI model for quality assessment. Guiding our research are the following key questions:

1. How does variation in lighting affect the accuracy and generalization capabilities of an AI model trained with synthetic data, as compared to using a static light source?
2. In what ways does changing the camera position impact the accuracy and generalization capabilities of an AI model trained with synthetic data, as opposed to a static camera position?
3. How does the comparison of high versus low noise levels, inherent to the ray tracing process in rendering, affect the accuracy and generalization capabilities of an AI model trained with synthetic data?

The remainder of this paper is structured to explore our methodology, present the experimental results, discuss these results, and conclude with insights aimed at informing future directions in synthetic data generation for AI-driven quality inspection.

2. Materials and Methods

In order to create our synthetic 2D image dataset, we use Blender, a versatile piece of open-source 3D graphics software (https://www.blender.org/download/). The integration of the Cycles rendering engine in Blender, which accurately simulates lighting using ray tracing, is essential to the creation of photorealistic images. Blender's advantages also lie in its rich feature set, which includes 3D modeling and sculpting, texturing, UV mapping, and rendering, as well as its powerful Python API that enables the automation of rendering tasks, which is essential for creating large datasets with variable parameters.

Table 1 outlines the variations in the rendering parameters we examined in Blender, each altered systematically to assess their impact on AI model performance. For instance, for the high noise data, a 0.1 noise threshold in Blender was used, a 10-fold increase compared to the low-noise setting. In the table, the Lighting column contrasts a single light source directly above the object with a high dynamic range imaging (HDRI) background for varied lighting. The Camera Position column compares a fixed camera directly above the object to a camera in random positions within a cube above the object, simulating varying

camera positions. The Noise (the noise parameter is linked to ray tracing, where the quantity and path of light rays determine each pixel's color and brightness in the rendered image) column showcases two scenarios: a high noise scenario with a 0.1 threshold and a low noise scenario with a 0.01 threshold.

Table 1. Variations of the analyzed cleaning parameters.

Lighting	Camera Position	Noise
One light source directly above the object	Exactly above the object	0.1 Noise threshold
Via the HDRI background	Random position in cube above the object	0.01 Noise threshold

2.1. Scene Setup

After the 3D modeling of a real injection molded part—a "mortar bowl"—was converted into a .stl geometry, it can be loaded into the blender scene and placed onto a desk geometry. For the material of the desk, we used a freely available asset for a wooden table, meaning the textures for the color, roughness, and normal map are input into the "Principled BSDF" (bidirectional scattering distribution function) shader. For the bowl, we also used the Principled BSDF shader and a Subsurface IOR of 1.46, as well as a Transmission of 0.1 and a Roughness of 0.2 as the properties for plastic material. For the images with variable lighting, a high dynamic range image (HDRI) was used as the background. The scene in Blenders layout view and rendered preview is shown in Figure 1. The reflections of the light sources generated from the HDRI and their physically correct reflection on the tabletop are worth noting, as well as the slight noise in darker areas, such as the right side of the table.

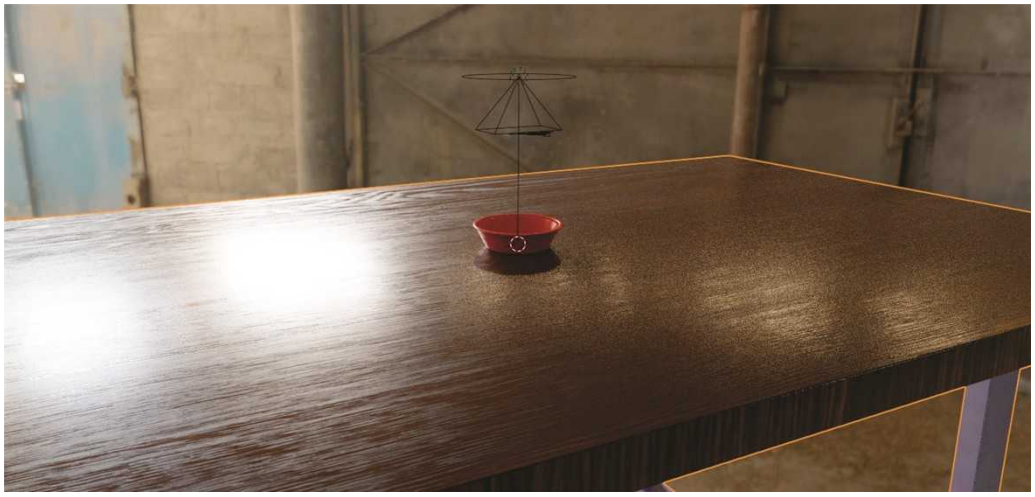


Figure 1. The scene in a rendered preview using ray tracing and a randomly selected HDRI image.

2.2. Defect Creation

The sole focus of our simulation is the "flash" defect, a prevalent issue in plastics production and one of the most common defects in injection-molded parts. Flash is characterized by excess material that extends beyond the intended parting line or mold cavity, with its severity ranging from a barely perceptible thread to a wide rim. This defect is not only widespread but is also of significant concern in the industry. It can result from various factors, including improper mold design, inadequate mold maintenance, excessive injection pressure or speed, and material shrinkage during the molding process.

The defects on the outer edge of the 3D model were created in such a way that they come as close as possible to those of the real component. In order to do this, the nodes on the outer edge of the 3D object's mesh were first added to a collection and then saved (see Figure 2). Each time a part with a defect is rendered, the defect creation algorithm iterates

over all the points in the collection of the original model's geometry, randomly selects a subset of vertices, and applies a perturbation to their position, followed by a smoothing process to ensure a natural transition.



Figure 2. Manipulated vertices highlighted.

The algorithm is parameterized by the 'strength' of perturbation, the 'perturb_fraction', which is the fraction of total vertices to be perturbed, and the 'smoothness', which defines the number of iterations for smoothing the perturbations to create a realistic defect edge. Given a mesh with N selected vertices, the algorithm selects a subset of M vertices to perturb, where M is determined by the perturb fraction:

$$M = \lceil N \cdot \text{perturb_fraction} \rceil. \tag{1}$$

Each vertex, v_i, in this subset is then randomly perturbed by a value, p_i, within the range $[0, \text{strength}]$. In order to ensure a smooth transition and avoid sharp edges, this perturbation is smoothed over smoothness iterations. During each iteration, the perturbation for a vertex v_i is recalculated as the average of its own value and the perturbations of its immediate neighbors, giving the smoothed perturbation p_i^{smooth} as

$$p_i^{\text{smooth}} = \frac{p_i + \sum_{j \in \text{neighbors}(i)} p_j}{\text{number of neighbors}(i) + 1}. \tag{2}$$

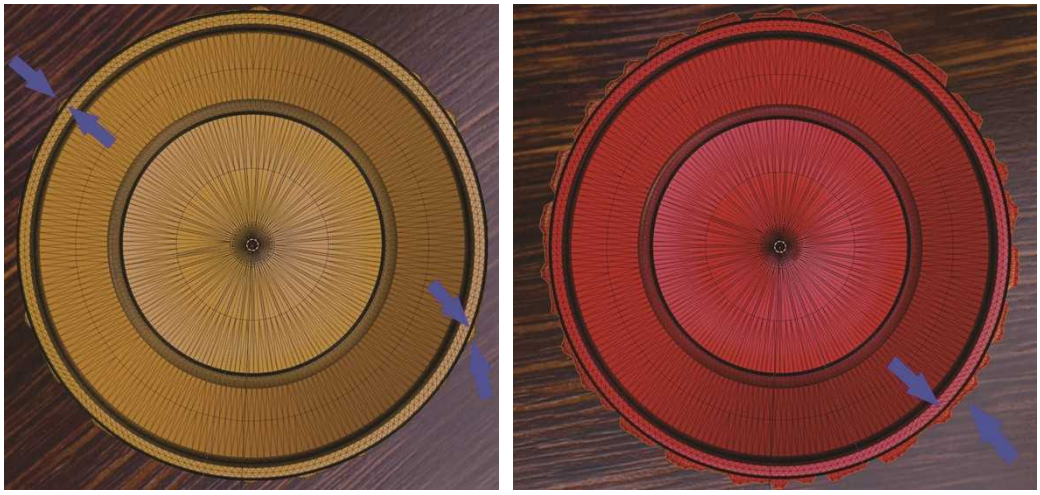
Finally, the vertices are displaced along their normal vectors by the smoothed perturbation amount, updating the position of each vertex v_i to a new position $\mathbf{v}_i'$ by

$$\mathbf{v}_i' = \mathbf{v}_i + p_i^{\text{smooth}} \cdot \mathbf{n}_i. \tag{3}$$

Adjusting the algorithm's parameters allows for the simulation of defects that align with the diverse types and severities encountered in actual injection-molded parts. In our experiments, we determined a smoothness setting of 100 to be universally suitable for the type of defects we aimed to replicate. We varied the strength of the perturbation within the range [2.0, 25.0] and the perturb_fraction within [0.01, 0.1]. Both parameters were drawn from a normal distribution with a mean of 0.055 and a standard deviation of 0.025 for the perturb_fraction, and a mean of 13.5 with a standard deviation of 5.75 for the strength. Values falling outside the defined ranges were clamped to the interval's boundaries.

The strength parameter determines the scale of the defects, affecting their size and the extent of deformation on the mesh. Conversely, the perturb_fraction sets the occurrence rate of the defects, which translates to the frequency and variability of the defects' presentation on the mesh surface. In calibrating these parameters, our objective was to replicate

defects with a high degree of subtlety. We set the median defect size to be relatively small, and the lower threshold of the strength parameter was selected such that defects at a high perturb_fraction are just barely perceptible, and those at the lower end are virtually invisible to the naked eye without zooming in on the image. This nuanced approach aims to enhance the AI's sensitivity to minute irregularities, challenging its detection capabilities. Figure 3 depicts defects of median and maximum possible sizes, characterized by excess material on the outer edge (rendered from a top-down camera perspective). The mesh is displayed to better illustrate the variations in the dimensions of these defects.



(**a**) (**b**)

Figure 3. Examples of different defect sizes. (**a**) Median defect strength and fraction; (**b**) maximum defect strength and fraction. Some of the defects at the brim are marked.

2.3. Render Settings

By randomly selecting one variant of each parameter (as depicted in Table 1), we rendered 10,000 images without defects and 10,000 images with defects, which were generated using the algorithm described in Section 2.2. This resulted in a total dataset of 20,000 images for all variations. Each image was rendered with parameters chosen randomly from those outlined in Table 1, with a 50% probability for each variant. The test dataset, consisting of 4000 images, was rendered separately with the same settings.

Each image was rendered at a resolution of 1920 × 1080 pixels using the Cycles engine (without denoising) using the GPU. In every rendering cycle, the material color was varied randomly among four options: light blue, dark blue, orange, and red. If the part was to include a defect, it was created by manipulating the object's geometry with a random size and shape.

With dependence on the randomly selected variant for each parameter, the following render settings were configured via the Python API:

Noise: Adaptive sampling was enabled for both variants. For lower noise levels, the maximum number of samples was set to 500, and the adaptive threshold was set to 0.01. For higher noise scenarios, the maximum samples were reduced to 50, and the threshold was increased to 0.1.

Lighting: Depending on the random selection, either variable lighting was used by selecting an HDRI from 191 available 2 k textures downloaded from Poly Haven [21] or a disk-shaped area light positioned directly above the object with an energy setting of 5000 was used.

Camera: If the top-view camera was selected, Camera Object 1, located directly above the bowl, was used. Otherwise, Camera Object 2 was chosen with its position randomly determined within a cube above the object and oriented towards it.

2.4. Preliminary Experiments

Choice of classification model: The objective of classification in this research is not to achieve the highest possible accuracy on test or real-world data but rather to assess the comparative impact of different rendering parameters on model performance. The chosen model should not only exhibit good generalization but should also operate efficiently to accommodate the numerous training and inference iterations required by our experiments. We evaluated various models, including different variants of EfficientNet [22] and EfficientNetV2 [23], to determine whether the size of the input images and the complexity of the networks significantly influence performance.

Image normalization: The role of image normalization in the training and testing phases was examined by training the model on a subset of 4000 images randomly selected from the entire dataset of 20,000. Three different normalization schemes were applied:

(a) ImageNet with mean = [0.485, 0.456, 0.406], and std = [0.229, 0.224, 0.225];
(b) Rendered dataset mean = [0.3320, 0.2324, 0.2056], and std = [0.1839, 0.1681, 0.1932];
(c) No normalization

Dataset size: Rendering a substantial volume of images with accurate light simulation is time-intensive; generating 20,000 images required approximately four and a half days using our equipment. Consequently, to manage the rendering duration—given that seven models were to be trained for our studies—without compromising the integrity of the training data, we evaluated the impact of dataset size on model performance. We considered datasets of 6000 and 8000 images, selected randomly from the full dataset to ensure uniform distribution.

2.5. Training EfficientNet

For training the EfficientNetV2-RW-T model, we utilized PyTorch and ensured reproducibility by setting the same seed for all random number generators in torch, numpy, and CUDA. The batch size was determined by the memory capacity of the graphics card, with 24 being the maximum number of samples per batch that could be accommodated. Our dataset was partitioned into an 80/20 split for training and validation. Data loaders were configured with specific transformations for the training and testing datasets (refer to Appendices A and B for details). The training process spanned over 40 epochs, utilizing an initial learning rate of 1×10^{-3}.

2.6. Obtaining Real Test Data

For all subsequent evaluations (Sections 3.2 and 3.3), the models were trained and validated according to the methodology described in Section 2.5. The model with the highest validation accuracy after each epoch was selected for further evaluation. We assessed each model's performance using the synthetic test set and a dataset of real objects photographed under varying environmental conditions. These conditions included placing the object on a wooden workbench and within a specialized inspection device (the inspection device, provided by SQB GmbH, features a back light, top light, and a segmented ring light. It is particularly well-suited for experiments involving variations in lighting and camera parameters, making it an ideal setup for determining optimal image acquisition configurations and for capturing training data) with both light and dark backgrounds. For each setup, we captured images in both well-lit and dimly-lit scenarios, taking two photographs of each object with slight rotations and positional adjustments. Due to the limited availability of real parts with the specified flash defect, only four were available for testing. Conversely, five defect-free parts were used to represent the 'good' class. (Note that three of the defective parts are orange, and one is dark blue. For the IO parts, we had one part of each color available).

3. Results

3.1. Conclusions from Preliminary Experiments

The following results were obtained from preliminary tests to find a valid experimental setup, as described in Section 2.4.

Choice of classification model: Initial tests on a subset of 4000 images from our dataset, encompassing all parameter variations, revealed no significant performance disparities attributable to different input sizes (see Appendix A.1). Consequently, with an emphasis on reducing model size and expediting training, we opted for a pretrained version of efficientnetv2_rw_t, which offers a suitable balance between efficiency and predictive capability.

Image normalization: Confusion matrices, as shown in Table A4, indicate discernible but modest differences in model performance across the various normalization techniques. Upon visual examination, images after normalization, especially those with darker backgrounds, may lose realism and appear overly dark, as can be seen in Figure A1. In light of these observations and the similar performance metrics of the models tested, we opted against using normalization. This decision also streamlines the testing process.

Dataset size: Model performance metrics on the test data are documented in Table A5. The analysis revealed marginal differences among the models, with the model trained on 6000 images (6k) unexpectedly outperforming the one trained on 8000 (8k). Based on these findings, we opted to render a dataset of 6000 images for each of our experiments, with an equal split between the defective and nondefective samples.

3.2. Noise

Tables 2 and 3 present the confusion matrices for the classification results of the models trained on high-noise and low-noise image data, respectively. The left side of each table displays the results on the rendered test dataset, while the right side presents the findings on the real data. Figures A2 and A3 depict the receiver operating characteristic curve (ROC) for the respective noise models when tested against real data. (As a reminder, the area under the receiver operating characteristic curve (AUC-ROC) provides a quantifiable measure of a classifier's accuracy, plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various thresholds. An AUC of 1.0 signifies perfect classification, while an AUC of 0.5 corresponds to the performance of random guessing (diagonal)).

Table 2. Confusion matrix for a high-noise model.

	Synth. Data		**Real Data**	
Predicted:	**G**	**NG**	**G**	**NG**
Actual: G	1721	279	60	0
Actual: NG	182	1818	19	29

Table 3. Confusion matrix for a low-noise model.

	Synth. Data		**Real Data**	
Predicted:	**G**	**NG**	**G**	**NG**
Actual: G	1483	517	53	7
Actual: NG	205	1795	18	30

The overall performance of both models on the test datasets is deemed reasonable. The model trained on low-noise images demonstrates a marked reduction in performance on the synthetic test data, suggesting a potential difficulty in generalizing to the high-noise conditions that are also present in the rendered test set. However, when assessed on real-world images, the low-noise model exhibits better performance but still does not quite reach the effectiveness of the high-noise model. It shows comparable proficiency in recognizing defective objects but tends to misclassify a higher number of nondefective parts.

The ROC values provide further insight into the classification abilities of the models: the high-noise model attains an AUC of 0.9469, while the low-noise model reaches an AUC of 0.8271. Both metrics surpass the baseline of random classification, indicating a significant ability to discriminate between classes. Notably, the high-noise model shows distinctly superior performance in accurately classifying real data compared to the low-noise model, underscoring its robustness in practical settings.

3.3. Camera Position

Tables 4 and 5 show the confusion matrices on the synthetic as well as the real test data. The respective ROC curves are depicted in Figures A4 and A5.

Table 4. Confusion matrix for the top camera model.

	Test Dataset		**Real Dataset**	
Predicted:	**G**	**NG**	**G**	**NG**
Actual: G	961	1039	51	9
Actual: NG	268	1732	18	30

Table 5. Confusion matrix for the variable camera model.

	Test Dataset		**Real Dataset**	
Predicted:	**G**	**NG**	**G**	**NG**
Actual: G	1830	170	60	0
Actual: NG	241	1759	17	31

The comparative analysis of model performance, with respect to camera angles, discloses stark differences. The top camera model erroneously classified over half of the nondefective objects within the synthetic test dataset, indicating a significant challenge in generalizing from the training data to varied camera perspectives. In marked contrast, the variable camera model demonstrated consistent and robust performance across the entirety of the synthetic test data. Remarkably, in real-world evaluations, the variable camera model excelled, correctly classifying all nondefective objects—despite the fact that all real images were captured from a top-down perspective. This performance gap is further highlighted in the receiver operating characteristic analysis, where the variable camera model achieved a robust AUC of 0.9448, which is significantly higher than that of the top camera model, the latter nearing the threshold of random classification efficacy.

3.4. Lighting

Again, Tables 6 and 7 show the confusion matrices on the synthetic as well as the real test data. The respective ROC curves are depicted in Figures A6 and A7.

Table 6. Confusion matrix for the top light model.

	Test Dataset		**Real Dataset**	
Predicted:	**G**	**NG**	**G**	**NG**
Actual: G	1850	150	60	0
Actual: NG	308	1692	31	17

Table 7. Confusion matrix for the HDRI light model.

	Test Dataset		**Real Dataset**	
Predicted:	**G**	**NG**	**G**	**NG**
Actual: G	1834	166	60	0
Actual: NG	237	1763	19	29

The comparative evaluation between models using HDRI versus top light illumination highlights the subtle yet significant performance variances. On the synthetic tests, the HDRI model outperforms marginally, as it compensates for misclassifying 16 nondefective units by correctly identifying an extra 71 defects. For the real images, while both models accurately classify all nondefective parts, the HDRI model proves superior, detecting 29 out of 48 defects, surpassing the top light's 17. The AUC metric underlines these outcomes, with the HDRI model achieving a robust AUC of 0.948, showcasing its strong discriminative power, which is in contrast to the top light model's more modest AUC of 0.7295.

3.5. All Variations

Upon analyzing the images, it was observed that the model trained with varied rendering parameters predominantly misclassified only those with the smallest flash defects and suboptimal lighting conditions (the metrics are given in Table 8 and Figure 4). In setups with a dark background, every image was correctly classified, and with adequate lighting, all images with larger defects were identified accurately regardless of the background.

Table 8. Confusion matrix for all variations.

	Test Dataset		Real Dataset	
Predicted:	**G**	**NG**	**G**	**NG**
Actual: G	1840	160	59	1
Actual: NG	166	1834	14	34

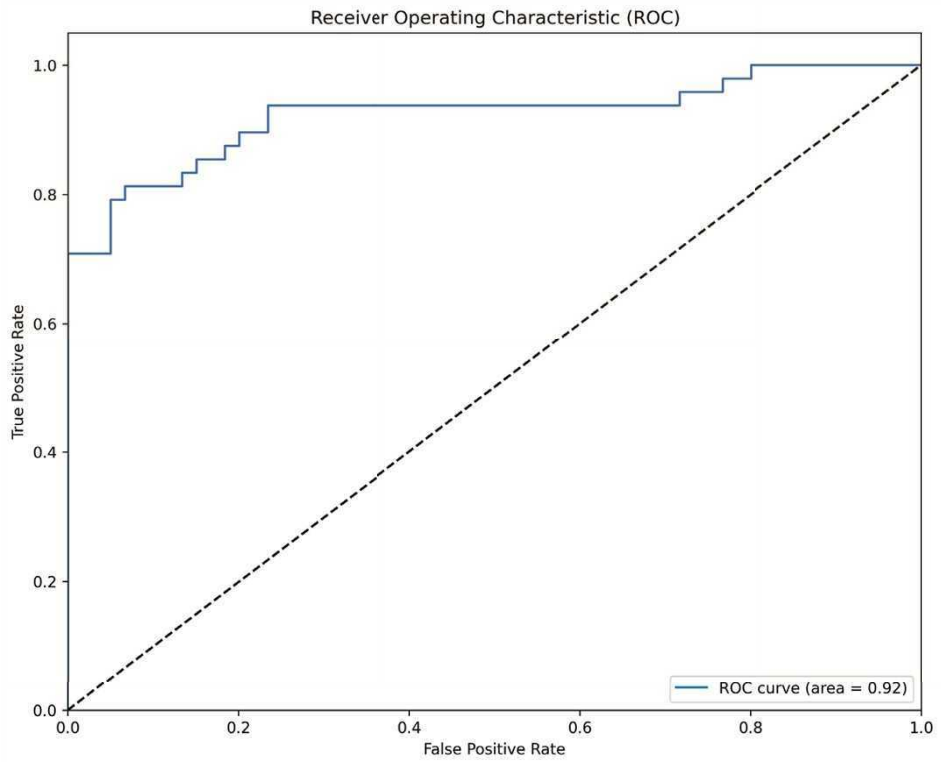


Figure 4. AUC of 0.9233 for the model with all variations using real data.

Illustrations of these observations include a real image with defects incorrectly classified as nondefective, shown in Figure 5, and an image with the smallest detected defect classified correctly, presented in Figure 6.

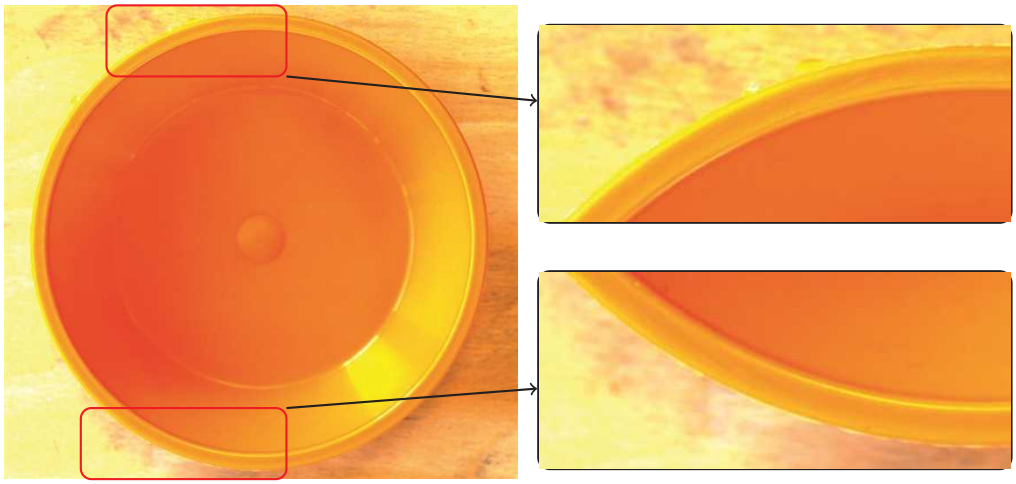


Figure 5. A real part with a small flash that was classified as good.

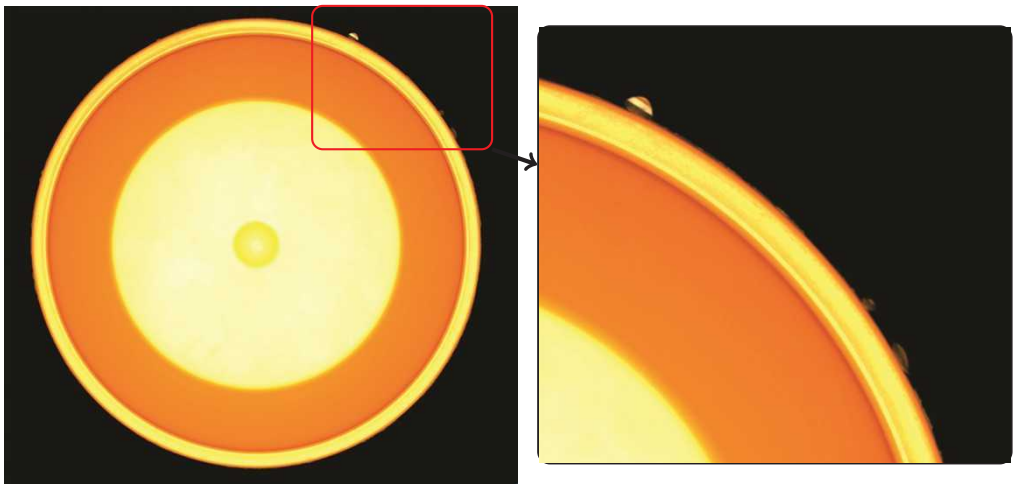


Figure 6. A real part with the smallest flash and darkest background was classified correctly.

Consequently, the experiments culminated, albeit unintentionally, in the creation of an effective classifier, affirming the overall success of the AI training process. This result underscores the model's adeptness at discerning crucial features for precise defect identification.

4. Discussion

In our investigations, we intentionally set the defect parameters to low levels to create subtle defects, resulting in a substantial number of images without recognizable defects. This strategy was not geared towards maximizing classification performance but, rather, was aimed at enabling a meaningful comparative analysis between models trained on varied datasets. During training, we deliberately refrained from using augmentations that could alter image noise or brightness to preserve the purity of the data for assessing the impact of rendering parameters. Universally, the models trained solely on synthetic images demonstrated admirable adaptability when applied to real image data, thus bridging the "reality gap" identified by Tobin et al. [24], and they did so without the need for normalization techniques.

Regarding the **noise** simulation parameter, the high-noise model remained robust when applied to real, lower-noise data (e.g.,Table 2), while the low-noise model struggled to adapt to noisier conditions (e.g., Table 3), suggesting a potential benefit in training with raytracing at lower sampling rates. Similarly, the models with varied **lighting** parameters proved advantageous for classification tasks (e.g., Tables 6 and 7), indicating the effectiveness of incorporating lighting variation into the training process. The adaptability of models

to **camera position** variations also yielded informative outcomes. The models trained with a static top camera perspective performed poorly when faced with deviations in camera angle within the synthetic test set (e.g., Table 4), a trend that was unexpectedly mirrored in real-world scenarios, where the camera positions were consistent, yet the objects were placed only millimeters apart (e.g., Tables 4 and 5). This leads to our recommendation to include at least minimal camera position variation in rendering synthetic training data for inspection tasks to enhance generalization—even if the actual operational setup involves a fixed camera position.

Our findings also imply the feasibility of prioritizing training with noisy images, which are not only quicker to render via raytracing but also seem to enhance model performance. Including a limited selection of low-noise images might suffice to maintain the model's ability to generalize.

In our case, the preliminary tests indicated that models without normalization could match or exceed the performance of normalized counterparts on our test data (e.g., Section 3.1). Nonetheless, we advise caution in extrapolating this result to other contexts. In situations where there is significant brightness variation within a dataset, particularly when real and synthetic data are combined, image normalization could be crucial. This area merits further investigation to determine the most effective image data preprocessing techniques for machine learning applications using synthetic data.

5. Conclusions

In this work, we addressed the previously unexplored importance of simulation parameters in generating synthetic data for enhancing AI performance. Our approach utilized a binary classification task for defect detection using a single defect type and one 3D model. The study's findings underscore the significant impact of varying illumination and camera positions on AI performance. Specifically, changing camera position led to a 9.26% (please note that the % sign here and in the following refers to percentage points) increase in model accuracy and a 21.53% improvement in AUC when tested against real-world data. Similarly, employing HDRI lighting instead of top lighting resulted in an 11.11% boost in classification accuracy and a 16.04% increase in AUC. Counterintuitively, we also discovered that training with higher noise not only aids in model generalization but also speeds up data generation, showing a dual advantage with a 5.56% rise in accuracy and an 11.98% enhancement in AUC compared to models trained with lower noise levels.

5.1. Limitations

Although this study provides valuable insights into the impact of rendering parameters on AI model performance, it has limitations. Firstly, our investigation into the noise parameter was limited to only two variations, which may not fully capture the spectrum of noise levels encountered in real-world scenarios. Secondly, we did not account for variations in texture in the background, a factor that could significantly influence AI classification accuracy. While focusing solely on binary classification and the "flash" defect type might be seen as a limitation, these choices were made to constrain the scope of the study and manage its complexity.

5.2. Practical Implications

Despite these limitations, the practical implications of our research potentially extend far beyond the quality inspection of injection-molded parts. Our findings are pertinent to a broad spectrum of production processes where 3D models are used to algorithmically simulate defects. Tools like Blender or Unity can be leveraged to generate realistic synthetic data, which then serve as training material for AI models tasked with detecting these defects in actual products. Our study demonstrates that manipulating rendering parameters like lighting, camera position, and noise can significantly boost AI model accuracy and generalization. This is vital for those using rendered synthetic data for AI training. Notably, certain parameter variations are more impactful than others, highlighting the importance

of a targeted approach in their selection. Effectively prioritizing these key parameters and varying only those that are beneficial can not only streamline the creation of high-quality synthetic training data but also lead to a dataset that is higher in quality, smaller in size, and generated in less time.

5.3. Future Research

When looking ahead, several promising research directions emerge. Investigating whether variations in the base or background of synthetic images can further improve model accuracy and adaptability is one of those. Another key area is determining the optimal mix of parameter variations in training data, for instance, understanding if a dataset comprising 80% images with lighting variations outperforms one with a lower proportion of such variations. Furthermore, exploring the benefits of a continuous range in variable parameters, like noise levels, instead of binary high/low levels, might offer more nuanced generalization capabilities, particularly in scenarios involving different sensor types or exposure times. These insights pave the way for future research to optimize synthetic data generation for machine learning applications in quality inspection and beyond.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft preparation, funding acquisition, project administration D.S.; writing—review and editing, supervision, G.N.; visualization, D.S. and G.N.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Bundesministerium für Bildung und Forschung (BMBF), grant number 01IS22019 A-E.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We created the synthetic data entirely from the 3D model of the bowl. All rights to this model are owned by Eichsfelder Technik eitech GmbH. https://www.eitech.de/ (accessed on 15 January 2024).

Conflicts of Interest: Author Dominik Schraml was employed by the company Steinbeis Qualitätssicherung und Bildverarbeitung GmbH. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Experiment Details

Appendix A.1. Model

As in defect detection tasks, the "not good" (NG) parts are the class of interest we want to primarily identify; the following naming is used:

True Positive (TP): Defective parts correctly classified as NG.
True Negative (TN): Good parts correctly classified as G.
False Positive (FP): Good parts incorrectly classified as NG.
False Negative (FN): Defective parts incorrectly classified as G.

Appendix A.2. Normalization

The following confusion matrices were the results of training on the 4k images after 40 epochs and inference using the test data.

Table A1. Confusion matrices for models trained with different architectures. From left to right: efficientnet-Bb6-ns; efficientnetv2-rw-t; tf_efficientnetv2_m_in21ft1k.

	B6-ns		**V2-rw-t**		**V2_m_in21ft1k**	
Predicted	**G**	**NG**	**G**	**NG**	**G**	**NG**
Actual: G	1823	177	1974	26	1923	77
Actual: NG	200	1800	297	1703	273	1727

Table A2. Metrics for all models (for NG).

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
efficientnet-Bb6-ns	90.58%	91.05%	**90.00**%	90.52%
efficientnetv2-rw-t	**91.93**%	**98.50**%	85.15%	**91.34**%
tf_efficientnetv2_m_in21ft1k	91.25%	95.73%	86.35%	90.80%

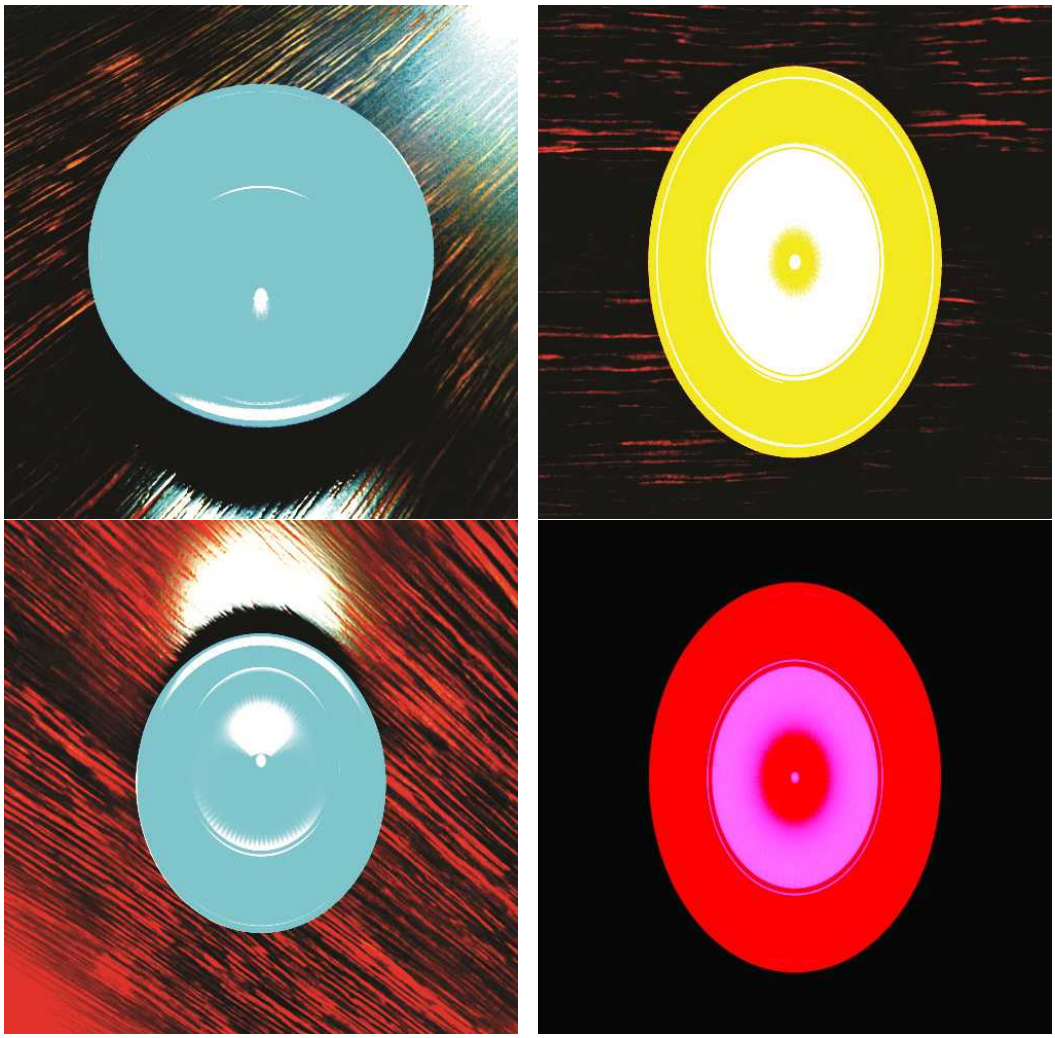


Figure A1. Four examples of rendered images after normalization.

Appendix A.3. Training Dataset Size

We randomly selected 6000 and 8000 images from the rendered complete dataset of 20,000 images and trained three different models (one on each). The resulting confusion matrices are shown in Table A5.

Table A3. Performance metrics for models trained using different normalizations (in %).

Metric	Dataset	ImageNet	None
Precision	92.76%	92.3%	**98.5%**
Accuracy	88.5%	90.7%	**92.0%**
Recall	**91.0**%	89.1%	85.1%
F1-Score	90.6%	90.7%	**91.3%**

Table A4. Confusion matrices for models trained using different normalizations.

	Dataset		ImageNet		None	
Predicted	**G**	**NG**	**G**	**NG**	**G**	**NG**
Actual: G	1862	138	1851	149	1974	26
Actual: NG	230	1770	218	1782	297	1703

Table A5. Confusion matrices for models trained using different numbers of images.

	6000 Images		8000 Images		20,000 Images	
Predicted	**G**	**NG**	**G**	**NG**	**G**	**NG**
Actual: G	1910	90	1770	230	1782	218
Actual: NG	185	1815	164	1836	143	1857

Appendix B. ROC Curves

Appendix B.1. Noise

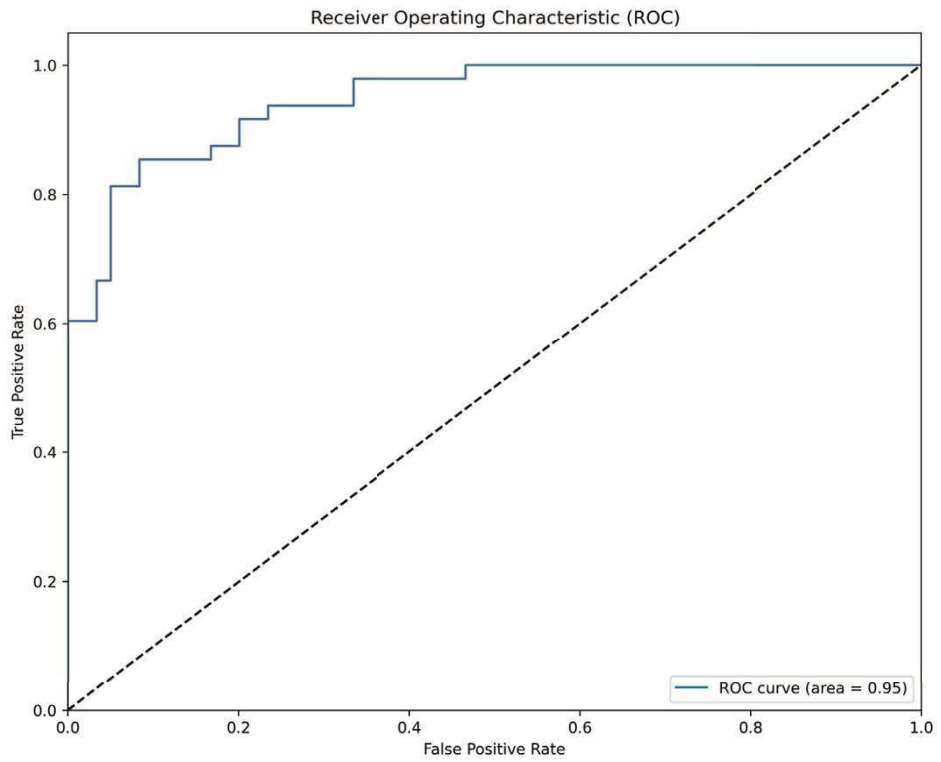


Figure A2. AUC for the high-noise model: 0.9469.

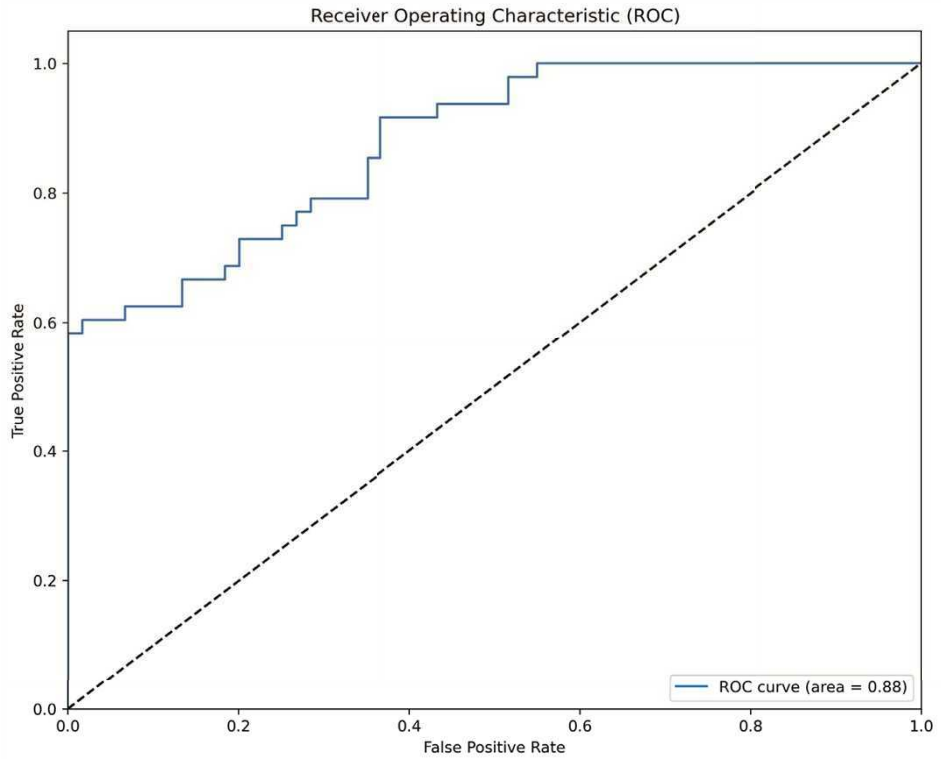


Figure A3. AUC for the low-noise model: 0.8271.

Appendix B.2. Camera

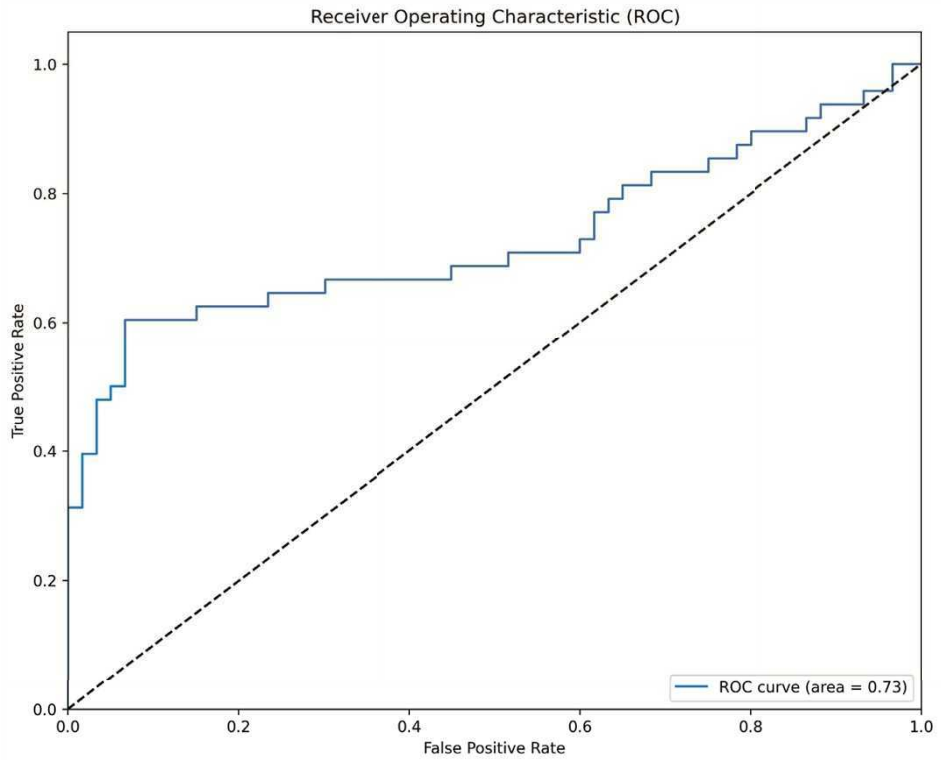


Figure A4. AUC for the top camera model: 0.7295.

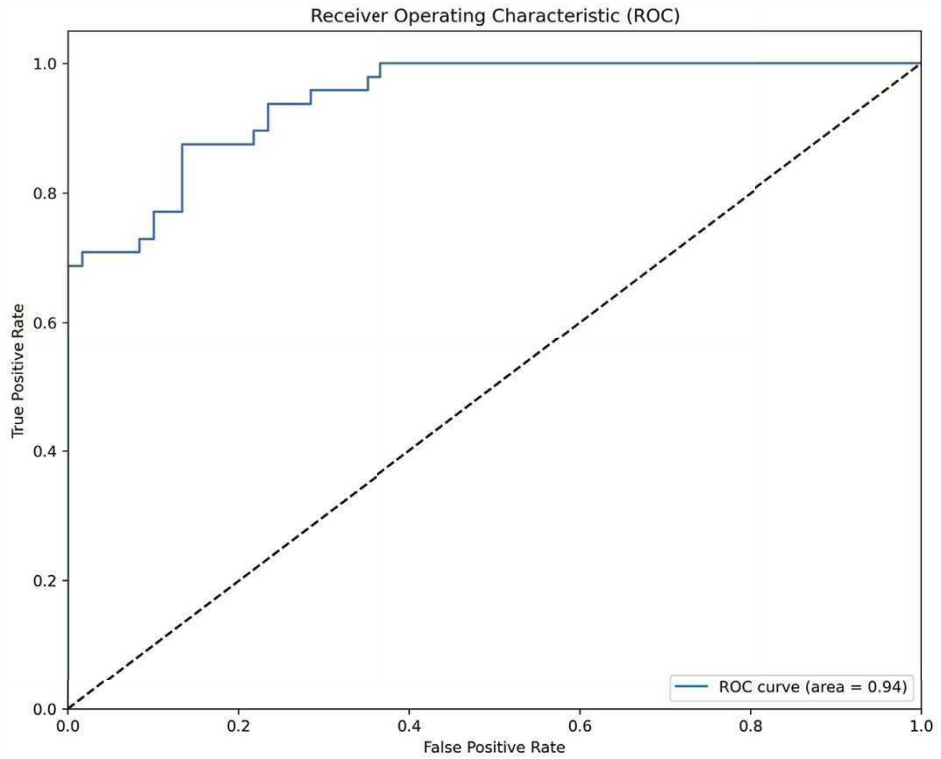


Figure A5. AUC for the variable camera model: 0.9448.

Appendix B.3. Lighting

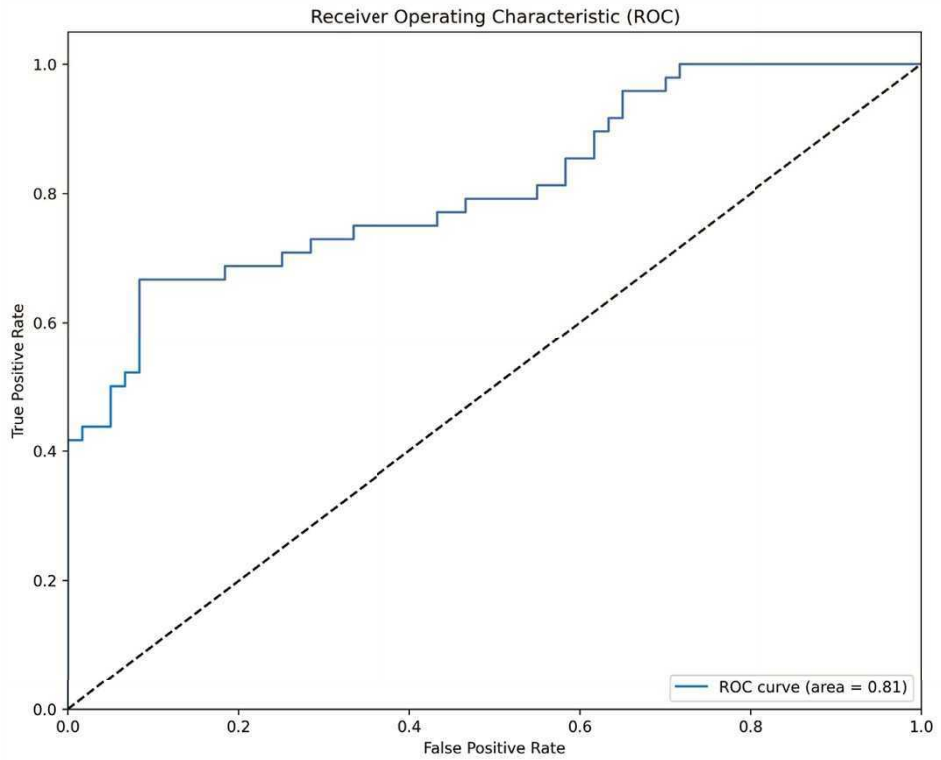


Figure A6. AUC for the top lighting model: 0.8111.

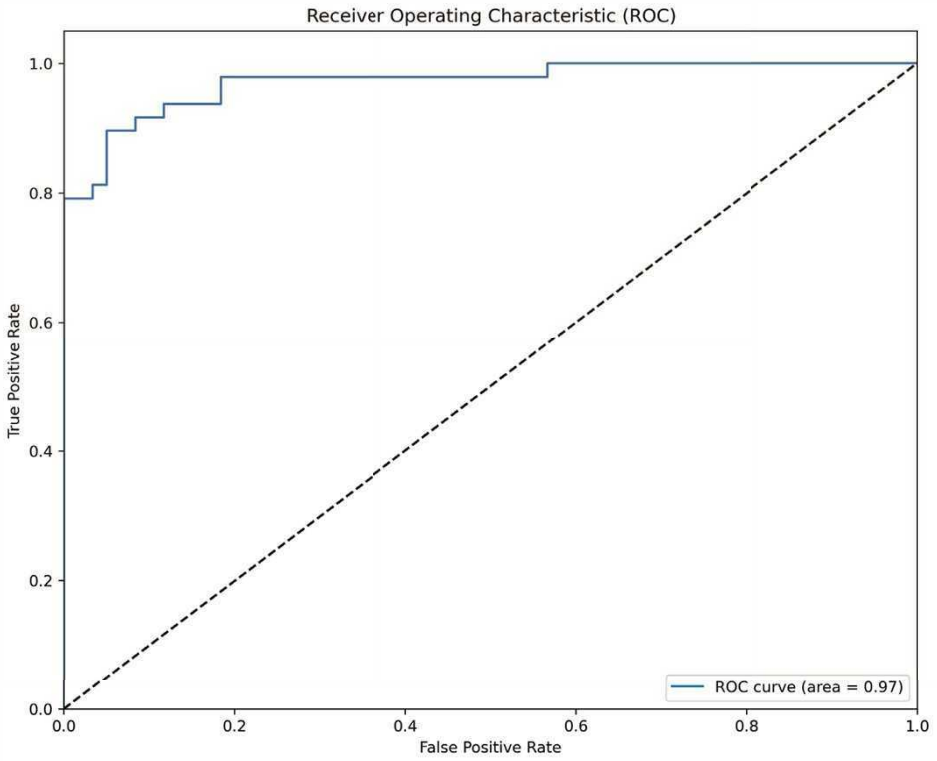


Figure A7. AUC for the variant lighting via HDRI model: 0.9715.

Appendix C. Code

Listing 1. Train transformation pipeline.

```
transforms.Compose([
transforms.Resize(528),
transforms.RandomRotation(degrees=(-30, 30)),
transforms.RandomResizedCrop(size=500, scale=(0.8, 1.0)),
# both equally likely - 0.5 that neither will occur
transforms.RandomHorizontalFlip(0.29289322),
transforms.RandomVerticalFlip(0.29289322),
transforms.ToTensor(),  ])
```

Listing 2. Validation and test transformation pipeline.

```
transforms.Compose([
transforms.Resize(528),
transforms.CenterCrop(size=500),
transforms.ToTensor(),  ])
```

References

1. Schenkenfelder, B.; Kaltenleithner, S.; Sabrowsky-Hirsch, B.; Klug, C.; Lumenta, D.B.; Scharinger, J. Synthesizing Diagnostic Burn Images For Deep Learning Applications. In Proceedings of the 2022 Annual Modeling and Simulation Conference (ANNSIM), San Diego, CA, USA, 18–20 July 2022.
2. Howells, B.; Charles, J.; Cipolla, R. Real-time analogue gauge transcription on mobile phone. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Virtual, 19–25 June 2021.
3. Viggh, H.; Loughran, S.; Rachlin, Y.; Allen, R.; Ruprecht, J. Training Deep Learning Spacecraft Component Detection Algorithms Using Synthetic Image Data. In Proceedings of the 2023 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2023; pp. 1–13.
4. Anderson, J.W.; Ziolkowski, M.; Kennedy, K.; Apon, A.W. Synthetic Image Data for Deep Learning. *arXiv* **2022**, arXiv:2212.06232.

5. Jonker, L.M. Robotic Bin-Picking Pipeline for Chicken Fillets with Deep Learning-Based Instance Segmentation Using Synthetic Data. Master's Thesis, University of Twente, Enschede, The Netherlands, 2023.
6. Bosnar, L.; Hagen, H.; Gospodnetic, P. Procedural Defect Modeling for Virtual Surface Inspection Environments. *IEEE Comput. Graph. Appl.* **2023**, *43*, 13–22. [CrossRef] [PubMed]
7. Gutierrez, P.; Luschkova, M.; Cordier, A.; Shukor, M.; Schappert, M.; Dahmen, T. Synthetic training data generation for deep learning based quality inspection. In *Fifteenth International Conference on Quality Control by Artificial Vision*; Komuro, T., Shimizu, T., Eds.; SPIE: San Diego, CA, USA, 2021; p. 13.
8. Wunsch, L.; Anding, K.; Polte, G.; Liu, K.; Notni, G. Data augmentation for solving industrial recognition tasks with underrepresented defect classes. *Acta Imeko J.* **2023**, *12*, 1–5. [CrossRef]
9. Fuchs, P.; Kröger, T.; Garbe, C.S. Defect detection in CT scans of cast aluminum parts: A machine vision perspective. *Neurocomputing* **2021**, *453*, 85–96. [CrossRef]
10. Boikov, A.; Payor, V.; Savelev, R.; Kolesnikov, A. Synthetic Data Generation for Steel Defect Detection and Classification Using Deep Learning. *Symmetry* **2021**, *13*, 1176. [CrossRef]
11. Bonfiglioli, L.; Toschi, M.; Silvestri, D.; Fioraio, N.; de Gregorio, D. The Eyecandies Dataset for Unsupervised Multimodal Anomaly Detection and Localization. In *Computer Vision—ACCV 2022*; Wang, L., Gall, J., Chin, T.J., Sato, I., Chellappa, R., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2023; pp. 459–475.
12. Cordier, A.; Gutierrez, P.; Plessis, V. Improving generalization with synthetic training data for deep learning based quality inspection. *arXiv* **2022**, arXiv:2202.12818.
13. Delgado, G.; Cortés, A.; García, S.; Loyo, E.; Berasategi, M.; Aranjuelo, N. Methodology for generating synthetic labeled datasets for visual container inspection. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *175*, 103174. [CrossRef]
14. Hoskere, V.; Narazaki, Y.; Spencer, B.F. Physics-Based Graphics Models in 3D Synthetic Environments as Autonomous Vision-Based Inspection Testbeds. *Sensors* **2022**, *22*, 532. [CrossRef] [PubMed]
15. Leon-Alcazar, J.; Alnumay, Y.; Zheng, C.; Trigui, H.; Patel, S.; Ghanem, B. Learning to Read Analog Gauges from Synthetic Data. *arXiv* **2023**, arXiv:2308.14583.
16. Ruediger-Flore, P.; Glatt, M.; Hussong, M.; Aurich, J.C. CAD-based data augmentation and transfer learning empowers part classification in manufacturing. *Int. J. Adv. Manuf. Technol.* **2023**, *125*, 5605–5618. [CrossRef]
17. Schmedemann, O.; Baaß, M.; Schoepflin, D.; Schüppstuhl, T. Procedural synthetic training data generation for AI-based defect detection in industrial surface inspection. *Procedia CIRP* **2022**, *107*, 1101–1106. [CrossRef]
18. Schoepflin, D.; Holst, D.; Gomse, M.; Schüppstuhl, T. Synthetic Training Data Generation for Visual Object Identification on Load Carriers. *Procedia CIRP* **2021**, *104*, 1257–1262. [CrossRef]
19. Wong, M.Z.; Kunii, K.; Baylis, M.; Ong, W.H.; Kroupa, P.; Koller, S. Synthetic dataset generation for object-to-model deep learning in industrial applications. *PeerJ Comput. Sci.* **2019**, *5*, e222. [CrossRef] [PubMed]
20. Zhang, X.; Jia, N.; Ivrissimtzis, I. A study of the effect of the illumination model on the generation of synthetic training datasets. *arXiv* **2020**, arXiv:2006.08819.
21. Poly Haven 2023. Available online: https://polyhaven.com/ (accessed on 22 August 2023).
22. Tan, M.; Le, V.Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *International Conference on Machine Learning*; PMLR: Westminster, UK, 2019.
23. Tan, M.; Le, V.Q. EfficientNetV2: Smaller Models and Faster Training. In *International Conference on Machine Learning*; PMLR: Westminster, UK, 2021.
24. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the IROS Vancouver 2017, Piscataway, NJ, USA, 24–28 September 2017; pp. 23–30.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Convolutional Neural Networks for Raw Signal Classification in CNC Turning Process Monitoring

Emmanuel Stathatos, Evangelos Tzimas, Panorios Benardos * and George-Christopher Vosniakos

Manufacturing Technology Laboratory, School of Mechanical Engineering, National Technical University of Athens, Heroon Polytechniou 9, GR15772 Athens, Greece; mstatha@mail.ntua.gr (E.S.); v_tzimas@mail.ntua.gr (E.T.); vosniak@central.ntua.gr (G.-C.V.)

* Correspondence: pbenard@mail.ntua.gr; Tel.: +30-210-772-1799

Abstract: This study addresses the need for advanced machine learning-based process monitoring in smart manufacturing. A methodology is developed for near-real-time part quality prediction based on process-related data obtained from a CNC turning center. Instead of the manual feature extraction methods typically employed in signal processing, a novel one-dimensional convolutional architecture allows the trained model to autonomously extract pertinent features directly from the raw signals. Several signal channels are utilized, including vibrations, motor speeds, and motor torques. Three quality indicators—average roughness, peak-to-valley roughness, and diameter deviation—are monitored using a single model, resulting in a compact and efficient classifier. Training data are obtained via a small number of experiments designed to induce variability in the quality metrics by varying feed, cutting speed, and depth of cut. A sliding window technique augments the dataset and allows the model to seamlessly operate over the entire process. This is further facilitated by the model's ability to distinguish between cutting and non-cutting phases. The base model is evaluated via k-fold cross validation and achieves average F1 scores above 0.97 for all outputs. Consistent performance is exhibited by additional instances trained under various combinations of design parameters, validating the robustness of the proposed methodology.

Keywords: industry 4.0; smart manufacturing; process monitoring; signal processing; deep learning; part quality prediction; CNC machining

1. Introduction

Industry 4.0 is characterized by the integration of digital technologies, including Big Data, the Internet of Things, Machine Learning (ML), and cyber-physical systems into manufacturing [1,2]. Cyber-physical systems often appear in the form of Digital Twins, which typically incorporate advanced ML models for monitoring individual machines [3,4] or broader systems [5,6]. These models carry out real-time inference based on data collected from the physical assets, in order to evaluate the monitored system's status and even propose corrective actions when needed [7]. CNC machine tools are prime candidates for integration with such process monitoring models. The machine tool provides useful process-related data in the form of signals, and the model uses these signals to predict outcomes, such as quality metrics of the produced parts [8] and equipment health [9].

Signals have been widely used as model inputs in the domains of machine health monitoring and predictive maintenance. The typical procedure involves hand-crafted feature design in the time, frequency, or time–frequency domain, followed by feature extraction from the signals and model training [10]. This methodology has been employed in manufacturing for part quality characterization [11,12], chatter identification [13,14], tool wear prediction [15,16], and tool breakage detection [17]. However, manual feature design requires deep expert knowledge, is strictly application specific, and can lead to poor model

performance due to inadequate information representation [18], especially for complex domains and multi-sensor setups.

Deep Learning (DL) offers powerful modeling technologies to alleviate these issues, either in combination with traditional techniques or as standalone alternatives [10]. The most prominent types of DL models are Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). RNNs had been the standard for processing sequential data before Transformers emerged in 2017 [19]. These networks capture the temporal relationships of the elements in their input and have been successfully applied in tasks such as time-series classification, prediction, and language processing. However, they face difficulties with long sequences due to memory limitations and the vanishing or exploding gradients problem [20]. Long Short-Term Memory networks (LSTMs), a more advanced form of RNNs, offer improvements but they still struggle with processing high-rate signals, such as the ones common in industrial applications. For this reason, they must typically be combined with a feature extraction step [21,22], focusing on learning temporal dependencies between features instead of the elements of the raw input.

CNNs operate by extracting local information from the input arrays in the form of feature maps [20,23]. With each successive convolutional layer, these feature maps detect increasingly complex patterns that underlie the specific characteristics of the input data. Due to this capability, CNNs have contributed significantly to the Artificial Intelligence boom of the 2010s. They find diverse applications, from image recognition [24–26] to state representation in sophisticated reinforcement learning frameworks [27,28].

In process monitoring, CNNs offer the potential to avoid the manual feature design and extraction steps in favor of adaptive and arbitrarily rich feature representations, which are integrated in the model itself. Since most CNN architectures were originally designed for two-dimensional (2D) images, transforming signal data into 2D form was a logical approach. In CNC turning, Ibarra-Zarate et al. [29] used 2D CNNs on images of the Mel-Frequency Cepstral Coefficients obtained from acoustic emissions to predict surface roughness. Kuo et al. [30] applied 2D CNNs on fractional order chaos maps of vibration data for chatter detection. Hanchate et al. [31] proposed a framework for predicting average surface roughness in CNC grinding, employing 2D CNNs on time–frequency spectrogram frames of vibration signals. Furthermore, they implemented an explainable AI methodology to identify the most pertinent time–frequency bands influencing the predictions. A similar technique was used by Tran, Liu, and Tran [32] to detect chatter in CNC milling. They applied 2D CNNs on scalograms of the continuous wavelet transform of cutting force signals to classify cutting regions as stable, transitive, or unstable. Kounta et al. [33] focused on chatter prediction via transfer learning. They developed a classifier by fine-tuning pretrained deep 2D CNNs on the normalized Fast Fourier Transform (FFT) images of vibration data. Transfer learning was also employed by Unver and Sener [34] on intrinsic mode functions of cutting force signals. They proposed a combination of 2D CNNs and analytical solutions of the stability boundary for chatter detection in shoulder milling.

CNNs, however, are not restricted to 2D inputs. Signals are one-dimensional (1D) arrays and using 1D convolutions on them removes a preprocessing step while retaining all feature extraction abilities. Additionally, working with 1D data can result in more lightweight models that run faster and require less memory. In the industrial domain, 1D CNNs have been successfully applied to fault detection in rotating machinery, either directly on the raw time-series data [35–38] or on frequency-domain data, obtained after applying the FFT [39,40].

This approach has recently started gaining traction in manufacturing, predominantly in CNC milling. Zhang et al. [41] employed CNNs with 1D-adapted inception modules and residual blocks for chatter identification based on raw cutting force signals. Lu et al. [42] developed vibration-based 1D CNN models to predict chatter during milling of thin-walled parts. The CNNs are assisted by an attention mechanism that adaptively identifies information-rich frequency bands, while reducing noise interference. Huang and Lee [43]

applied 1D CNNs to raw vibration, acoustic emission, and spindle current signals to estimate tool wear and surface roughness development. They trained separate models for each metric and performed an influential sensor selection analysis, which preserves estimation accuracy. Lin et al. [44] also developed 1D CNN regression models for surface roughness and reported the superior performance of CNNs when directly compared with FFT feature extractors.

From the above analysis, it is evident that in the realm of process monitoring there is a recent trend to move away from manual feature extraction methods, with deep learning approaches that employ CNNs showing promising results. Much of the research has focused on chatter detection and tool wear prediction. In terms of part quality, surface roughness has garnered considerable interest while there is a lack of attention to other important quality indicators, such as dimensional accuracy. Regarding implementation, the majority of classification models are binary, for example, detecting stable or unstable cut. Regression models can effectively work only when the machine is actively cutting. To the authors' best knowledge, all individual models monitor a single metric.

This study addresses the need for part quality prediction in the context of a process monitoring framework in manufacturing. Recognizing the limitations of traditional feature extraction methods, the proposed methodology employs 1D CNNs on raw time series data from a CNC turning center, including vibrations, motor speeds, and motor torques, see Figure 1. Training data are acquired via a small number of experiments with varying process conditions to produce variability in the monitored metrics. A sliding window technique augments the dataset and allows for continuous monitoring of the entire process. Three quality metrics are selected: arithmetic average roughness (Ra), average peak-to-valley roughness (Rz) [45], and diameter deviation from the nominal value. The classification approach allows for the incorporation of labels not necessarily related to quality assessment. In this case, the model can detect when the machine is not cutting, eliminating the need for external triggers to initiate the inference process. All three metrics, including the non-cutting condition, are monitored by a single model, resulting in an efficient and easily deployable process monitoring framework. Furthermore, several instances of the model are trained for various combinations of design parameters, validating the robustness of the methodology and network architecture under various conditions.

The rest of the paper is organized as follows: Section 2 details the equipment and experimental procedure and presents the experimental data. It also delves into the mapping between signals and quality measurements. Section 3 analyzes the design of the classes, the preparation of data for neural network training, and outlines the network architecture. Section 4 presents and discusses the model's performance under various conditions, including multiple combinations of design parameters, reduced input from a subset of available signal channels, and alternative definitions of classes. A datastream simulation demonstrates the model's capability to closely follow a machining process in near-real-time. Section 5 highlights the main conclusions of the study and ends with a discussion on potential future work.

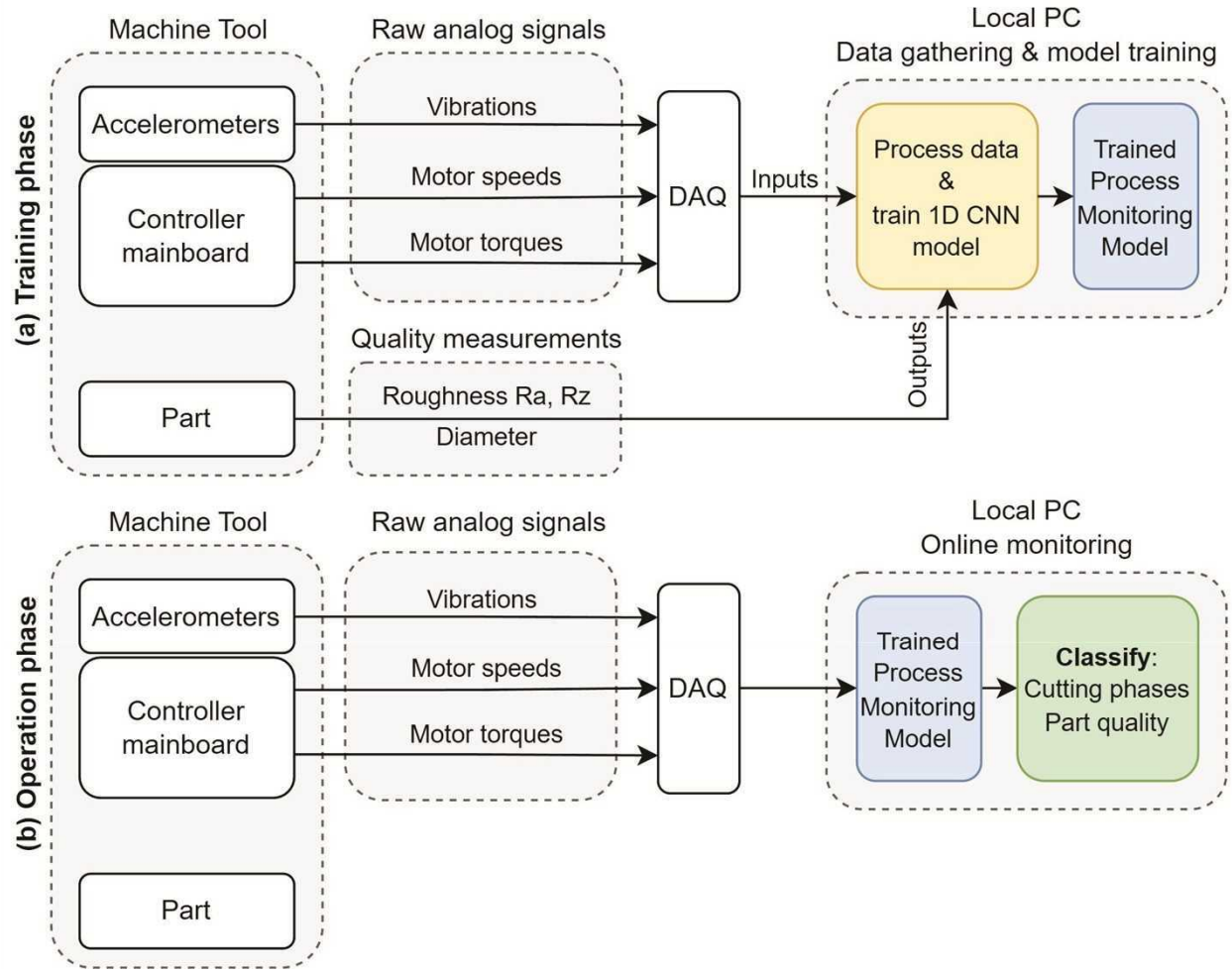


Figure 1. Overview of the proposed methodology: (**a**) Training phase; (**b**) Operation phase.

2. Experimental Setup

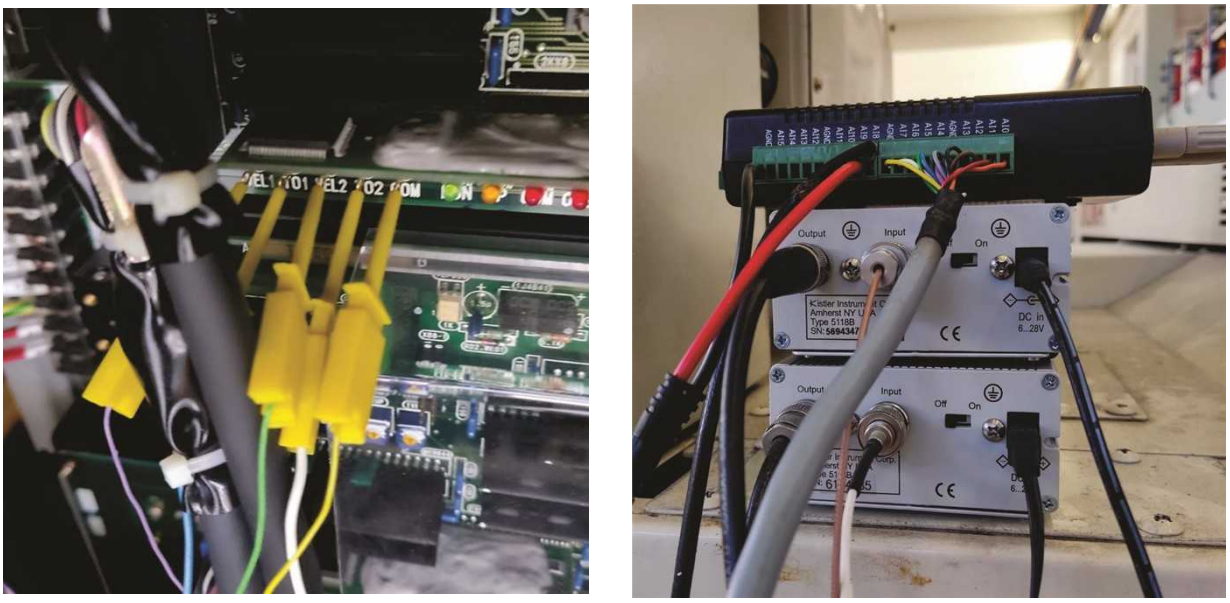
2.1. Machine Tool and Instrumentation

Machining experiments were conducted on an Okuma™ LB10II turning center, equipped with an OSP700L controller, as shown in Figure 2. The machine tool has a maximum turning diameter of 170 mm, a maximum part length of 300 mm, a maximum spindle speed of 10,000 RPM, and 12 turret positions. It has a serial port with DNC capability, which is used to load the G-code. The cutting inserts used are Sandvik Coromant™ CNMG 12 04 08-MR 4305, mounted on a Sandvik Coromant™ PCLNR 2020K 12 tool holder.

The machine tool lacks connectivity for providing real-time process data such as spindle rotational speed (RPM), axes linear velocities, and cutting forces. To address this, a direct connection was made to the machine controller mainboard, which provides raw analog signals for motor speeds and torques. The signals for motor speeds correspond to the spindle's rotation speed and the linear velocities of the X (radial) and Z (longitudinal) axes. Motor torques correlate with cutting force components. Connection to the controller pins was established using hook-type connectors, see Figure 3a. The other ends of the connectors were soldered to a 10-pin port bolted in a slot that was cut in the machine cover. In essence, the machine tool was fitted with a custom-created port, facilitating straightforward physical access to the controller's raw analog signals for the data acquisition system.



Figure 2. OKUMA LB10ii turning center.



(**a**) (**b**)

Figure 3. (**a**) Connections to machine controller; (**b**) DAQ and amplifiers.

For the acquisition of vibration signals, two single-axis accelerometers were used: a KISTLER Type 8640A50 with a ±50 g range and a KISTLER Type 8704B500 with a ±500 g range. Ideally, accelerometers would be mounted directly on the tool holder. However, the presence of a turret, which rotates to change tools, renders this approach unsafe. Additionally, equipping each potential tool required for a given process scenario with dedicated accelerometers is prohibitively costly for a practical application. The proposed solution involves mounting the accelerometers behind the turret using a common magnetic base, see Figure 4a, aligning their axes of measurement with the axes of the turning center (X and Z). The exact placement of the accelerometers is not critical. However, it is crucial that they remain stationary once installed, as moving them would alter the signal characteristics between runs, impacting both experimental results and operational monitoring. This placement results in vibrations traveling through a kinematic chain containing multiple

structural elements of the machine before reaching the sensors. Consequently, the signals are weakened, and the readings obtained do not correspond exclusively to vibrations on the expected axis. Instead, the vibrations detected are vectors with components from all axes, indicating a more complex signal composition. The hypothesis behind this sensor placement was that crucial process characteristics are retained in the signal, and neural networks would be capable of discerning them. Successfully training these models should demonstrate their robustness, even when relying on far from ideal input data.



(**a**) (**b**)

Figure 4. (**a**) Placement of accelerometers behind the machine tool turret; (**b**) Platform for measurement of surface roughness.

The accelerometers were connected to KISTLER amplifiers (Type 5118B2) using low-noise coaxial cables to ensure high-quality signal transmission. A 10× gain was selected for the second, lower sensitivity accelerometer to bring both sets of readings to the same scale. Accelerometer and motor signals were passed through an ADVANTECH USB DAQ (Type USB-4711A) as shown in Figure 3b to ensure signal synchronization and to facilitate connection to a PC gateway via USB port. The DAQNavi software (version 4.0.9.0) was employed for signal acquisition.

Signals were recorded with a sampling rate of 1 kHz. All types of acquired signals along with the respective DAQ channels are presented in Table 1.

Table 1. DAQ setup with acquired signals and respective axes.

DAQ channel	2	3	4	5	6	7	8	9
Axis	Spindle	Spindle	Z	Z	X	X	X	Z
Signal (V)	Torque	Speed	Velocity	Torque	Velocity	Torque	Vibration	Vibration
Source	Machine controller						Accelerometers	

Part quality measurements were carried out manually using relevant instruments after each cut and without removing the part from the spindle. Part diameter was measured with a Tesa^{TM} Micromaster digital CAPA μ micrometer. For surface roughness, a Taylor Hobson^{TM} Surtronic 3+ was employed, mounted on a custom jig with a magnetic base, see Figure 4b. This setup enabled quick and stable mounting and easy dismounting after each set of measurements.

2.2. Experimental Procedure

The experimental strategy involves taking longitudinal passes of a specific length on cylindrical workpieces, using various combinations of process parameters, namely depth of

cut, feed, and cutting speed, and recording the signals described in Section 2.1. After each pass, the process is halted, and manual quality measurements are performed at specific positions along the workpiece. For surface roughness metrics, Ra and Rz are selected. For dimensional accuracy, the part diameter is measured, which allows for the calculation of diameter deviation (Ddev) from the nominal value, as specified by the G-code.

The bar stock material is CK45 steel with an initial diameter of 32 mm and a total length of 75 mm. The machined length is set to 45 mm, allowing for the clamping of the workpiece to the chuck and providing some leeway for safety. Quality measurements are taken from three regions, each 15 mm long, denoted as L1, L2, and L3, as shown in Figure 5. These regions accommodate the stroke length of the Taylor Hobson instrument. The part diameter is measured at the midpoint of each region. Each individual measurement is repeated three times at 120-degree angular intervals, and the average of these measurements is taken as the final value. A python script automatically extracts G-code files for each set of process parameters. Constant linear cutting speed is employed (G96 command) and pauses for manual measurements after each pass are specified using the M1 command (optional stop). The cutting fluid utilized is Premiercut GP Semi-Synthetic Cutting Fluid at a 5% dilution, in accordance with the manufacturer's specifications.



Figure 5. Machined specimens. L1, L2, and L3 denote measurement regions.

To produce variability in the quality metrics with a limited number of experiments, an orthogonal array is employed with four levels for each of the three factors, as presented in Table 2, resulting in a total of 16 experiments (i.e., passes). The spindle speed is also calculated, due to the G96 command, to ensure that the max rotational spindle speed of the machine tool is not exceeded. Each set of eight consecutive passes, with the specific sequence of depths of cut, reduces the initial diameter of the workpiece by 20 mm. Therefore, starting from a diameter of 32 mm, the part is machined down to a final diameter of 12 mm, at which point a new workpiece is used. For the specified design, only two parts are required. Taking consecutive passes on the same part ensures that different diameters are exposed to a variety of process conditions. This effectively introduces a fourth factor in the experimental design, the slenderness of the part, as expressed by the Length-to-Diameter ratio (L/D). This variation in stiffness broadens the coverage of the process input space, potentially leading to greater variation in the measured quality metrics. Given that the length is constant in this case, the diameter itself is a representation of this slenderness.

Table 2. Experimental design.

Experiment No	Workpiece Id	Depth of Cut (mm)	Feed (mm/rev)	Cutting Speed (m/min)	Final Diameter (mm)	Spindle Speed (RPM)
1	Part 1	0.5	0.2	160	31	1642.89
2		1	0.3	190	29	2085.48
3		1.5	0.4	220	26	2693.39
4		2	0.5	250	22	3617.16
5		0.5	0.3	220	21	3334.67
6		1	0.2	250	19	4188.29
7		1.5	0.5	160	16	3183.1
8		2	0.4	190	12	5039.91
9	Part 2	0.5	0.4	250	31	2567.02
10		1	0.5	220	29	2414.76
11		1.5	0.2	190	26	2326.11
12		2	0.3	160	22	2314.98
13		0.5	0.5	190	21	2879.95
14		1	0.4	160	19	2680.5
15		1.5	0.3	250	16	4973.59
16		2	0.2	220	12	5835.68

2.3. Mapping Signals to Quality Measurements

In the proposed model, signals are inputs and quality metrics, which are expressed as labels corresponding to specific ranges of these metrics, are outputs. These data need preprocessing in order to map quality measurements to the appropriate signal segments so that the input–output pairs can ultimately be constructed. To ensure a continuous signal history for each of the two specimens, the first step involves merging the signals from individual passes into a single, continuous signal for each part. This step is required because the process and data acquisition are paused after each pass to allow for manual measurements. The continuous signals contain both cutting and non-cutting regions. The cutting regions, where the tool is engaged with the workpiece, have a direct correspondence to the measured metrics. Non-cutting regions are a result of the free movement of the axes as the machine repositions the tool to follow the process plan indicated by the G-code.

The next step is to identify these cutting regions based on information included in the signals. The most appropriate signal on which to base this segmentation is spindle speed. This is expected to remain constant at various levels for the duration of a pass. Due to the usage of G96 (constant linear cutting speed), the smaller the machined diameter, the higher the spindle speed plateaus, for a given cutting speed. The raw, noisy spindle speed signal is cleaned by applying a moving average with a window of 100 ms. The cleaned signal is then numerically differentiated, and the flat regions, where the gradient is below a threshold of 0.03, are identified. Flat regions shorter than 500 ms are disregarded to exclude irrelevant periods, such as when the spindle is re-engaged after each pause but operates at constant RPM for a brief duration before actual cutting starts. Additionally, any flat regions below a 0.5 V cutoff are also ignored to prevent misidentifying non-cutting regions where the spindle is stopped, and the spindle speed plateaus near zero. The beginning and end of the remaining flat regions denote the timestamps for the pure cutting phases. A custom python script automates the described procedure. The values for the flatness threshold, the minimum segment length, and the amplitude cutoff are given in Table 3. These were determined through experimentation involving trial-end-error and are effective across a wide range of process conditions. From this point forward, the term ‘pass’ will be used to denote each pure cutting region.

Table 3. Parameters for identifying cutting regions.

Flatness Threshold	Minimum Segment Length	Amplitude Cutoff
0.03	500 ms	0.5 V

The resulting segmentation into cutting and non-cutting regions is illustrated in Figure 6. This figure displays the spindle speed signals, both raw and cleaned, for the entire history of the first specimen (Part 1). Green dashed lines mark the beginning of each pass, while red dashed lines indicate the end. Outside of these regions, the machine tool is not cutting, i.e., the tool is not engaged with the workpiece. The eight passes corresponding to the first half of the experimental design, which are associated with Part 1, are clearly identifiable. An additional step is then necessary to further segment each pass into the three sub-regions (L1, L2, and L3) at which measurements were taken. This requires knowledge of the measurement strategy, i.e., in our case three equidistant measurements on each pass. This segmentation is indicated by yellow lines in Figure 6, only for the first pass to avoid overcrowding the illustration.

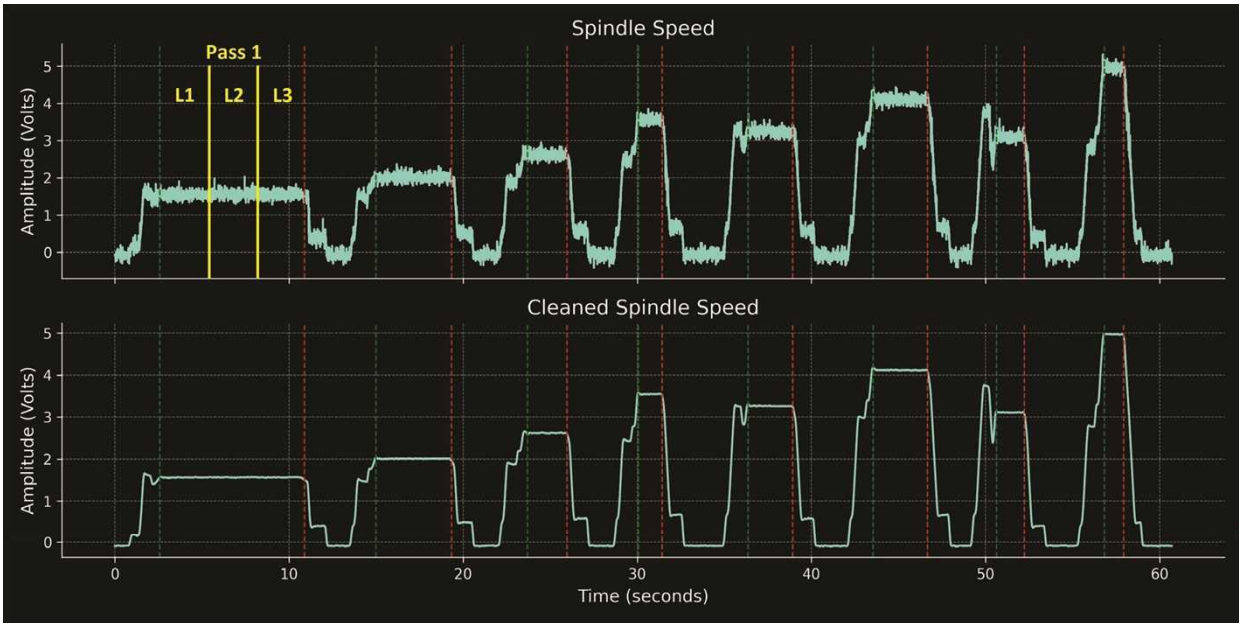


Figure 6. Segmentation of signals based on flat regions of spindle speed for Part 1.

2.4. Process Signals

Figure 7 illustrates the continuous history of Part 1 for the channels corresponding to spindle torque and speed, as well as X and Z axis vibrations. The start and end of individual passes are marked by vertical dashed lines (green for start, red for end). The pure cutting phases are clearly distinguishable in the vibration signals. Their amplitude increases significantly during these periods, from near-zero values otherwise, in alignment with the spindle speed plateaus. This alignment further validates the method of locating cutting phases based on spindle speed. The spindle torque tends to overshoot, either positively or negatively, during spindle start-up, shutdown, or changes in RPM. Notably, within the cutting phases, it stabilizes at a non-zero value, correlating with the tangential cutting force component.

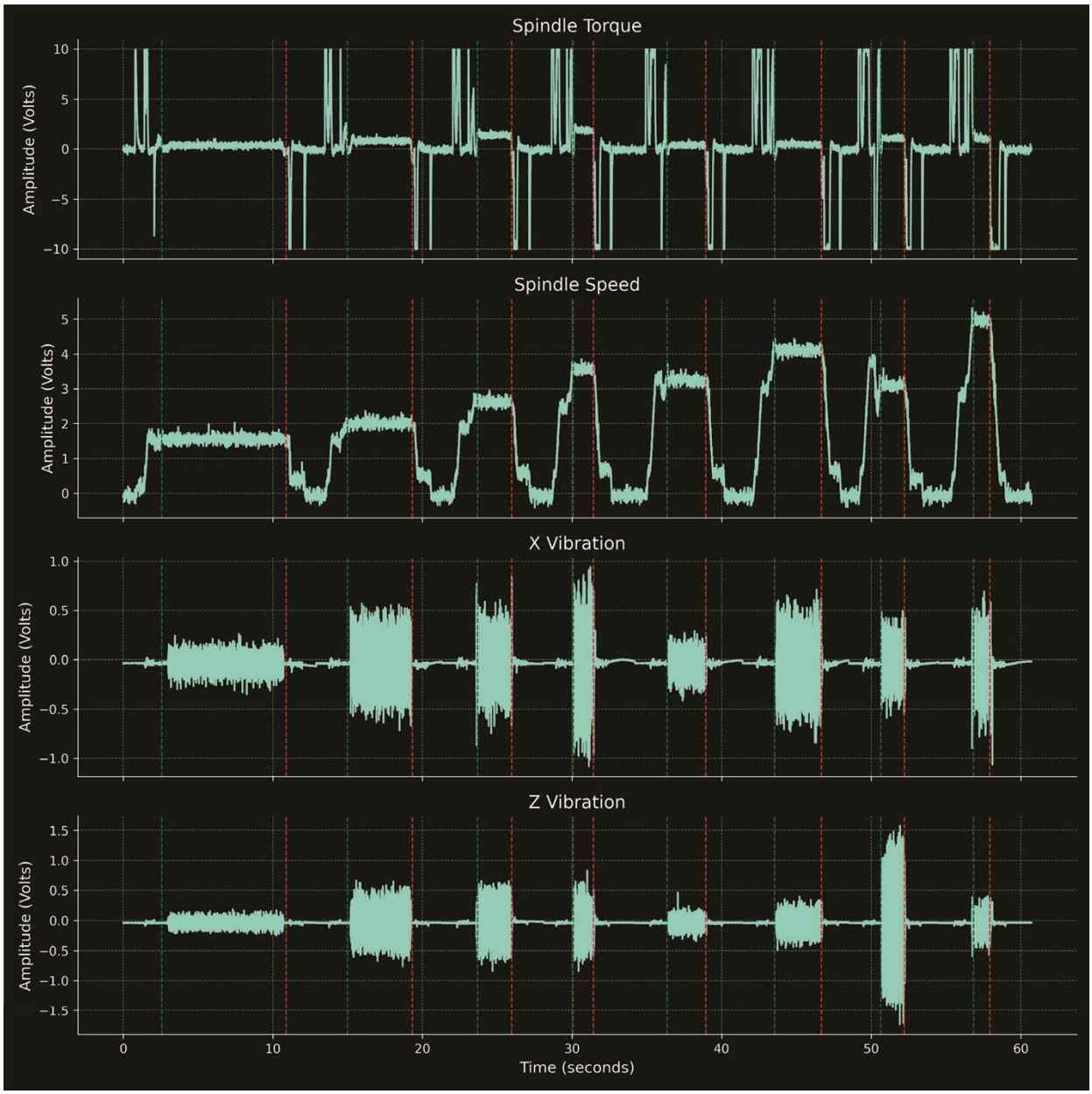


Figure 7. Spindle torque and speed, X and Z vibrations for Part 1.

The remaining four signal channels are shown in Figure 8. Rapid G00 movements outside of cutting phases result in overshooting that is clearly identified in the X and Z velocity signals. During cutting, the Z velocity takes small negative values corresponding to the feed parameter. This effect is more pronounced during shorter passes, which corresponds to higher feed rates. As expected in purely longitudinal turning, the X velocity remains near zero. Both X and Z torques display inertial readings during axes start and stop phases. The Z torque exhibits correlation with axial force, taking constant negative values during cutting phases. The X torque remains near zero, reflecting the minimal radial force component during cutting.

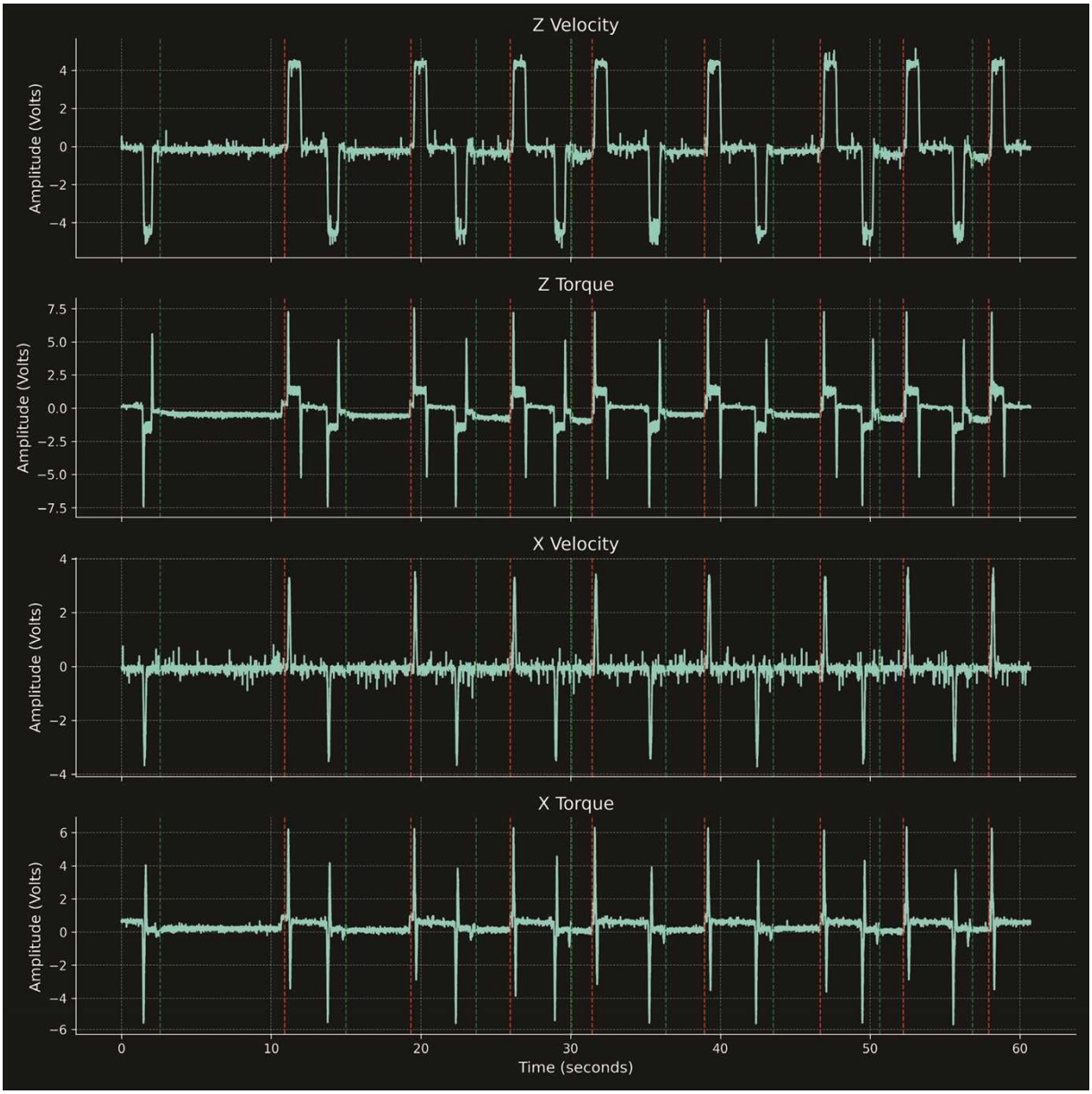


Figure 8. X and Z velocities and torques for Part 1.

2.5. Quality Measurements

The manual measurements for all quality metrics of the two specimens are compiled in Figures 9–11. Classes and their corresponding ranges, as summarized in Table 4, are indicated with yellow annotations and horizontal lines. The measurement IDs keep track of the pass number and the specific region for each measurement: L1, L2, and L3. The experimental design produced a satisfactory range of values, providing a suitably large output space for neural network training. Measurements tend to form triplets, corresponding to the three length segments of the specimens for a given pass. The data are insufficient to assess the statistical significance of any observed measurement variations along the length of the specimen. At this stage, they are considered within measurement uncertainty for hand-operated instruments.

Table 4. Labels for all metrics with corresponding ranges.

Metric	Label	Range (μm)
Roughness Ra	Class 1	[0, 2.5)
	Class 2	[2.5, 4.5)
	Class 3	[4.5, 10]
Roughness Rz	Class 1	[0, 10)
	Class 2	[10, 16)
	Class 3	[16, 20]
Diameter deviation Ddev	Class 1 Under	[−20, 0)
	Class 1 Over	[0, 20)
	Class 2 Over	[20, 80]
All metrics	Not Cutting	N/A

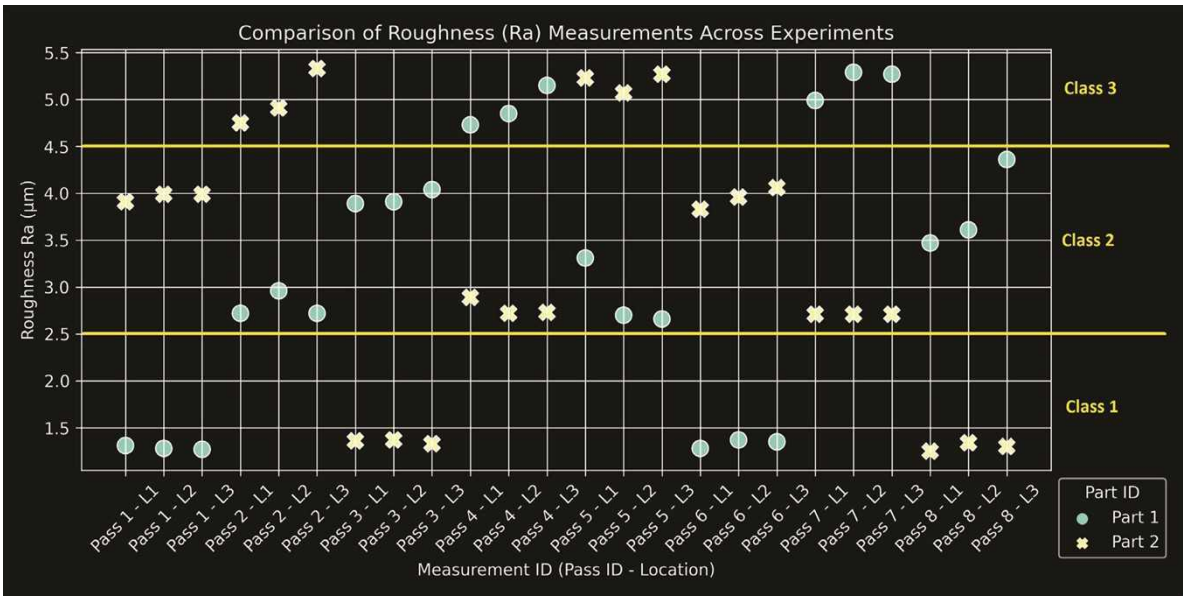


Figure 9. Compilation of Ra measurements with class ranges.

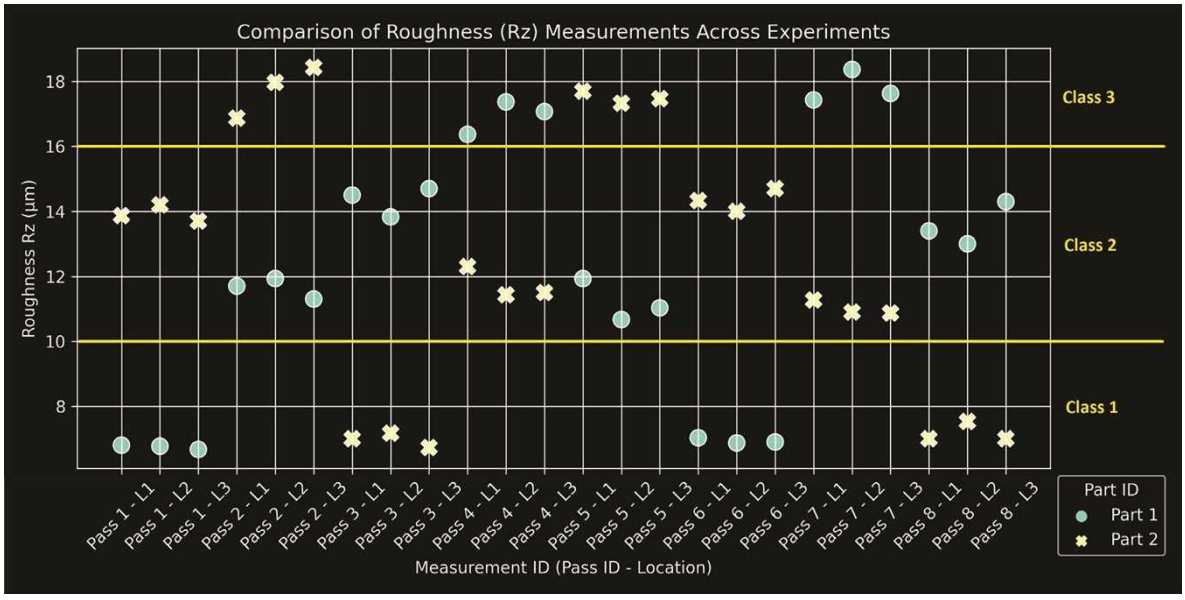


Figure 10. Compilation of Rz measurements with class ranges.

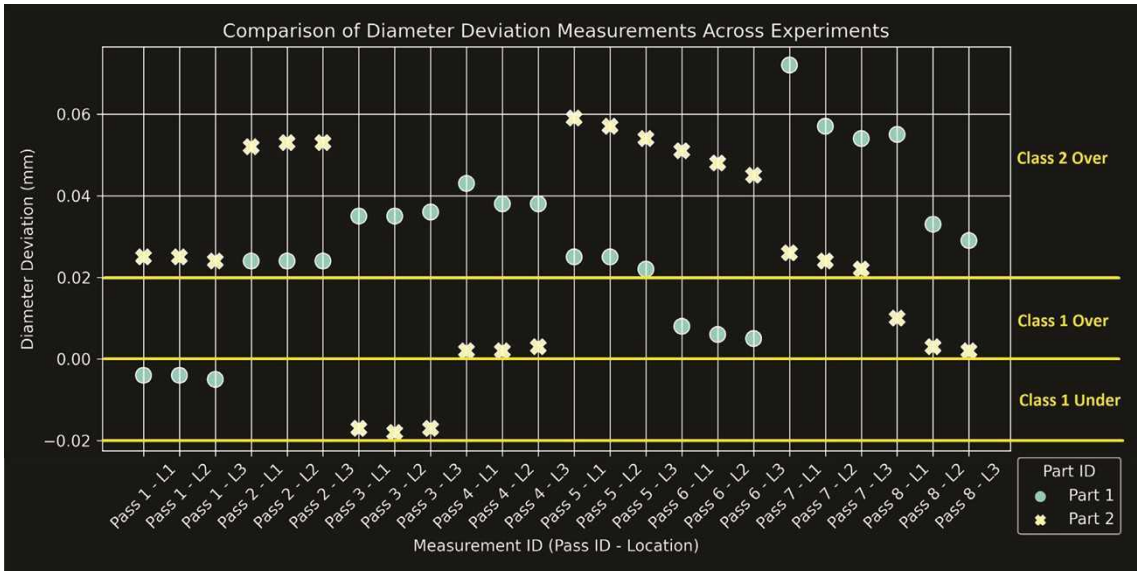


Figure 11. Compilation of Ddev measurements with class ranges.

3. Neural Network Design

3.1. Data Preparation for Neural Network Training

Convolutional Neural Networks have a fixed input length. Furthermore, the developed model should provide inference during the actual turning process. In order to satisfy both conditions, a sliding window strategy is employed to create the final input–output pairs for neural network training.

Firstly, the quality measurements are categorized into specific classes with corresponding labels, as detailed in Table 4. For the two roughness metrics, three classes of increasing value ranges were created, denoted as Class 1, Class 2, and Class 3. For the diameter deviation, two tight classes at either side of zero deviation were created: Class 1 Under, for diameters smaller than the nominal, and Class 1 Over, for diameters larger than the nominal but within a cutoff that mirrors the negative threshold. A third class, Class 2 Over, encompasses all other greater positive deviations. Lastly, a common label is introduced to each metric to denote non-cutting regions. This additional class is crucial for the neural network's training, as it includes free axes movements, which are a standard part of any process plan. Consequently, the network can learn to differentiate between cutting phases (where a specific quality label is expected) and Not Cutting phases. Thus, each metric ends up with four classes, leading to an imbalanced multiclass classification problem.

Since there are currently no specifications tied to the quality metrics, the proposed class definitions were based on the following rationale: for Ra and Rz, class ranges are determined with the intent of delineating 'good', 'normal' and 'bad' quality levels. Of course, the range and combinations of process parameters correspond to roughing conditions, meaning that the measured metrics do not dictate the final quality of the part. Thus, within this context, 'good' quality may suggest that more aggressive conditions could be feasible. Conversely, 'bad' quality might indicate roughness so significant that it could adversely affect a subsequent finishing pass. Although this is speculative and beyond the scope of the study, it does offer a potential perspective. Consequently, Class 1 is narrow in both range and representation. Class 2 is deliberately broader to capture the most common range of measurements. Class 3, equal in representation to Class 1, contains the most extreme results. A similar thought process is behind the diameter deviation class assignment, with the additional consideration that negative values are now possible. This fact naturally leads to two mirrored classes for minor deviations around zero: Class 1 Under and Class 1 Over. These classes could both represent acceptable deviations, though distinctions between over and under sizing may be crucial for specific applications, and the model is capable of making this differentiation. All larger deviations are categorized into Class 2 Over, in order

to maintain the same number of classes as for the other two metrics. These class definitions are used in the subsequent results analysis. In Section 4.5, a non-engineered system with equally distributed classes will also be examined.

Having established the class ranges, the actual values of measurements are transformed into the corresponding labels. These labels are then repeated along the signal timeline in accordance with the mapping described in Table 4. Outside of the cutting regions, the label Not Cutting is consistently applied. The final step is to create a sliding window, characterized by its length and a step size that allows for overlap between successive window placements. As the window slides along the signals, it generates input–output pairs for the neural network: the portion of the signals contained within the window serves as the input, and the dominant label for each metric is assigned as the output. Each of these pairs constitutes a single data point. Considering the dominant label as the output handles cases where the window inadvertently spans multiple classes, as is the case, for example, when transitioning from a non-cutting to a cutting region.

This sliding window technique enables the neural network to function as a process monitoring tool. Furthermore, it augments the dataset size. Given the 1 kHz sampling rate, window and step sizes will be expressed in samples, directly corresponding to milliseconds (1 ms per sample). While the window size determines the input size of the neural network, the step size is responsible for the magnitude of this augmentation, as shown in Table 5. For instance, a window size of 500 samples and a step size of 400 samples yield 289 data points from 16 experiments, reducing the step size to 100 samples results in 1153 data points. Both window and step sizes should be tailored to the specific characteristics of each application. In the case presented, a window size of 500 samples and a step size of 100 samples will be the default, with further exploration detailed in Section 4.3. The crucial point to note is that this augmentation allows the model to be effectively trained using data from just two machined parts.

Table 5. Size of dataset for 500 samples window and various step sizes.

Step Size (Samples)	Number of Data Points
100	1153
250	461
400	289

3.2. Neural Network Architecture

Typical classification problems have a single output layer. For binary classification, this layer has one output node with a sigmoid activation function, which outputs values from 0 to 1, directly corresponding to the probability of the positive class. In a multiclass problem, the output layer has as many nodes as there are classes, utilizing a single softmax activation. This function converts the combined output of these nodes into a probability distribution, representing the likelihood of each class. For the three metrics of interest—Ra, Rz, and Ddev—three such models could be trained, each monitoring a specific metric. However, this study adopts a more general approach, creating a multi-output, multiclass neural network capable of handling all metrics simultaneously. This requires three separate output layers, each with its own softmax activation. Having a single model predict all three metrics is more challenging, but it offers a versatile framework which covers the cases where either one or multiple metrics are of interest. Additionally, this approach simplifies deployment and enhances inference efficiency, as running a single multi-output model is more time-efficient than running three separate models.

The proposed neural network architecture is shown in Figure 12. The shape of the input layer is determined by the window size and the number of signal channels. For example, for a window size of 500 samples and the eight monitored channels (see Table 1), a single input instance to the model is an array of shape 500×8. Technically, this forms a tensor of shape $N \times 500 \times 8$, where N denotes the batch size. This structure is utilized in

neural networks both during training and, when feasible, in inference, to process multiple inputs in parallel.

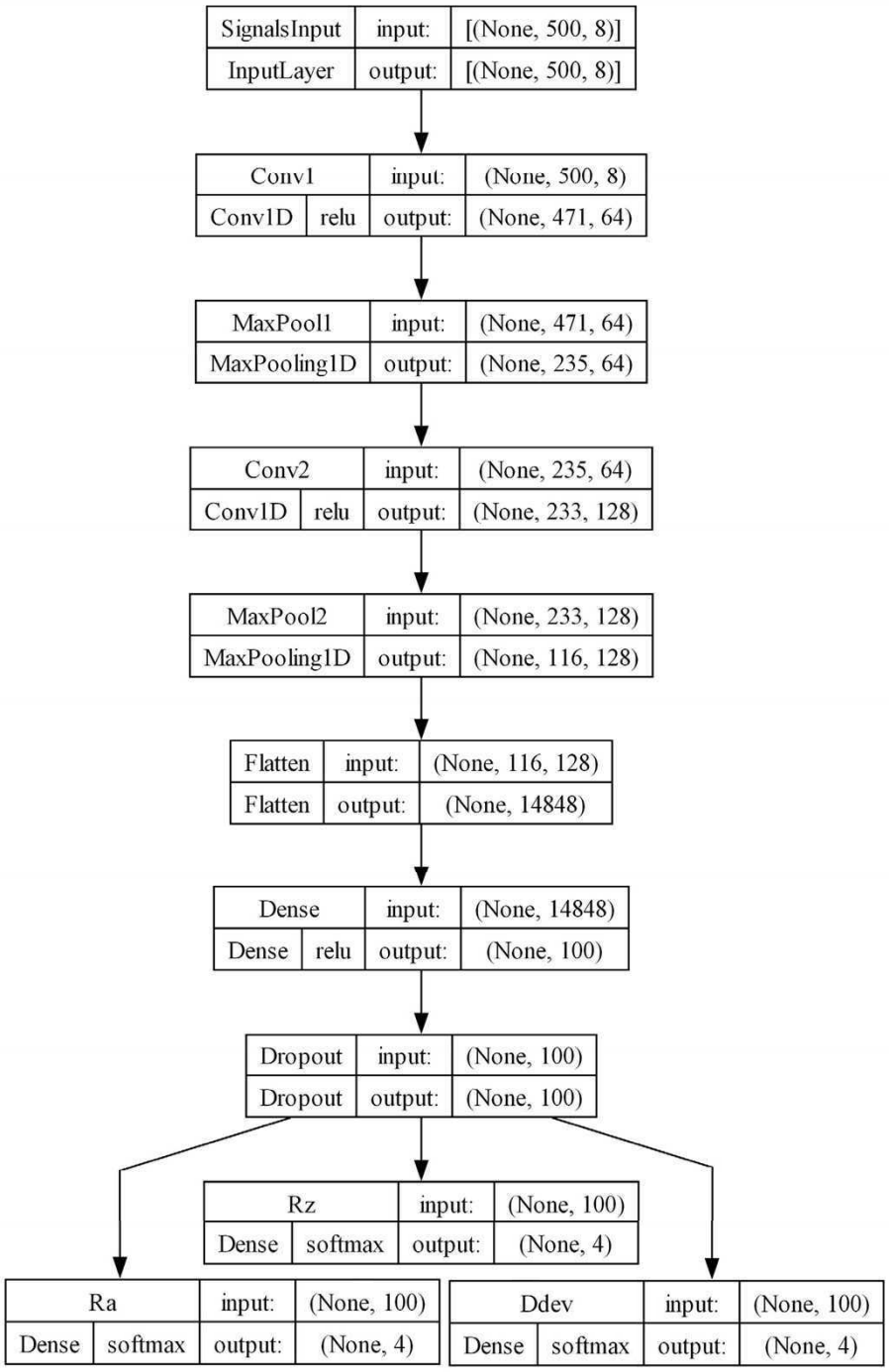


Figure 12. Neural network architecture for window size of 500 samples and 8 signal channels.

The input is passed through two convolutional layers with kernel sizes 30 and 3, respectively. The first kernel is relatively large in order to smooth out noise and capture more long-term features in the signals, as suggested by Zhang et al. [35]. The second is of more typical size and it synthesizes first-order features into more complex representations. After each convolution, a max pooling layer reduces the spatial dimensionality and enhances feature robustness by focusing on dominant features. The output from the second max pooling layer is flattened and passed through a dense layer, followed by a dropout layer. This mitigates overfitting by not allowing the network to become excessively dependent

on specific nodes. Lastly, three outputs with softmax activations give the probability distribution of classes for each of the metrics of interest. The parameters for all layers are summarized in Table 6. The model was implemented in TensorFlow™ (v2.10.1) using the Keras API.

Table 6. Neural network layer parameters.

Layer	Parameters
Conv1	kernel 30, filters 64, stride 1
MaxPool1	kernel 2, stride 1
Conv2	kernel 3, filters 128, stride 1
MaxPool2	kernel 2, stride 1
Flatten	-
Dense	size 100
Dropout	rate 0.2
Output X3	3 × size 4
Total trainable parameters: ~1.5 M	

Depending on the application and the complexity of features in the monitored inputs, more convolutional layers can easily be added, as well as additional dense layers before the outputs. Furthermore, the size of the first, large kernel may be adjusted based on the sampling rate of the signals. For signals sampled in the tens of kilohertz range, it might need to be larger, and conversely smaller for signals sampled at lower rates. Such an investigation is outside the scope of this study, which employs a fixed sampling rate of 1 kHz. The architecture shown represents a lightweight and robust baseline model, which demonstrates strong performance for the task at hand.

3.3. Model Training Parameters

Several instances of the model will be trained to thoroughly examine its performance under various scenarios. This section establishes the parameters common to all training instances, unless otherwise specified in the respective sections. These parameters are summarized in Table 7. The detailed results for each examined scenario are presented in Section 4.

Table 7. Default model training parameters.

Parameter	Value
Window size	500 samples
Step size	100 samples
Optimizer	Adam
Learning rate	0.0005
Loss	Categorical cross-entropy
Batch size	128
Max epochs	100
Validation split	0.15 of training
Patience	10 epochs
Min delta	0.001
Test split	0.15 of total
Split random state	42

4. Results and Discussion

4.1. K-Fold Cross Validation

Before presenting detailed performance metrics for specific instances of the model, k-fold cross-validation is employed to validate the modeling approach and the network architecture. Instead of using the validation/test split of Table 7, the entire dataset is split into 10 sets. The model is trained 10 times, each time withholding a different set from

the training process, which is subsequently used for testing the model's generalization performance. This strategy mitigates the risk of bias introduced by randomly selecting a favorable test set. Since there is no explicit validation set, early stopping is not employed, and each model is trained for 50 epochs. Models are evaluated by their F1 score, which is the harmonic mean of precision and recall. Results from the cross-validation for all three quality metrics are summarized in Table 8.

Table 8. *k*-fold cross validation results ($k = 10$).

Metric	F1 Macro Average										
	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Fold 6**	**Fold 7**	**Fold 8**	**Fold 9**	**Fold 10**	**Average**
Ra	0.94	0.97	0.98	1.00	0.97	0.98	0.99	0.95	1.00	0.98	97.76
Rz	0.98	0.97	0.97	1.00	0.96	0.98	1.00	0.97	1.00	0.99	98.34
Ddev	0.97	0.97	1.00	0.96	0.98	1.00	0.99	0.97	1.00	1.00	98.46

The average F1 score across all folds is above 0.97 for all three metrics. The lowest F1 scores for Ra, Rz, and Ddev are 0.94, 0.96, and 0.96, respectively. While some variation in the F1 scores is expected, the consistently high scores across all metrics suggest that the model is robust and performs well on the available data.

4.2. Detailed Performance Metrics

To produce visualizations and detailed metrics for the performance of the model, a specific instance is trained with explicit validation and test sets, according to Table 7. For reproducibility, the split random state for the test set is set to 42. Early stopping is activated with 10 epochs patience for improvement on the validation set and a minimum delta of 0.001. A plot of the training and validation loss is shown in Figure 13. Detailed performance metrics are summarized in Table 9.

The model exhibits high precision and recall, within the anticipated margins based on the preceding cross validation. This results in a minimal number of misclassifications, i.e., only 1 out of 173 total classifications for each class, as illustrated in the corresponding confusion matrices shown in Figure 14.

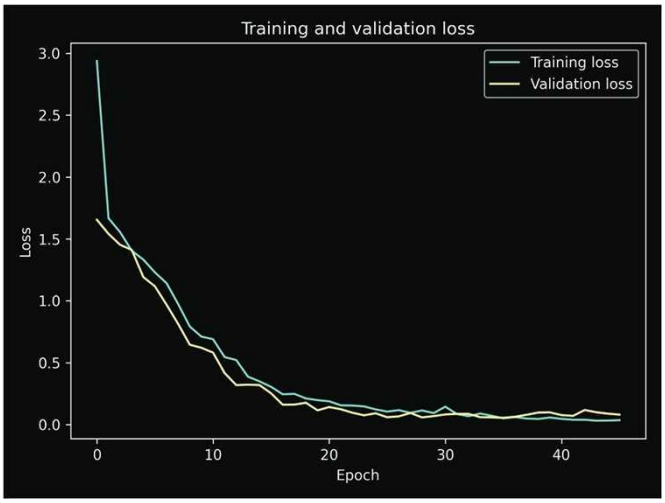
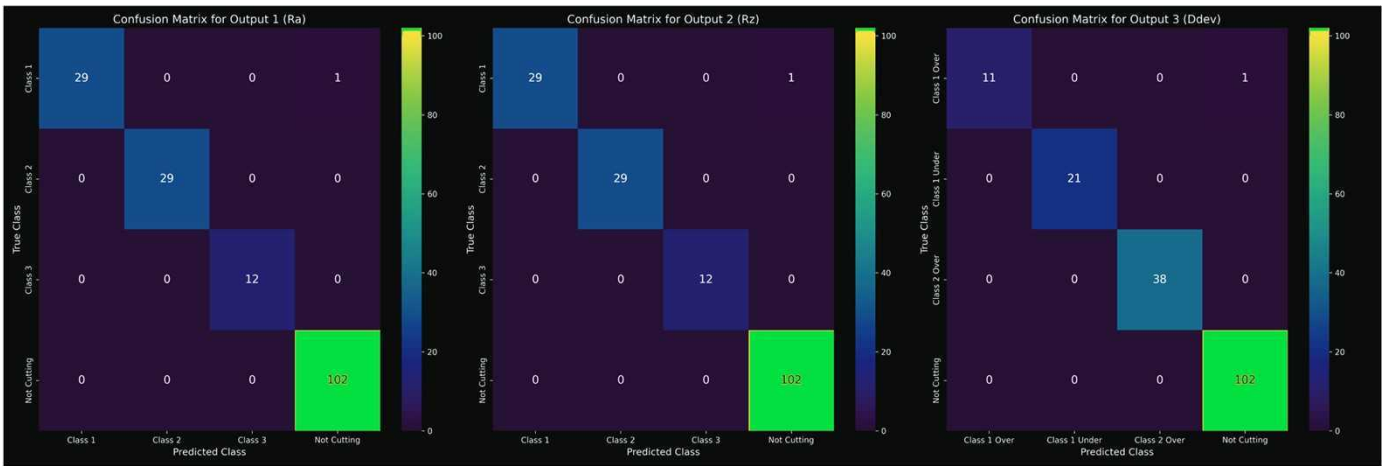


Figure 13. Training and validation loss.

Table 9. Model performance metrics.

Metric	Label	Precision	Recall	F1-Score	Support
Ra	Class 1	1.00	0.97	0.98	30
	Class 2	1.00	1.00	1.00	29
	Class 3	1.00	1.00	1.00	12
	Not Cutting	0.99	1.00	1.00	102
Rz	Class 1	1.00	0.97	0.98	30
	Class 2	1.00	1.00	1.00	29
	Class 3	1.00	1.00	1.00	12
	Not Cutting	0.99	1.00	1.00	102
Ddev	Class 1 Under	1.00	1.00	1.00	21
	Class 1 Over	1.00	0.92	0.96	12
	Class 2 Over	1.00	1.00	1.00	38
	Not Cutting	0.99	1.00	1.00	102



(**a**) (**b**) (**c**)

Figure 14. Confusion matrices: (**a**) Ra; (**b**); Rz; (**c**) Ddev.

4.3. Sliding Window Parameters

The sliding window strategy introduces a form of data augmentation by essentially duplicating sub-segments of signals for successive data points on the signal timeline. This may introduce a positive bias in the testing performance, by having specific sub-segments present in both the training and test sets. On the other hand, increasing the step size reduces the amount of data available for training and decreases the representation of each label in testing. Therefore, a comparison of performance for varying step sizes is not straightforward. To alleviate the diminishing of dataset sizes, validation during training is omitted, and the corresponding data are used for training instead. The models are then trained for 50 epochs for various step sizes. To normalize the differences in support during testing, the F1 weighted average score is utilized, which takes into account the representation of each label in the test set. The window size is kept at 500 samples. Results are given in Table 10.

Table 10. Model performance for varying step size and window size of 500 samples.

Step Size	F1 Weighted Average			Data Points (Train/Test)
	Ra	Rz	Ddev	
200	0.99	1.00	0.99	490/87
300	0.97	0.97	0.95	327/58
400	1.00	0.98	0.92	245/44
500	0.91	0.91	0.85	196/35

There is a noticeable trend that as the step size increases, the F1 scores tend to decrease. It is inconclusive whether this is due to the smaller training sets, which may not provide the model with sufficient variability and quantity of data to learn effectively, or to the reduced overlap of data. However, it is noteworthy that for a step size of 500 samples, corresponding to zero overlap of data, the model's performance could still be considered satisfactory, even with an extremely small training set of less than 200 data points.

To further test the robustness of the model against the sliding window parameters, several instances are trained for varying window sizes. The step size is kept constant at 100 samples, which results in very similar dataset sizes. Results are summarized in Table 11. No noticeable trend emerges, the observed variations being within the expected margin for the inherent randomness in training. Thus, the model exhibits the ability to accommodate various window sizes, depending on the requirements of a given application. Notably, for window and step size equal to 100 samples, which again corresponds to zero overlap of sub-segments but with a sizeable dataset this time, the model's performance remains extremely high. Therefore, whether overlap exists or not, it does not appear to have a significant impact on the model's performance. As anticipated, the dataset size is a much more influential factor.

Table 11. Model performance for varying window size and step size of 100 samples.

Window Size	F1 Macro Average			Data Points (Train/Test)
	Ra	Rz	Ddev	
100	0.98	0.98	0.98	986/175
200	0.98	0.98	0.98	985/174
400	0.97	0.98	0.98	981/174
600	0.99	1.00	0.99	978/173
800	1.00	1.00	0.99	974/173
1000	0.97	0.97	0.99	971/172

4.4. Reduced Input

So far, all eight of the available signal channels were used as inputs to the model. However, depending on the built-in instrumentation of a given machine tool and the availability of external sensors, alternate use cases may have access to a subset of process data. Testing all possible combinations of potential input channels would be impractical, therefore some logical subsets are tested. Additional interesting combinations arising from the initial results are also examined, all summarized in Table 12.

Table 12. Model performance for reduced input.

Subset of Channels	F1 Macro Average		
	Ra	Rz	Ddev
Only vibrations	0.80	0.76	0.71
Only torques	0.94	0.94	0.96
Only velocities	0.99	0.99	0.93
Only Z velocity	0.89	0.90	0.79
Z velocity and all torques	0.99	0.99	0.98

The worst model performance is observed when relying solely on vibrations. This can be attributed, to an extent, to the placement of the accelerometers behind the turret, resulting in a weakened signal with a correspondingly lower signal-to-noise ratio that is also potentially contaminated with natural frequencies of various machine components as explained in Section 2.1. The high performance of torques on the diameter deviation metric can be attributed to their correlation with cutting forces, which are primary factors for part deflection. On the other hand, the high performance of velocities on roughness metrics highlights their strong dependence on the feed parameter. To elaborate further,

since the turning is purely longitudinal, the feed in the z-direction should be the most critical velocity. Indeed, using only the Z velocity signal yields very good performance, considering it involves using only one channel, and this performance surpasses that of any other single channel that was tested. Finally, combining torques and Z velocity provides a candidate for the best minimum set of inputs, with performance on par with the full set.

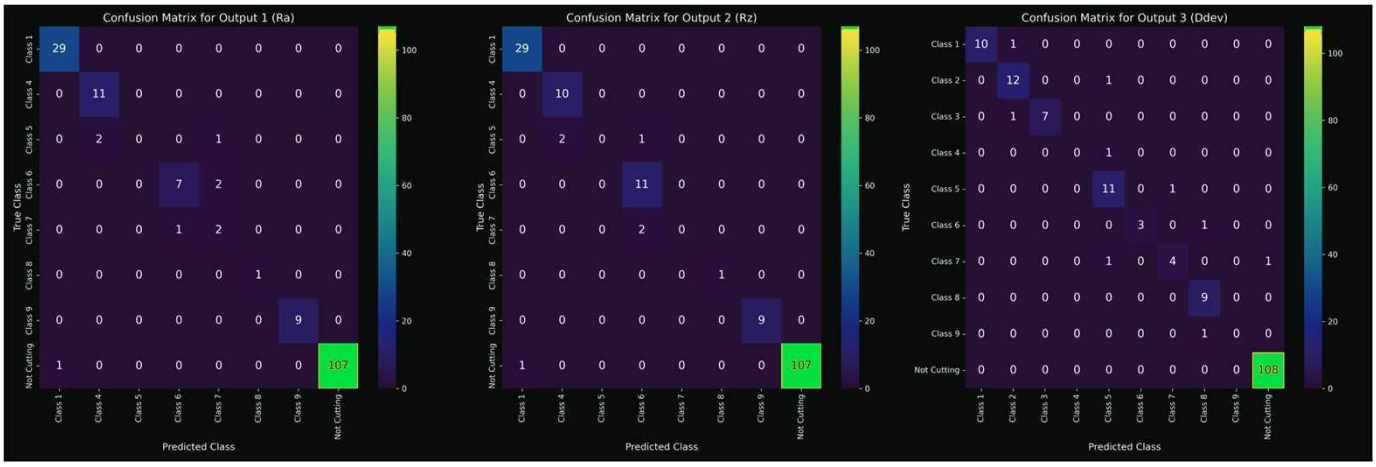
4.5. Alternative Definition of Classes

The rationale behind the definition of classes for the quality metrics was discussed in Section 3.1. In this section, the performance of the model is examined for a less 'engineered' class system, which simply has a number of equally distributed classes in the observed range of measurements for each metric. This offers the chance to test the model with a different number of classes as well. The results are summarized in Table 13. The '+1' in the number of classes denotes the common class 'Not Cutting.' Both the F1 macro and the weighted average are reported since as the number of classes increases, some are left with extremely low support, making the weighted average potentially more descriptive.

Table 13. Model performance for equally distributed classes.

Number of Equally Distributed Classes	F1 Macro Average			F1 Weighted Average		
	Ra	Rz	Ddev	Ra	Rz	Ddev
2 + 1	0.97	0.99	0.98	0.98	0.99	0.99
3 + 1	0.91	0.98	0.97	0.95	0.99	0.98
5 + 1	0.79	0.89	0.98	0.94	0.96	0.99
7 + 1	0.74	0.69	0.64	0.97	0.94	0.91
9 + 1	0.78	0.72	0.71	0.95	0.95	0.94

Both F1 scores are extremely high across all metrics for 2 + 1 and 3 + 1 classes. As the number of classes increases from 5 + 1 and above, the F1 macro scores tend to drop. This is expected, as the size of the dataset is not large enough to accommodate adequate learning examples for so many classes. Furthermore, the support for some classes in the testing set ends up too low, even zero in some cases, leading to ill-defined F1 scores. In cases like these, a single misclassification can dramatically change the macro average. This is reflected in the weighted averages, which maintain higher values. The confusion matrices for the 9 + 1 case are shown in Figure 15. For the two roughness metrics, Classes 2 and 3 have zero instances in the entire dataset, including the training and testing sets, and thus are not included in the confusion matrices. All 10 classes are represented for diameter deviation. Despite the large number of classes for the given dataset size, the performance of the model is deemed very satisfactory.



(**a**) (**b**) (**c**)

Figure 15. Confusion matrices for 9 + 1 equally distributed classes: (**a**) Ra; (**b**) Rz; (**c**) Ddev.

4.6. Datastream Simulation

The proposed model is designed to be lightweight in order to run during the actual turning process. Pure inference times are very low, either on the CPU or GPU of a typical home PC, clocking at under 20 ms. This inference time does not account for data transfer from and to memory, which is typically not an issue when performing batch processing, where data are transferred once, and inference is carried out for the entire batch in the reported times. During real operation however, the situation is different. Batch processing is not an option since data are being generated in real time from the machine, therefore the model needs to process a single data point at a time, which is extremely inefficient for a neural network. The extra overhead in this case is comparable to the inference time, resulting in a total processing time for a data point closer to 50 ms.

In order to verify the viability of the model in a realistic operation scenario, a datastream simulation is set up. The signal history of the first specimen is used as input. All data points are prepared according to the sliding window strategy, but the model is only allowed to process each of them at specified time intervals, simulating the real data acquisition procedure with a sampling rate of 1 kHz. First, the model has to wait for 500 ms for the first 500 samples window to fill. For each subsequent inference step, the wait time is 100 ms, corresponding to the step size of 100 samples. For each inference step, the exact processing time is calculated and subtracted from the wait time, simulating the continued data acquisition during model inference. The lag of the model versus the real machining time is plotted in Figure 16.

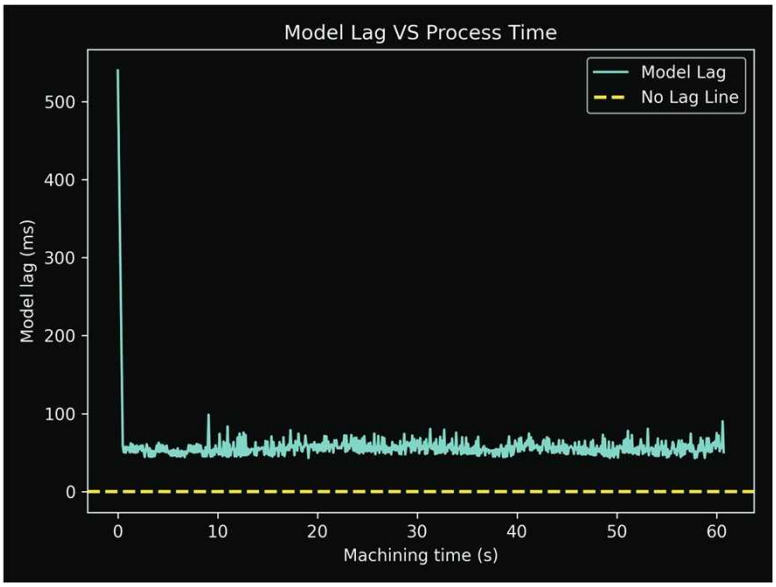


Figure 16. Simulation of model lag VS real process time.

As anticipated, for a total processing time for each data point less than the step size of the sliding window, the model does not drift away from the machining process. After an initial large lag which corresponds to the filling of the first window, the model quickly settles a few tens of milliseconds behind the real process. Random lag spikes may occur, due to arbitrary processes running on the test PC (i5 class CPU, 1060-3GB GPU, 16 GB RAM), however, the model quickly returns to its usual slight delay behind the process. This is presumed to represent an unfavorable scenario where a single-threaded Python program is responsible for performing the datastream simulation, including data transfer, time keeping, and inference. Thus, the model should not drift as long as the sliding window step size is larger than the average processing time of each inference step. Furthermore, there are techniques to further minimize processing times in a real application, such as deployment in optimized hardware environments or using more efficient computing frameworks, which are out of the scope of the present work.

5. Conclusions and Future Work

This study presented an innovative application of 1D convolutional neural networks for raw signal classification in CNC process monitoring. The proposed methodology eschews manually engineered feature extraction, typically employed in signal processing, in favor of temporal pattern identification by the neural network itself. Convolutional layers identify and synthesize pertinent, process-related information contained in multiple signal channels acquired from the machine tool, in order to classify the machining process with respect to its resulting quality. It was shown that with an extremely small number of experiments, in full compliance to industry expectations, and a sliding window strategy, a multi-output, multiclass model can be successfully trained to monitor several quality metrics at once. The performance of the model was investigated under various combinations of design parameters and constraints, and it proved to be a robust base for real-time monitoring and quality control in CNC machining. Furthermore, this approach provides a versatile alternative for the classification of high-rate signals across various industrial domains where such analysis is relevant.

To build upon the presented work, several potential avenues are available. Conducting more experiments will result in greater coverage of the input space of process conditions, as well as the output space of quality metrics. Moreover, additional experiments will increase the raw size of the dataset, which is generally beneficial to model training. Including finishing conditions in the experimental design will offer a complete view of a typical real machining application. In this context, it will be interesting to investigate the potential adverse effects of an aggressive roughing pass on a subsequent finishing pass. This could be valuable in the context of a process optimization framework, which can suggest efficient process planning while ensuring final quality. Further considerations relating to tool wear and the sustainability of the process can also be explored in this light.

Finally, testing the model during real machining operations presents its own set of challenges, associated with model deployment, establishing communication protocols, and ensuring seamless integration into the existing workflow. Addressing these challenges is crucial for the successful application of the model in a live industrial setting.

Author Contributions: Conceptualization, E.S.; methodology, E.S.; experiments, E.S. and E.T.; software, E.S. and E.T.; validation, E.S., E.T. and P.B.; investigation, E.S., E.T., G.-C.V. and P.B.; data curation, E.S. and E.T.; writing—original draft preparation, E.S.; writing—review and editing, G.-C.V., P.B. and E.T.; visualization, E.S.; supervision, G.-C.V. and P.B.; project administration, G.-C.V.; funding acquisition, G.-C.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the European Union under Grant Agreement No. 101091783, project 'MARS: Manufacturing Architecture for Resilience & Sustainability', HORIZON-CL4-2022-TWIN-TRANSITION-01-03.

Data Availability Statement: This study is part of an ongoing EU funded research project and related data cannot be made public at this time.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jimeno-Morenilla, A.; Azariadis, P.; Molina-Carmona, R.; Kyratzi, S.; Moulianitis, V. Technology Enablers for the Implementation of Industry 4.0 to Traditional Manufacturing Sectors: A Review. *Comput. Ind.* **2021**, *125*, 103390. [CrossRef]
2. Ching, N.T.; Ghobakhloo, M.; Iranmanesh, M.; Maroufkhani, P.; Asadi, S. Industry 4.0 Applications for Sustainable Manufacturing: A Systematic Literature Review and a Roadmap to Sustainable Development. *J. Clean. Prod.* **2022**, *334*, 130133. [CrossRef]
3. Luo, W.; Hu, T.; Zhang, C.; Wei, Y. Digital Twin for CNC Machine Tool: Modeling and Using Strategy. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1129–1140. [CrossRef]
4. Tao, F.; Zhang, M.; Liu, Y.; Nee, A.Y.C. Digital Twin Driven Prognostics and Health Management for Complex Equipment. *CIRP Ann.* **2018**, *67*, 169–172. [CrossRef]
5. Lee, J.; Bagheri, B.; Kao, H.A. A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems. *Manuf. Lett.* **2015**, *3*, 18–23. [CrossRef]

6. Söderberg, R.; Wärmefjord, K.; Carlson, J.S.; Lindkvist, L. Toward a Digital Twin for Real-Time Geometry Assurance in Individualized Production. *CIRP Ann.-Manuf. Technol.* **2017**, *66*, 137–140. [CrossRef]
7. Taha, H.A.; Yacout, S.; Shaban, Y. Autonomous Self-Healing Mechanism for a CNC Milling Machine Based on Pattern Recognition. *J. Intell. Manuf.* **2023**, *34*, 2185–2205. [CrossRef]
8. Guo, W.; Wu, C.; Ding, Z.; Zhou, Q. Prediction of Surface Roughness Based on a Hybrid Feature Selection Method and Long Short-Term Memory Network in Grinding. *Int. J. Adv. Manuf. Technol.* **2021**, *112*, 2853–2871. [CrossRef]
9. Elsheikh, A.; Yacout, S.; Ouali, M.S.; Shaban, Y. Failure Time Prediction Using Adaptive Logical Analysis of Survival Curves and Multiple Machining Signals. *J. Intell. Manuf.* **2020**, *31*, 403–415. [CrossRef]
10. Zhao, R.; Yan, R.; Chen, Z.; Mao, K.; Wang, P.; Gao, R.X. Deep Learning and Its Applications to Machine Health Monitoring. *Mech. Syst. Signal Process.* **2019**, *115*, 213–237. [CrossRef]
11. Du, C.; Ho, C.L.; Kaminski, J. Prediction of Product Roughness, Profile, and Roundness Using Machine Learning Techniques for a Hard Turning Process. *Adv. Manuf.* **2021**, *9*, 206–215. [CrossRef]
12. García Plaza, E.; Núñez López, P.J.; Beamud González, E.M. Efficiency of Vibration Signal Feature Extraction for Surface Finish Monitoring in CNC Machining. *J. Manuf. Process.* **2019**, *44*, 145–157. [CrossRef]
13. Wang, B.; Wei, Y.; Liu, S.; Gu, D.; Zhao, D. Intelligent Chatter Detection for CNC Machine Based on RFE Multi-Feature Selection Strategy. *Meas. Sci. Technol.* **2021**, *32*, 095904. [CrossRef]
14. Yao, Z.; Mei, D.; Chen, Z. On-Line Chatter Detection and Identification Based on Wavelet and Support Vector Machine. *J. Mater. Process. Technol.* **2010**, *210*, 713–719. [CrossRef]
15. Xu, Y.; Gui, L.; Xie, T. Intelligent Recognition Method of Turning Tool Wear State Based on Information Fusion Technology and BP Neural Network. *Shock Vib.* **2021**, *2021*, 7610884. [CrossRef]
16. von Hahn, T.; Mechefske, C.K. Machine Learning in CNC Machining: Best Practices. *Machines* **2022**, *10*, 1233. [CrossRef]
17. Li, G.; Fu, Y.; Chen, D.; Shi, L.; Zhou, J. Deep Anomaly Detection for CNC Machine Cutting Tool Using Spindle Current Signals. *Sensors* **2020**, *20*, 4896. [CrossRef] [PubMed]
18. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef] [PubMed]
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5999–6009.
20. Lecun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
21. Zhang, K.; Zhou, D.; Zhou, C.; Hu, B.; Li, G.; Liu, X.; Guo, K. Tool Wear Monitoring Using a Novel Parallel BiLSTM Model with Multi-Domain Features for Robotic Milling Al7050-T7451 Workpiece. *Int. J. Adv. Manuf. Technol.* **2023**, *129*, 1883–1899. [CrossRef]
22. Yang, J.; Wu, J.; Li, X.; Qin, X. Tool Wear Prediction Based on Parallel Dual-Channel Adaptive Feature Fusion. *Int. J. Adv. Manuf. Technol.* **2023**, *128*, 145–165. [CrossRef]
23. Krichen, M. Convolutional Neural Networks: A Survey. *Computers* **2023**, *12*, 151. [CrossRef]
24. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
25. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; Volume 45, pp. 770–778.
27. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv* **2017**, arXiv:1712.01815.
28. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
29. Ibarra-Zarate, D.; Alonso-Valerdi, L.M.; Chuya-Sumba, J.; Velarde-Valdez, S.; Siller, H.R. Prediction of Inconel 718 Roughness with Acoustic Emission Using Convolutional Neural Network Based Regression. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 1609–1621. [CrossRef]
30. Kuo, P.H.; Tseng, Y.R.; Luan, P.C.; Yau, H.T. Novel Fractional-Order Convolutional Neural Network Based Chatter Diagnosis Approach in Turning Process with Chaos Error Mapping. *Nonlinear Dyn.* **2023**, *111*, 7547–7564. [CrossRef]
31. Hanchate, A.; Bukkapatnam, S.T.S.; Lee, K.H.; Srivastava, A.; Kumara, S. Explainable AI (XAI)-Driven Vibration Sensing Scheme for Surface Quality Monitoring in a Smart Surface Grinding Process. *J. Manuf. Process.* **2023**, *99*, 184–194. [CrossRef]
32. Tran, M.Q.; Liu, M.K.; Tran, Q.V. Milling Chatter Detection Using Scalogram and Deep Convolutional Neural Network. *Int. J. Adv. Manuf. Technol.* **2020**, *107*, 1505–1516. [CrossRef]
33. Kounta, C.A.K.A.; Arnaud, L.; Kamsu-Foguem, B.; Tangara, F. Deep Learning for the Detection of Machining Vibration Chatter. *Adv. Eng. Softw.* **2023**, *180*, 103445. [CrossRef]
34. Unver, H.O.; Sener, B. A Novel Transfer Learning Framework for Chatter Detection Using Convolutional Neural Networks. *J. Intell. Manuf.* **2023**, *34*, 1105–1124. [CrossRef]

35. Zhang, W.; Peng, G.; Li, C.; Chen, Y.; Zhang, Z. A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals. *Sensors* **2017**, *17*, 425. [CrossRef]
36. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. A Deep Convolutional Neural Network with New Training Methods for Bearing Fault Diagnosis under Noisy Environment and Different Working Load. *Mech. Syst. Signal Process.* **2018**, *100*, 439–453. [CrossRef]
37. Kolar, D.; Lisjak, D.; Pająk, M.; Pavković, D. Fault Diagnosis of Rotary Machines Using Deep Convolutional Neural Network with Wide Three Axis Vibration Signal Input. *Sensors* **2020**, *20*, 4017. [CrossRef]
38. Wu, C.; Jiang, P.; Ding, C.; Feng, F.; Chen, T. Intelligent Fault Diagnosis of Rotating Machinery Based on One-Dimensional Convolutional Neural Network. *Comput. Ind.* **2019**, *108*, 53–61. [CrossRef]
39. Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufier, M.; Verstockt, S.; Van de Walle, R.; Van Hoecke, S. Convolutional Neural Network Based Fault Detection for Rotating Machinery. *J. Sound Vib.* **2016**, *377*, 331–345. [CrossRef]
40. Souza, R.M.; Nascimento, E.G.S.; Miranda, U.A.; Silva, W.J.D.; Lepikson, H.A. Deep Learning for Diagnosis and Classification of Faults in Industrial Rotating Machinery. *Comput. Ind. Eng.* **2021**, *153*, 107060. [CrossRef]
41. Zhang, P.; Gao, D.; Hong, D.; Lu, Y.; Wu, Q.; Zan, S.; Liao, Z. Improving Generalisation and Accuracy of On-Line Milling Chatter Detection via a Novel Hybrid Deep Convolutional Neural Network. *Mech. Syst. Signal Process.* **2023**, *193*, 110241. [CrossRef]
42. Lu, Y.; Ma, H.; Sun, Y.; Song, Q.; Liu, Z.; Xiong, Z. An Interpretable Anti-Noise Convolutional Neural Network for Online Chatter Detection in Thin-Walled Parts Milling. *Mech. Syst. Signal Process.* **2024**, *206*, 110885. [CrossRef]
43. Huang, P.M.; Lee, C.H. Estimation of Tool Wear and Surface Roughness Development Using Deep Learning and Sensors Fusion. *Sensors* **2021**, *21*, 5338. [CrossRef] [PubMed]
44. Lin, W.-J.; Lo, S.-H.; Young, H.-T.; Hung, C.-L. Evaluation of Deep Learning Neural Networks for Surface Roughness Prediction Using Vibration Signal Analysis. *Appl. Sci.* **2019**, *9*, 1462. [CrossRef]
45. *ISO 21920-2:2021*; Geometrical Product Specifications (GPS), Surface Texture: Profile Part 2: Terms, Definitions and Surface Texture Parameters. ISO: Geneva, Switzerland, 2021. Available online: https://www.iso.org/standard/72226.html (accessed on 15 November 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Experimental Analyses and Predictive Modelling of Ultrasonic Welding Parameters for Enhancing Smart Textile Fabrication

Mohamed Baraya [1], Mohamed S. El-Asfoury [1], Omnia O. Fadel [1] and Ahmed Abass [1,2,*]

[1] Department of Production Engineering and Mechanical Design, Faculty of Engineering, Port Said University, Port Fuad 42526, Egypt; mohamed.baraya@eng.psu.edu.eg (M.B.); mohamed.saad@eng.psu.edu.eg (M.S.E.-A.); omniaothman@eng.psu.edu.eg (O.O.F.)
[2] Department of Materials, Design and Manufacturing Engineering, School of Engineering, University of Liverpool, Liverpool L69 3GH, UK
* Correspondence: a.abass@liverpool.ac.uk

Abstract: This study aims to illustrate the design, fabrication, and optimisation of an ultrasonic welding (UW) machine to join copper wires with non-woven PVC textiles as smart textiles. The study explicitly evaluates UW parameters' impact on heat generation, joint strength, and electrical properties, with a comprehensive understanding of the process dynamics and developing a predictive model applicable to smart textiles. The methodological approach involved designing and manufacturing an ultrasonic piezoelectric transducer using ABAQUS finite element analyses (FEA) software and constructing a UW machine for the current purpose. The full factorial design (FFD) approach was employed in experiments to systematically assess the influence of welding time, welding pressure, and copper wire diameter on the produced joints. Experimental data were meticulously collected, and a backpropagation neural network (BPNN) model was constructed based on the analysis of these results. The results of the experimental investigation provided valuable insights into the UW process, elucidating the intricate relationship between welding parameters and heat generation, joint strength, and post-welding electrical properties of the copper wires. This dataset served as the basis for developing a neural network model, showcasing a high level of accuracy in predicting welding outcomes compared to the FFD model. The neural network model provides a valuable tool for controlling and optimising the UW process in the realm of smart textile production.

Keywords: ultrasonic welding; smart textiles; copper wire joining; artificial neural network (ANN); full factorial experimental design (FFD)

1. Introduction

Smart textiles, also known as e-textiles or smart fabrics, are a transformative category of materials with integrated electronic components that can sense, respond to, and communicate data [1]. These textiles have gained increasing importance in various fields due to their versatility and potential applications [2]. Smart textiles offer many benefits, from enhancing our daily lives to advancing industries and healthcare [1]. Their applications encompass wearables that monitor health and fitness, such as heart rate or temperature, and offer real-time feedback. In the fashion industry, they enable interactive clothing and accessories that change colour or pattern in response to environmental stimuli. Moreover, the military and sports industries are adopting smart textiles for improved performance and safety [3]. In healthcare, smart textiles assist in remote patient monitoring and developing prosthetics. Furthermore, the automotive and aerospace sectors utilise these materials to enhance passenger comfort and safety [4]. Smart textiles are also crucial in the Internet of Things (IoT), facilitating seamless data collection, transmission, and processing in smart homes, agriculture, and industrial automation [5].

Smart textiles are created through various fabrication methods, each with unique advantages and disadvantages. Traditional methods include embroidery and weaving, where

conductive threads are integrated into the textile structure [6,7]. Another method involves screen printing, where conductive ink is applied to fabric to create conductive patterns [8,9]. Emerging techniques include three-dimensional (3D) knitting and additive manufacturing, which enable intricate designs and precise integration of electronic components [10,11]. Embroidery and weaving provide durability and flexibility but may lack precision [12]. Screen printing offers cost-effectiveness and scalability but can be limited in conductivity and washability [13]. At the same time, 3D knitting and additive manufacturing provide high precision and customisation but may involve complex setups and higher production costs [10]. The choice of method depends on the specific application requirements, such as flexibility, conductivity, and production volume.

Therefore, UW stands out as an optimal method for smart textile fabrication due to its ability to create strong, conductive bonds without the need for adhesives or additional materials [14]. This technique uses high-frequency vibrations to create localised heat, securely bonding conductive materials. Unlike traditional methods, UW ensures high precision and reliability, making it ideal for integrating electronic components into textiles. Additionally, it offers advantages such as rapid processing, energy efficiency, and minimal material wastage [15]. UW has emerged as a promising technique for fabricating smart textiles, offering numerous advantages. However, several challenges exist in applying this method, which has sparked current research efforts to address these issues.

One primary challenge is achieving strong and reliable bonds between conductive elements and textile substrates. While UW can provide secure connections, ensuring consistent conductivity and durability across various textile materials and conductive elements is non-trivial [16]. Additionally, selecting suitable welding parameters, such as amplitude, pressure, and time, can be challenging. These parameters greatly affect the quality of welds, but their optimal values may vary based on the specific materials and design of the smart textile [17–20]. Another significant challenge is managing the localised heat generated during the UW process. Excessive heat can damage sensitive electronic components or the textile substrate itself, reducing the functionality and longevity of the smart textile. Moreover, ensuring the uniformity of the welds across large textile surfaces is another pressing challenge, as non-uniformity can lead to weak spots and reduced performance [21].

Furthermore, achieving flexibility and stretchability in smart textiles while utilising UW remains an ongoing research endeavour. Smart textiles are intended to conform to the body or textile structure while retaining functionality. Balancing the demands of strong, conductive bonds with the need for textile flexibility and comfort presents a challenge and material optimisation problem [22,23].

The optimisation of UW parameters plays a vital role in achieving high-quality products. It can be optimised through a comprehensive experimental approach, where pressure and time are systematically tested and analysed to determine the optimal settings for achieving strong and reliable welds. Backpropagation neural network (BPNN) modelling is a highly effective method used in artificial intelligence and machine learning for optimisation and other objectives [24]. With that approach, one can develop artificial neural networks whose internal parameters are optimised sequentially by minimising prediction errors. It starts with an initial model, and then through a process called backpropagation, the error is calculated and passed backwards; thus, the network can learn itself as a more complex pattern and predict from the new data [25]. Various fields, such as biomaterials and manufacturing, have also employed this modelling method [26,27]. This can present the importance of BPNN as an integral element of most modern machine learning schemes.

From the preceding narratives, it can be concluded that UW offers significant promise for smart textile fabrication, but several challenges persist in ensuring consistent performance, reliability, and comfort. Research efforts are directed toward optimising welding parameters, developing high-strength wire-fabric connections, and addressing materials and design challenges. As these challenges are addressed, the potential of smart textiles will be further unlocked for diverse applications. Therefore, the current investigation attempts

to fix some previously mentioned barriers concerning optimising the UW parameters to improve the smart fabrics' strength and electrical characteristics.

2. Methodology

Firstly, FEA modelling using the ABAQUS 6.14 software package (Dassault Systèmes Simulia Corp, Johnston, Rhode Island) has been employed to design an ultrasonic piezoelectric transducer, which acts as the heartbeat of the UW system. The FEA tool enabled the simulation and optimisation of the transducer's geometry to ensure efficient and precise ultrasonic energy generation. Subsequently, the design was fabricated by carefully selecting materials that aligned closely with the FEA model. The transducer was designed to generate consistent ultrasonic vibrations for the experimental work within the study.

After designing the ultrasonic transducer, the effort moved to designing and fabricating a specialised UW setup. This setup was tailored to the unique requirements of joining copper wires with non-woven PVC textiles. During the design phase, components such as the transducer, horn, and fixture were thoughtfully selected and strategically configured to ensure compatibility with the target materials. Furthermore, the setup was accurately constructed to allow for precise control of critical welding parameters, including copper wire diameter in mm, welding time in sec and pressure in bar. Three levels were specified for each of the three UW parameters as shown in Table 1. This precision became instrumental in achieving consistent weld quality and laid the groundwork for the subsequent comprehensive investigation into the effects of welding process parameters.

Table 1. UW parameters.

Parameters	Levels		
	−1	**0**	**+1**
Wire diameter (mm)	0.2	0.3	0.4
Pressure (bar)	0.4	0.6	0.8
Time (s)	5	8	10

In order to systematically investigate the sophisticated relationship between welding process parameters and process outcomes, the Design of Experiment (DOE) technique was utilised to conduct number of experiments using Minitab 16 software (Minitab, LLC, Lock Haven, PA, US). The full factorial DOE model comprised 3 parameters with 3 levels for each. A complete combination array consisting of $3^3 = 27$ trials was executed, and the weldability performance was recorded, as showcased in Table 2. This method proved efficient in exploring various parameter combinations while minimising the required experimental runs. The research aimed to collect robust data that could be statistically analysed for patterns and trends by varying welding parameters such as time, pressure, and copper wire diameter.

The experimental data obtained from the 27-array design became the foundation for the next phase. These data were employed to develop an interactive model using the backpropagation neural network (BPNN) technique. The BPNN model was trained on the experimental outcomes, transforming it into a predictive tool for process control and optimisation. Its capacity for learning and generalisation allowed it to offer accurate predictions regarding the relationships between welding parameters and critical factors, such as the heat generated (in terms of welding temperature), joint strength (in terms of peeling force), and post-welding electrical properties (in terms of electrical resistance) of the copper wires. This methodology, from transducer design to BPNN modelling, forged a path toward a comprehensive understanding of the UW process, with significant implications for smart textile applications.

Table 2. Full 27-parameters array for the main UW input and output responses.

Experiment No.	Input Welding Parameter			Output Response		
	Diameter (mm)	Pressure (bar)	Time (s)	Heat Generation (°C)	Joint Strength (N)	Electrical Properties (Ohm)
1	0.2	0.4	5	68	19.65	0.02
2	0.2	0.4	8	69	23.02	0.021
3	0.2	0.4	10	79	24.08	0.021
4	0.2	0.6	5	79	23.78	0.024
5	0.2	0.6	8	80	25.08	0.0242
6	0.2	0.6	10	80	25.91	0.0245
7	0.2	0.8	5	79	24.02	0.0252
8	0.2	0.8	8	79	26.08	0.0255
9	0.2	0.8	10	80	27.91	0.0256
10	0.3	0.4	5	73	18	0.04
11	0.3	0.4	8	74	19.6	0.041
12	0.3	0.4	10	74	22.4	0.041
13	0.3	0.6	5	73	20.5	0.044
14	0.3	0.6	8	74	22.1	0.045
15	0.3	0.6	10	77	23.8	0.045
16	0.3	0.8	5	76	20.2	0.048
17	0.3	0.8	8	78	21.53	0.048
18	0.3	0.8	10	78	23	0.05
19	0.4	0.4	5	69	16.32	0.06
20	0.4	0.4	8	70	18.25	0.061
21	0.4	0.4	10	70	19.03	0.061
22	0.4	0.6	5	70	18.95	0.062
23	0.4	0.6	8	71	19.05	0.061
24	0.4	0.6	10	73	20.01	0.061
25	0.4	0.8	5	72	19.95	0.064
26	0.4	0.8	8	73	21.03	0.065
27	0.4	0.8	10	74	22.01	0.065

3. Transducer FEA Modelling and Manufacturing

The first step in constructing the FEA model was to build the geometry of the transducer parts. Dimensions of each part of the transducer were calculated based on the acoustic wave equation [28]. After that, the ABAQUS package has been used to build the model geometry shown in Figure 1a. Initially, the front mass geometry was assumed to be stepped. The stepped horn was distinguished by the high amplification of vibration amplitude [28].



Figure 1. Transducer model (**a**) geometry and (**b**) boundary conditions.

Based on the literature, proper materials have been assigned to each transducer's parts. Aluminium has been used for the front and the back masses. Simultaneously, copper and piezoelectric materials have been allocated for the electrodes and the rings, respectively.

Table 3 presents material properties for defining the materials in the FEA model. Afterwards, loads and boundary conditions are assigned. The pre-stressed bolt load, estimated based on the recommended piezoelectric stack compression state (around 30 MPa), is applied at the bolt cross-section between the back mass and the piezoelectric stack. Boundary conditions (BC) are also used for piezoelectric elements, facilitating electrical potential propagation from the base state [29,30]. Figure 1b illustrates these load and boundary conditions.

Table 3. Physical properties of transducer component materials.

No.	Material	Young's Modulus (GPa)	Density (Kg/m^3)	Poisson's Ratio	Wave Velocity (m/s)	Characteristic Acoustic Impedance ($\times 10^6$ Ns/m^3)
1	Aluminium (5083)	70.3	2660	0.33	5140	13.67
2	Steel (AISI 1045)	200	7870	0.3	5040	39.7
3	Piezoelectric (PZT)	73	7700	-	3080	23.72
4	Copper (99.97% pure)	115	8900	0.31	3595	31.9

Regarding the FEA model meshing process, the structure meshing technique is preferred for such an electromechanical system [29,30]. Moreover, suitable elements for the different parts of the transducer were selected. All metallic parts were 3D stress quadratic elements (C3D20R), while the piezoelectric rings were a piezoelectric quadratic element (C3D20RE).

Concerning the FEA results, Figure 2a,b illustrates displacement distribution and von Mises stress along the transducer's axial direction for the longitudinal mode shape observed at 25.87 kHz. The nodal plane position, which supports the transducer during experiments, is also presented. A welding horn with a calculated length was attached to the working end to maximise output amplitude. It should be noted that changes in the transducer mass can influence the system's resonance frequency.

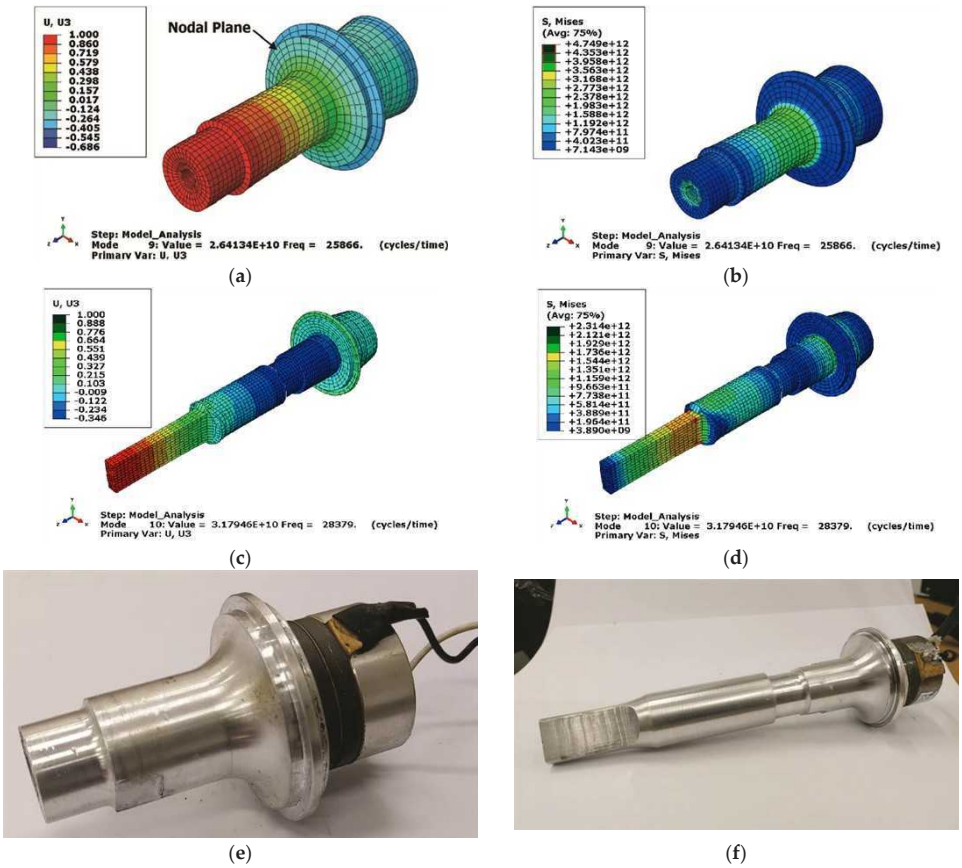


Figure 2. Along the axial direction of the transducer are the (**a**) normalised displacement, (**b**) von Mises stress in MPa, (**c**) normalised displacement transducer with ultrasonic horn, (**d**) von Mises stress in MPa transducer with ultrasonic horn, (**e**) fabricated transducer, and (**f**) fabricated transducer with ultrasonic horn.

Figure 2c,d shows displacement distribution, and von Mises stress along the transducer's axial direction for the longitudinal mode shape after adding the welding horn. It occurred at 28.38 kHz.

4. UW Machine and Experimental Process Design

In the current study, DOE is used to model and analyse the effect of the UW regime on the heat generation, joint strength, and electrical properties of welded joints for smart textiles. The general functional relationship can be written as in Equation (1):

$$Y = f(D, P, t) \tag{1}$$

where Y represents the dependent variable, while D, P, and t represent the independent variables of copper wire diameter, welding pressure, and welding time, respectively. The general functional relationship forms the basis of experimental design, defining factors and interactions. Analysis of Variance (ANOVA) assesses their significance once data are collected. Polynomial models, including linear and non-linear components, can be part of this relationship. This allows for a more flexible and accurate representation of the data, capturing potential curvature or interaction effects that may exist. ANOVA is crucial for evaluating these models and identifying statistically significant terms. Together, these tools enable systematic exploration and optimisation. The general form of the ANOVA model with polynomial terms can be represented in Equation (2):

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \ldots + \beta_n X^n + \varepsilon \tag{2}$$

In this Equation, X represents the independent variable, β_0, β_1, β_2, …, β_n are the coefficients associated with each polynomial term, X^2, X^3, …, X^n represent the different powers of the independent variable, and ε defines the error term.

A prototype vertical UW machine was designed to facilitate the experimental process. The UW parameters encompass a maximum power rating of 1 kW and a frequency of 28 kHz. To achieve the desired welding of fabric–copper connections, ultrasonic waves are applied perpendicularly to the surface, transversely engaging with the fabrics. The UW system comprises essential components, including a power supply, a converter, a booster, and a horn (Figure 3a).

This system transforms the electrical energy, supplied at a frequency of 50–60 Hz by the power supply, into mechanical vibration energy oscillating at 28 kHz through the converter. The resulting frictional heat generated by the vibration is then conveyed to the junction surface via the horn, facilitating the formation of a bond between the fabric and copper wire. The welding and holding time can be precisely adjusted within 1.0 s to 10 s, while the pneumatic pressure applied to the welding area can reach up to 1 MPa.

To make the proper contact point between the fabric and the copper wires, non-woven fabrics were stripped to about 60 mm in length and 20 mm in width. The polyester fabric was provided by Kar-tex (Cairo, Egypt) as a material for facemasks, personal protective clothes, diapers, and medical products in the market. Then, plain annealed stranded copper wires (99.97% pure) supplied by Elsewedy Electric (New Cairo, Egypt) with different diameters (0.2, 0.3, and 0.4 mm) were inserted, centred, and integrated between the two pieces of fabric (between 0.05 and 0.08 mm thick each) using an UW machine; the welded sample is shown in Figure 3b.

After UW, a Scanning Electron Microscope (SEM) model (SU-70), manufactured by Hitachi High-Tech, Japan, was used to investigate the bonding morphology of the fabric-copper joint. A peeling test was conducted to determine the mechanical strength of the joint to analyse the adhesion of electronic interconnectors. Therefore, the self-developed peeling test is ideal for inspecting joints between copper wires and textiles. The sample was clamped and fixed on the grips of the Zwick tensile testing machine model (Z010) manufactured by Zwick\Roell, Germany, with a 10 KN maximum capacity. The force was

applied to the textile with a 10 mm/min travel speed. The configuration of the peeling test is illustrated in Figure 4a.

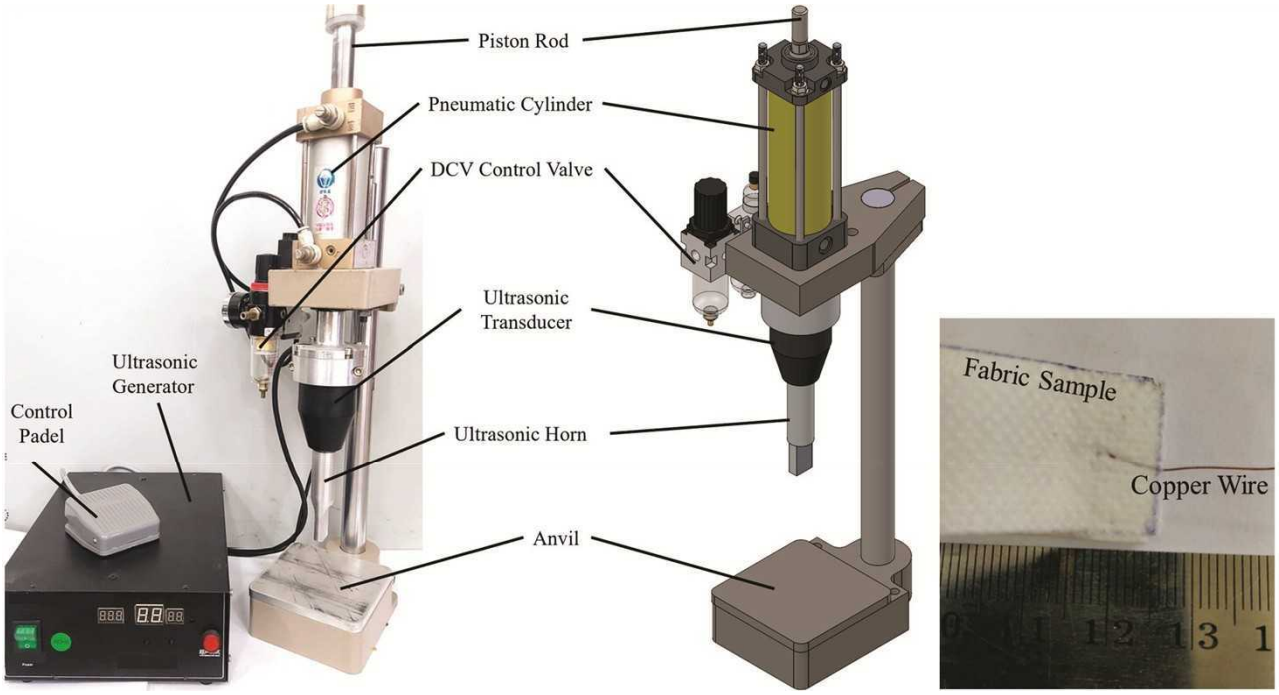


Figure 3. (**a**) Design and fabrication of UW machine; (**b**) produced ultrasonic welded sample for smart textile.

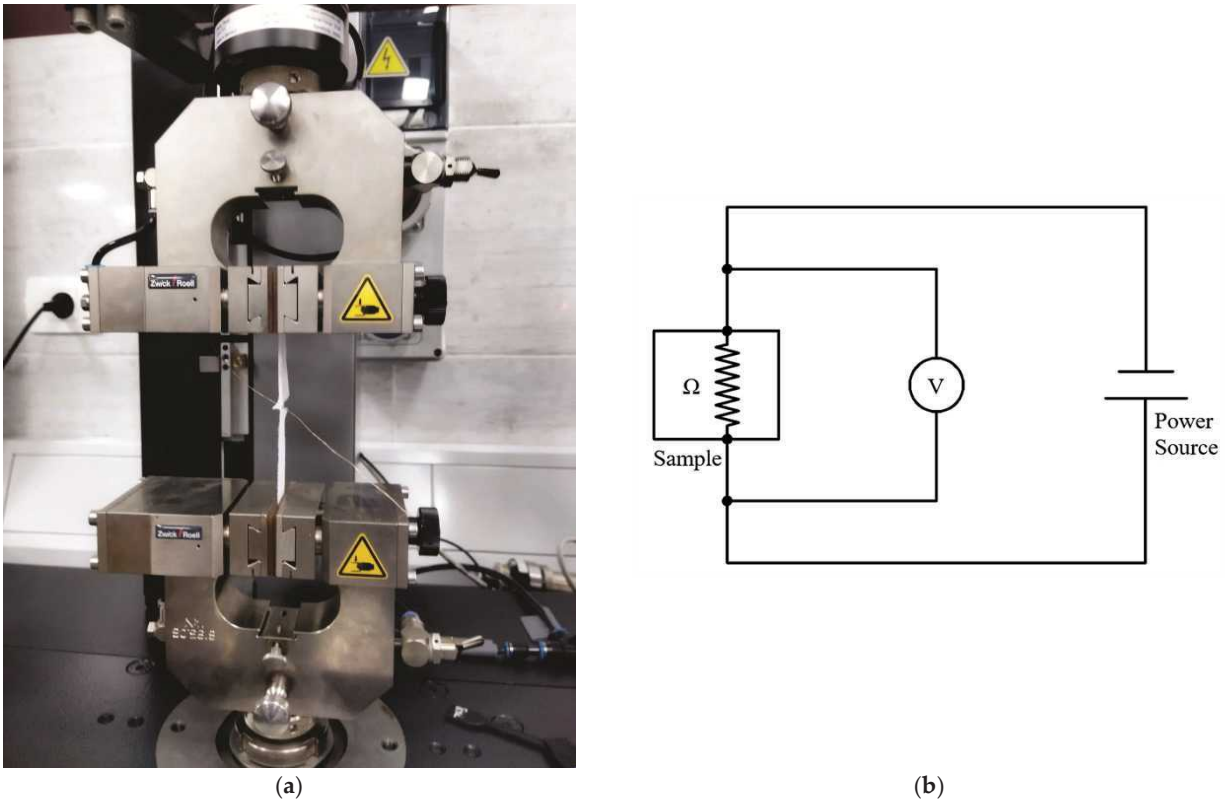


Figure 4. (**a**) Peeling test experimental configuration; (**b**) schematic of electrical resistance measurement setup.

A four-wire measuring instrument is used to measure the contact electrical resistance, as illustrated in Figure 4b. Two of the four contacts provided a current of 1 A. A voltmeter over the two remaining conductors measured the falling voltage at the resistor. Based on Ohm's law, it was possible to calculate the electrical resistance of the contact. The contact electrical resistances were in the range of only a few milliohms. Finally, an infra gun was used to measure the temperature through welding time.

5. Development of Feed Forward Back Propagation Network (FFBPN) for Prediction of Smart Textile Connection Quality

An artificial neural network (ANN) is a computational tool inspired by the complex neural networks of the human brain. It is excellently tuned to discern intricate patterns, make data-driven predictions, and quickly resolve complex engineering challenges. ANNs are adept at accurate outcome prediction and find versatile applications in various engineering domains. ANNs, as computational models, require training to learn and make predictions or classifications. The training process involves adjusting the neural network's internal parameters (weights and biases) based on a dataset to minimise prediction errors.

MATLAB R2020a was used to develop the FFBPN model. The FFBPN model was developed by taking the three parameters (wire diameter, pressure, and time) from the data and considering them as inputs. Heat generation, joint strength, and electrical properties were then taken one at a time as outputs. Three phases of model development were conducted, using 70%, 15%, and 15% of the available data for training, testing, and validation, respectively. If the model meets the performance criteria established during the training stage, it is considered successfully developed and validated; then, the model proceeds to the testing stage. Otherwise, it is recalled for retraining in the first stage. It is deemed validated once the model passes the testing stage and meets the performance criteria. The following four steps comprise the backpropagation method's training algorithm: initialisation of weights, feed-forward, backpropagation of errors, and updating the weights and biases, respectively. The proposed network architecture involves three input neurons for each input parameter, an output layer with one neuron corresponding to one output at a time, and a single hidden layer of neurons.

The Levenberg–Marquardt (LM) backpropagation algorithm will be used if the model does not meet the expected values [31]. This allows the procedure to be repeated until it finds the optimal requirement. When determining the architecture of a backpropagation neural network (BPNN) using a trial-and-error method, several design criteria can guide the process illustrated in Table 4.

Table 4. The configurations for the BPNN used for the regression problems.

Parameter	Value
Network Type	Feed-forward backpropagation
Number of neurons in the hidden layer	4:35, used value was 10
Number of hidden layers	1:5, used value was 1
Training function	Trainlm
Transfer function	sigmoid function Tansig
Learning Rate	0.05
Performance goal	0.0001

After the input layer receives the signals from the other source, the hidden layer converts them into a form that the output signal can utilise. This study's suggested neural network architecture is 3-10-1, as seen in Figure 5.

In the current investigation, experimental data from the DOE emphasises the development of an interactive backpropagation neural network (BPNN) model. This BPNN model was trained on experimental data to provide a predictive tool for process control and optimisation.

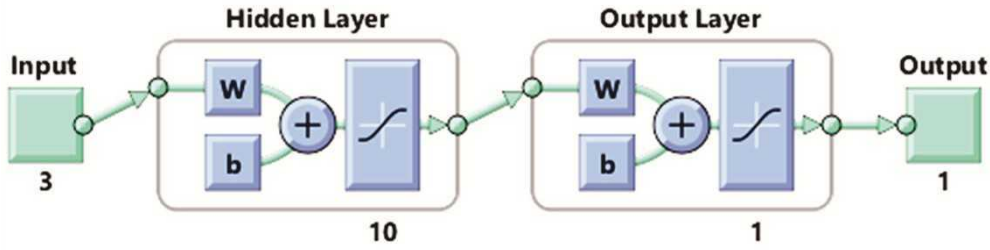


Figure 5. Proposed network architecture. The colours represent different types of layers or components. Turquoise colour represents input/output data and blue colour represents components of each layer and its functions like summation and transfer functions. Arrows depict the flow of in-formation, indicating how data moves from one layer to another. The plus symbol could be used to denote the addition of bias and weights in the equations that govern the calculations within the network.

6. Results and Discussion

6.1. Preliminary Results of the UW Process

The mechanical peeling tests showed a significant relationship between welding parameters and joint strength, as illustrated in Figure 6. For instance, with a 0.2 mm diameter copper wire at 0.4 bar pressure, the peeling force increased from 19.6 N to 24 N as welding time extended from 5 s to 10 s. This trend was consistent across different pressure levels, with the highest peeling forces of 26 N and 28 N observed at 0.8 bar pressure for welding times of 8 s and 10 s. Higher pressure levels enhanced bonding between the non-woven fabric layers by promoting intermolecular contact and improving adhesion strength.

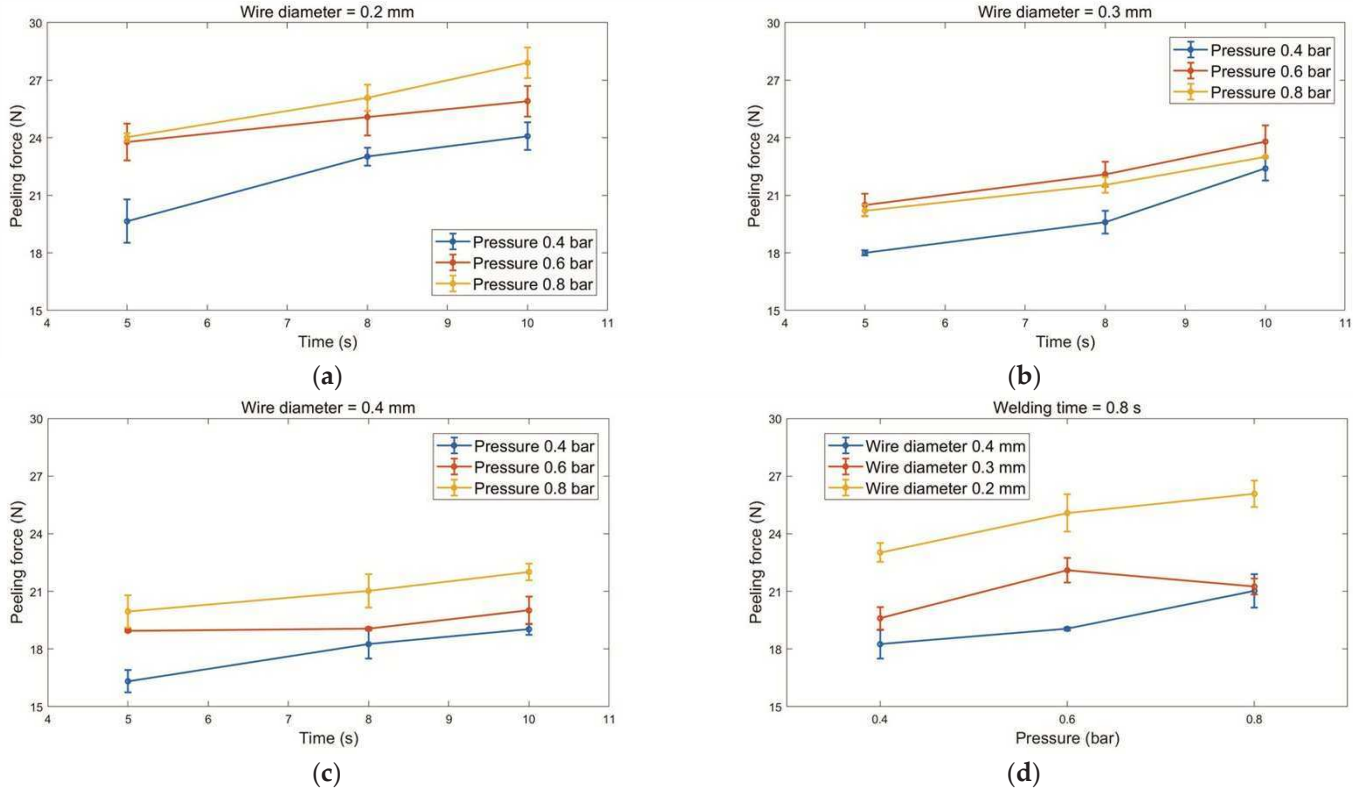


Figure 6. The peeling force for different welding pressures and welding time dependencies for wire diameter of (**a**) 0.2 mm, (**b**) 0.3 mm, (**c**) 0.4 mm, and (**d**) peeling force for different pressures.

The positive correlation between peeling force and welding time underscored that longer welding times facilitate thorough bonding and material interdiffusion, increasing joint strength. Conversely, increasing copper wire diameter led to a decrease in peeling force.

This phenomenon was attributed to the wire's heat dissipation characteristics. A larger wire diameter resulted in more efficient heat dissipation, weakening the bonding between fibres due to reduced heat input and diffusion. These findings emphasise the intricate interplay between welding parameters and joint strength in UW of non-woven fabric. While both pressure and time are vital, pressure appeared to have a more pronounced impact.

The electrical resistance measurements conducted on the wire joints highlighted the significant influence of welding parameters on electrical resistance. The electrical resistance values exhibited variations dependent on both wire diameter and welding conditions. Specifically, for wire diameters of 0.2, 0.3, and 0.4 mm, the recorded electrical resistance values were 0.02, 0.04, and 0.06 Ohm, respectively. All measurements were taken on wires with a fixed length of 60 mm to ensure equitable comparisons.

The electrical resistance data show a slight increase with welding time. For instance, with a 0.2 mm wire diameter at 0.4 bar pressure, electrical resistance increased from 0.02 to 0.021 Ohm, as depicted in Figure 7. However, a more pronounced rise in electrical resistance occurred with heightened pressure. At 0.8 bar pressure and a welding duration of 10 s, electrical resistance reached 0.0256 Ohm. Similar trends materialised with 0.3 and 0.4 mm wire diameters.

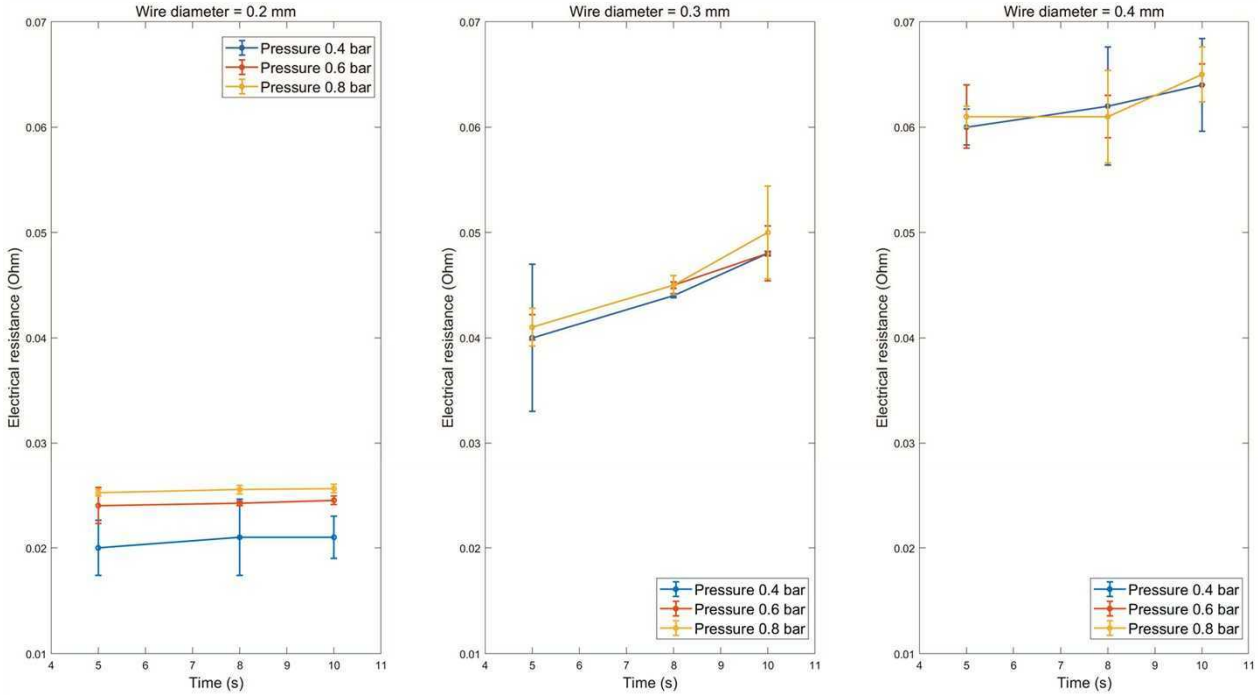


Figure 7. The electrical resistance for different welding pressures and welding time dependencies for wire diameter of (**a**) 0.2 mm, (**b**) 0.3 mm, and (**c**) 0.4 mm.

This relationship between wire diameter and electrical resistance can be attributed to the alterations in cross-sectional area, which led to heightened electrical resistance. While pressure could potentially deform the wire and influence its cross-section, the predominant factor affecting electrical resistance was deemed to be the cross-sectional area. These findings underscore the imperative consideration of welding parameters and wire diameter concerning electrical resistance measurements. Minimising electrical resistance in welded joints is critical to avoid electrical losses and inefficiencies.

The observed correlation between wire diameter and electrical resistance can be attributed to alterations in the weldment area, resulting in elevated electrical resistance. The increase in electrical resistance is primarily attributed to various defects induced by high welding pressure as revealed by SEM observation. As expected, the presence of unbonded regions at the joint interface contributes to an increase in electrical resistance. Consequently,

during charging cycles, the elevated temperature intensifies the electrical resistance, leading to overall performance degradation. Furthermore, the diffusion rate of elements appears to be influenced by welding pressure and time. Previous studies have frequently highlighted that the excess concentration of vacancies generated through severe plastic deformation enhances diffusion around the interface during ultrasonic welding [32,33]. As a result, faster diffusion at the interface with minimal pressure is preferable for achieving a conductive joint. This interpretation aligns with the observed electrical resistance of the 0.4 mm wire as both time and pressure increase, as shown in Figure 6c. These findings underscore the critical importance of considering welding parameters and wire diameter when conducting electrical resistance measurements. Minimising the electrical resistance of welded joints is imperative to mitigate electrical losses and enhance overall efficiency.

The temperature measurements during the UW process for a 0.2 mm copper wire revealed a consistent trend across different time durations and pressures. At a pressure of 0.4 bar, the temperature increased from 68 °C to 79 °C as the welding time increased from 5 s to 10 s, as shown in Figure 8a. This trend was observed consistently across all tested pressures. The highest temperature recorded was 80 °C at a pressure of 0.8 bar and a welding time of 10 s.

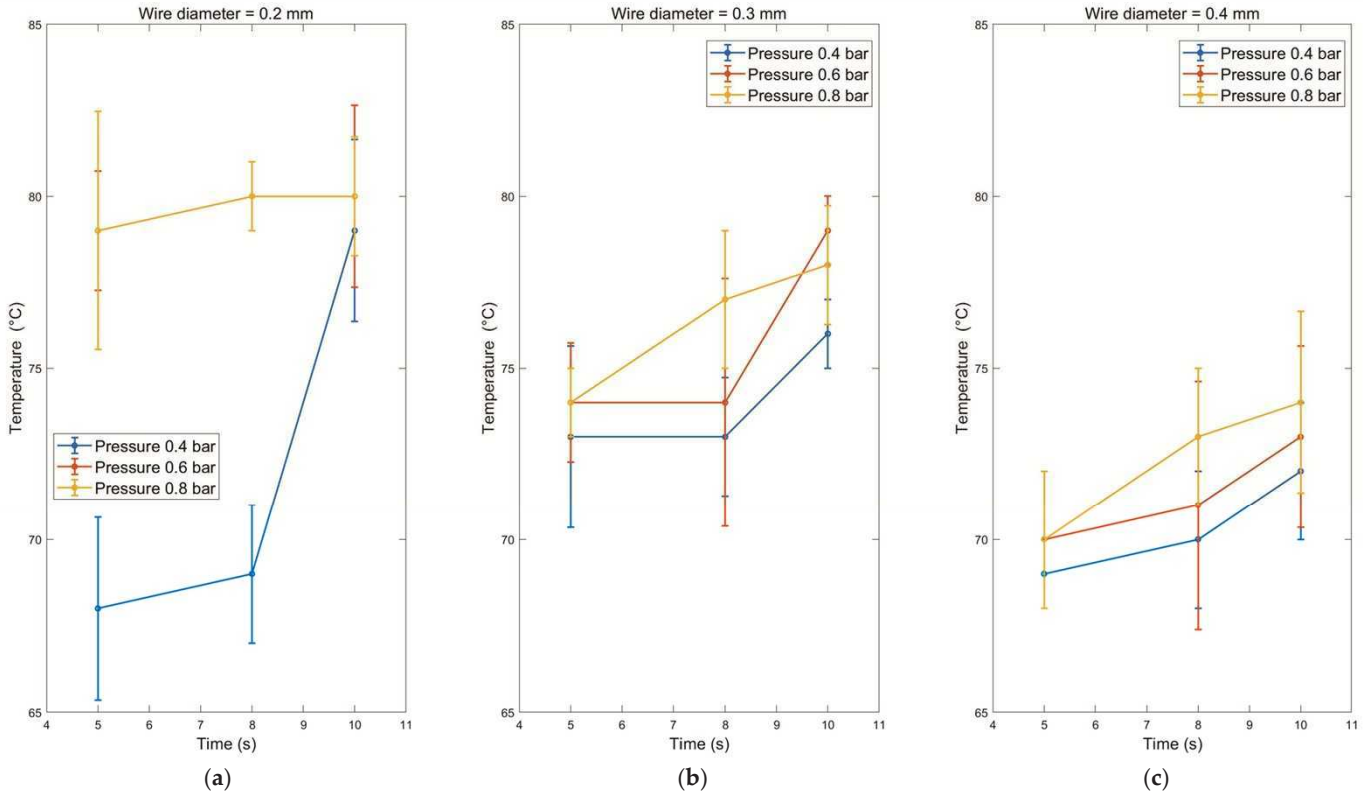


Figure 8. The temperature for different welding pressures and welding time-dependent for wire diameter (**a**) 0.2 mm, (**b**) 0.3 mm, and (**c**) 0.4 mm.

When considering the effect of wire diameter on temperature, it was observed that the temperature slightly decreased as the wire diameter increased. The minimum temperature and the highest pressure were recorded at a welding time of 5 s. This observation suggests that temperature is primarily influenced by the wire diameter, with larger diameters dissipating welding heat more rapidly Figure 8. Furthermore, the effects of welding time and pressure increased the temperature for each independent wire diameter. This indicates that both time and pressure contribute to generating heat during the welding process.

The observed temperature trends can be attributed to the thermal characteristics of the copper wire and the energy input during UW. As the welding time increases, more energy is delivered to the wire, resulting in a higher temperature. Similarly, increased pressure can

increase temperature due to enhanced energy transfer. However, the effect of wire diameter on temperature is linked to its ability to dissipate heat efficiently. These findings emphasise optimising welding parameters to achieve the desired temperature range for effective and reliable bonding. The results suggest that shorter welding times and lower pressures may help reduce excessive temperature generation, mainly when working with smaller wire diameters. Additionally, careful consideration of the wire diameter is necessary to ensure proper heat dissipation and prevent overheating.

Figure 9 shows an SEM of the welded joint under different parameters. The 0.4 mm copper wire with the highest pressure and time appeared to have an efficient bonding interface morphology. With increased welding time and fixing other parameters, the wire was squeezed into the gap regions of the fabrics. The high-frequency vibration caused friction and subsequent heat, which softened the copper in the bonding. It was then squeezed into the arc-like movement path. However, if the time was short, the energy and temperature produced from friction would not be sufficient, and material plastic flow became more difficult even if the pressure continued to increase. In addition, the wire diameter played a vital role in transferring the heat produced by friction, as indicated by Figure 9a,c.

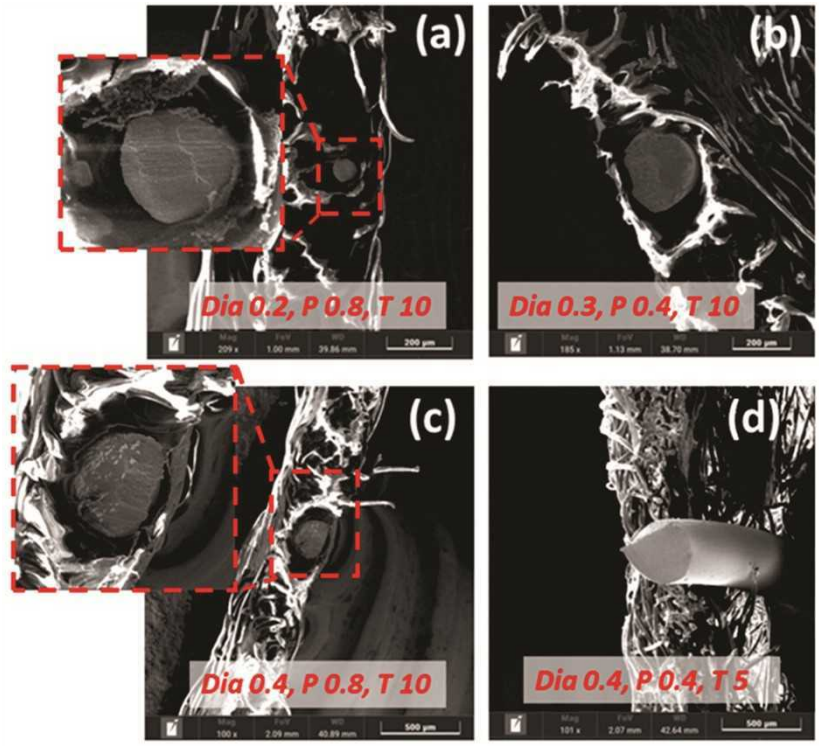


Figure 9. SEM of fabric–copper UW under different parameters. (**a**) Wire diameter 0.2, Pressure 0.8 bar & Time 10 s. (**b**) Wire diameter 0.3, Pressure 0.4 bar & Time 10 s. (**c**) Wire diameter 0.4, Pressure 0.8 bar & Time 10 s. (**d**) Wire diameter 0.4, Pressure 0.4 bar & Time 5 s.

As per UW analysis, after ultrasonic vibration was activated, the horn directly contacted the fabric layers, squeezed and rubbed it, and a relative sliding motion occurred, generating heat and bonding through the acoustic softening effect. Firstly, a virtual bond is formed quickly between the layer's interfaces. With the increase in welding time, the virtual bond would tear firstly and join again under the action of the horn pressure and shear force, and then produce bonding, interlocking to form a robust bond region, which leads to the increase in the joint bonding strength. However, since the force and relative sliding required to create a bond between the fabric layers need to be transferred through the copper wire, there is a certain hysteresis and attenuation, and this makes the time increase not always the influential parameter (Figure 9c,d).

6.2. Statistical Modelling of the UW Process Using the Design of the Experiment (DOE)

Regarding the DOE results, for the current investigation, a 95% level of statistical significance was selected. The determined correlation coefficients R^2 are 80.5%, 95.3%, and 99.37% for heat generation, joint strength, and electrical properties of the welded joints, respectively, showing that the data match the models very well and can be used to derive it. The models that resulted from the analysis can be expressed as:

$$T = 40.7 + 81.1\ D + 68.0\ P + 0.11\ t - 105.6\ D^2 - 26.4\ P^2 + 0.122\ t^2 - 50.0\ D\ P - 2.11\ D\ t - 1.23\ P\ t \quad (3)$$

$$S = 14.57 - 47.8\ D + 33.5\ P + 0.886\ t + 65.9\ D^2 - 17.06\ P^2 + 0.0314\ t^2 - 7.8\ D\ P - \ 1.577\ D\ t - 0.482\ P\ t \quad (4)$$

$$\Omega = -0.02659 + 0.2019\ D + 0.01666\ P + 0.000229\ t - 0.0096\ D\ P - 0.00030\ D\ t\ + 0.000024\ P\ t \quad (5)$$

where D, P, and T represent the independent variables of copper wire diameter, welding pressure, and welding temperature, where t, S, and Ω represent welding time, peeling force, and electrical resistance, respectively.

The impact of UW parameters on heat generation, joint strength, and electrical properties are tested experimentally for 3^3 = 27 combinations involving three variables, and a statistical approach is used to clarify the relative effects of different process parameters on the responses. Thus, a three-factor Analysis of Variance (ANOVA) approach is applied to the collected dataset using the Minitab program to examine the effects of various factors, including wire diameter, pressure, and time on the produced joints. ANOVA analysis results are displayed in Table 5 with a confidence level of 0.95. In this study, the degrees of freedom for single factors were 2, the degrees of freedom for interaction between factors were 4, and the error number was 8.

Table 5. Analysis of Variance (ANOVA) Test.

	Source	Degrees of Freedom (DF)	Adjusted Sums of Squares (Adj SS)	Adjusted Mean Squares (Adj MS)	F-Value	*p*-Value
Heat generation	Diameter (mm)	2	151.185	75.593	14.93	0.002
	Pressure (bar)	2	109.407	54.704	10.80	0.005
	Time (s)	2	38.741	19.370	3.82	0.068
	Diameter (mm) × Pressure (bar)	4	41.481	10.370	2.05	0.180
	Diameter (mm) × Time (s)	4	8.148	2.037	0.40	0.802
	Pressure (bar) × Time (s)	4	7.259	1.815	0.36	0.832
	Error	8	40.519	5.065		
	Total	26	396.741			
Joint strength	Diameter (mm)	2	114.759	57.3797	341.37	0.000
	Pressure (bar)	2	38.578	19.2892	114.76	0.000
	Time (s)	2	39.914	19.9569	118.73	0.000
	Diameter (mm) × Pressure (bar)	4	5.883	1.4706	8.75	0.005
	Diameter (mm) × Time (s)	4	2.953	0.7383	4.39	0.036
	Pressure (bar) × Time (s)	4	2.361	0.5902	3.51	0.062
	Error	8	1.345	0.1681		
	Total	26	205.793			
Electrical properties	Diameter (mm)	2	0.006787	0.003393	13,278.70	0.000
	Pressure (bar)	2	0.000141	0.000070	275.04	0.000
	Time (s)	2	0.000003	0.000001	5.33	0.034
	Diameter (mm) × Pressure (bar)	4	0.000020	0.000005	19.43	0.000
	Diameter (mm) × Time (s)	4	0.000001	0.000000	0.81	0.551
	Pressure (bar) × Time (s)	4	0.000001	0.000000	1.25	0.364
	Error	8	0.000002	0.000000		
	Total	26	0.006954			

For temperature, the wire diameter and pressure significantly influence the experimental results when their contributions to the response are 38.11% and 27.58%, respectively.

Meanwhile, the interaction between factors has no significant influence on temperature. The wire diameter, welding time, and pressure are the significant single factors that affect peeling force. The interaction between wire diameter and pressure follows it. According to the results of ANOVA, wire diameter and pressure are significant single factors followed by interaction between wire diameter and pressure that affect electrical properties.

The main effect plots show how each factor affects the response characteristic. As shown in Figure 10a,b, each temperature and peeling force trend decreased by increasing the diameter and decreasing pressure and time. However, the electrical resistance of weldment increased by increasing each diameter and pressure, as shown in Figure 10c. Meanwhile, the change in levels of time from 5 s to 10 s has a slight, almost negligible impact on electrical resistance.

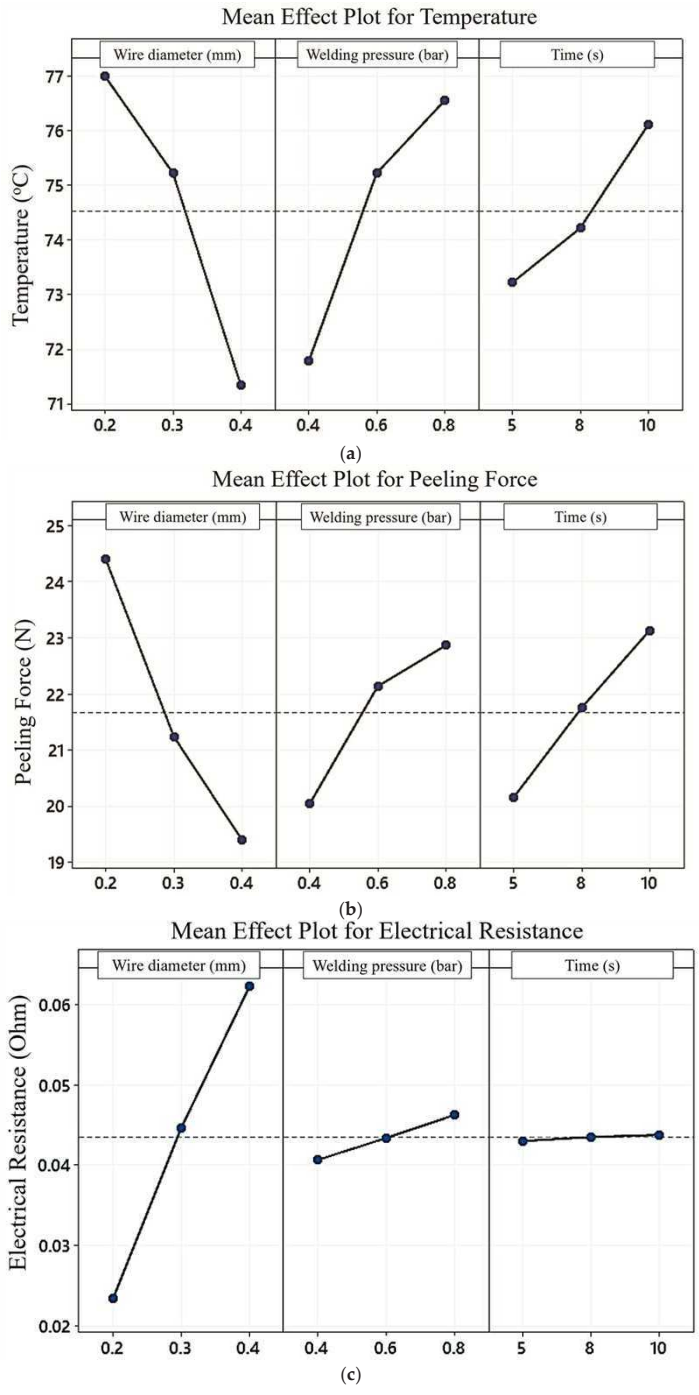


Figure 10. Mean effect plot of (**a**) temperature, (**b**) joint strength, and (**c**) electrical resistance.

Subplots (a), (b), and (c) show a preponderance of the effect of a diameter of 0.4 mm. From (a) and (b), it is concluded that the optimum combination of each process parameter for lower heat generation and joint strength is meeting at wire diameter, 0.4 mm; pressure, 0.4 bar; and time, 10 s. Figure 10c shows that the optimum combination of each process parameter for higher electrical resistance is meeting at wire diameter, 0.4 mm; pressure, 0.8 bar; and time, 5–10 s; this has the same effect.

6.3. Comparison of Experiment, BPNN and FFD Results

A comparison was conducted with experiment results to evaluate the predictive performance of BPNN and FFD. The performances of developed models via FFD and ANN of temperature, mechanical peeling force, and electrical resistance of the welded joints were evaluated using correlation coefficient (R) as shown in Figure 11a–f. Figure 11a,b show the correlation coefficient diagrams among predicted values of each approach and the observed values for measured temperature; the R of the FFD model equals 0.862, while the R of the ANN model equals 0.955. The correlation value for the overall performance of ANN for peeling force is very close to that of the FFD model, which equals 0.967 and 0.96, respectively, as shown in Figure 11c,d. The correlation coefficient R for both models resulting from both methods, FFD and ANN, was approximately the same and very close to 1 when studying electrical resistance, which is equal to 0.997, as shown in Figure 11e,f.

ANN has been found to perform better than FFD or other techniques for non-linear systems. Meanwhile, many repeated calculations are required because of the resilience of the ANN model. This technique also obscures the input factors' contributions and how they interact. The performances of FFD and BPNN models are tested by statistical criteria such as the coefficient of correlation (R), which provides the value of proportionality between the data from the experiment and the estimation, while the accuracy of the estimation results is measured by mean absolute percentage error (MAPE) which can be calculated by following the formula, The results are summarised in Table 6.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_{a,i} - y_{p.i}}{y_{a.i}}\right| \quad (6)$$

where n is the number of experiments, $y_{a,i}$ is the actual value, and $y_{p,i}$ is the predicted value.

Table 6. Comparison of statistical analysis between FFD and ANN models.

	Output					
	Heat Generation		**Joint Strength**		**Electrical Properties**	
	FFD	**ANN**	**FFD**	**ANN**	**FFD**	**ANN**
R	0.862	0.955	0.96	0.967	0.997	0.997
MAPE	0.018153	0.0078	0.0297	0.021998	0.0257	0.0231

Finally, according to the developed FFD and ANN models, the percentage of overall error for ANN and FFD for each heat generation, joint strength, and electrical properties is approximately 3.91%, 6.84%, and 0.49%, respectively, confirming the high accuracy of the predictive model. The FFD model results in an average error percentage of 4.69% and 0.7% for joint strength and electrical properties, respectively, while the rate of overall error of modelling heat generation is 19.52%, which means there may be other variables affecting the process that have not yet been taken into account which are responsible for this amount of variation. The results indicate that both the developed models are highly accurate in predicting the joint strength and electrical properties; however, ANN is more accurate than FFD in predicting heat generation model. The analysis shows that wire diameter mainly contributes 38.11%, 55.76%, and 97.59% to heat generation, joint strength, and electrical properties, respectively. Although the diameter of the wire is considered the primary variable affecting all output responses, its relatively small contribution rate of

38.11% in heat generation compared to the other responses, reflects that the levels used in this experiment were very narrow and reflected in that percentage.

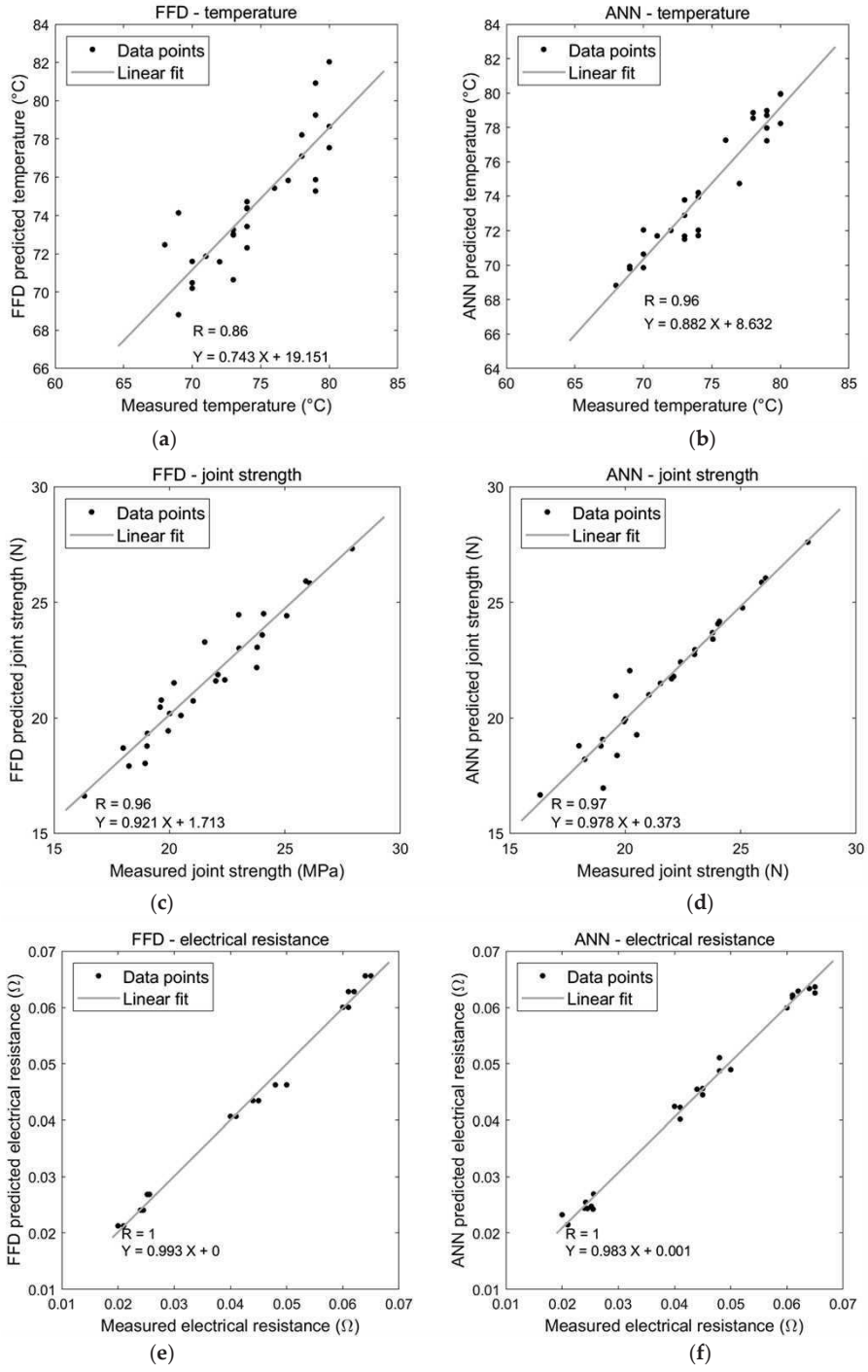


Figure 11. Graphical comparison of correlation plots for FFD and ANN Models for weldment proprieties. (**a**) Correlation plot of FFD for temperature. (**b**) Correlation plot of ANN for temperature. (**c**) Correlation plot of FFD for peeling force. (**d**) Correlation plot of ANN for peeling force. (**e**) Correlation plot of FFD for electrical resistance. (**f**) Correlation plot of ANN for electrical resistance.

7. Conclusions

This work involves the design of a piezoelectric transducer in parallel with the prototype development of a UW machine. This integrated approach aims to explore the impact of welding parameters, such as time, pressure, and wire diameter, on the joint efficiency of

smart textiles. The experimental results derived from this endeavour were subsequently employed in the optimisation process, combining full factorial design (FFD) and artificial neural networks (ANNs) to enhance and control UW processes. The main insights can be summed up as follows:

Both pressure and time are vital; pressure appeared to have a more pronounced impact. The results suggest that an optimal combination of pressure and time can be determined to achieve the desired joint strength, considering specific wire diameter and material characteristics.

Reducing wire diameter and controlling pressure becomes instrumental in achieving lower electrical resistance values and enhancing the overall electrical properties of the joints.

According to experimental research, the minimum heat generation, maximum strength, and maximum electrical resistance of UW of copper wire and non-woven PVC textiles can reach 68 °C, 27.91 N, and 0.065 Ohm. The wire diameter significantly impacts the joint strength, electrical properties, and heat generation followed by welding pressure, but the welding time makes a slightly significant difference. Given the interactions between factors, the interactions between wire diameter and welding pressure slightly affect FFEAct responses. However, the interaction between the other two groups has no significant impact on the experimental results.

The proposed FFD and ANN models accurately predict the UW process's heat generation, joint strength, and electrical properties to join copper wires with non-woven PVC textiles. However, the predictive capability of the FFD model shows a high error percentage of 19.52% only in the case of heat generation. This reflects the necessity of reconsidering the size of the orthogonal array, which plays a critical role in the efficiency of the predictive model.

The ANN with a 3-10-1 architecture is an optimum network with a high correlation coefficient obtained on validation datasets. The model accuracy of ANN was found to be better than FFD, and the former was found to be statistically robust and accurate in predicting heat generation. At the same time, they are almost identical in robustness and accuracy in predicting joint strength and electrical properties.

Author Contributions: Conceptualization, M.B.; Methodology, M.B. and O.O.F.; Software, M.B., M.S.E.-A. and O.O.F.; Validation, M.S.E.-A., O.O.F. and A.A.; Formal analysis, M.B., M.S.E.-A. and O.O.F.; Investigation, M.S.E.-A., O.O.F. and A.A.; Resources, M.B. and A.A.; Data curation, M.B. and M.S.E.-A.; Writing—original draft, M.B., M.S.E.-A., O.O.F. and A.A.; Writing – review & editing, M.B., M.S.E.-A., O.O.F. and A.A.; Visualization, M.S.E.-A. and O.O.F.; Supervision, M.B.; Project administration, M.B. and A.A.; Funding acquisition, M.B. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the support the Egyptian Academy of Scientific Research and Technology provided for funding the experimental work conducted in this study.

Data Availability Statement: The data supporting this study's findings are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fernández-Caramés, T.M.; Fraga-Lamas, P. Towards the Internet of Smart Clothing: A Review on IoT Wearables and Garments for Creating Intelligent Connected E-Textiles. *Electronics* **2018**, *7*, 405. [CrossRef]
2. Miah, M.R.; Yang, M.; Hossain, M.M.; Khandaker, S.; Awual, M.R. Textile-based flexible and printable sensors for next generation uses and their contemporary challenges: A critical review. *Sens. Actuators A Phys.* **2022**, *344*, 113696. [CrossRef]
3. Cherenack, K.; Van Pieterson, L. Smart textiles: Challenges and opportunities. *J. Appl. Phys.* **2012**, *112*, 091301. [CrossRef]
4. Sajovic, I.; Kert, M.; Boh Podgornik, B. Smart Textiles: A Review and Bibliometric Mapping. *Appl. Sci.* **2023**, *13*, 10489. [CrossRef]
5. Ojuroye, O.; Torah, R.; Beeby, S.; Wilde, A. Smart textiles for smart home control and enriching future wireless sensor network data. In *Sensors for Everyday Life: Healthcare Settings*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 159–183.
6. Tseghai, G.B.; Malengier, B.; Fante, K.A.; Nigusse, A.B.; Van Langenhove, L. Integration of conductive materials with textile structures, an overview. *Sensors* **2020**, *20*, 6910. [CrossRef]
7. Stoppa, M.; Chiolerio, A. Wearable electronics and smart textiles: A critical review. *Sensors* **2014**, *14*, 11957–11992. [CrossRef]

8. Kim, I.; Shahariar, H.; Ingram, W.F.; Zhou, Y.; Jur, J.S. Inkjet process for conductive patterning on textiles: Maintaining inherent stretchability and breathability in knit structures. *Adv. Funct. Mater.* **2019**, *29*, 1807573. [CrossRef]
9. Kazani, I.; Hertleer, C.; De Mey, G.; Schwarz, A.; Guxho, G.; Van Langenhove, L. Electrical conductive textiles obtained by screen printing. *Fibres Text. East. Eur.* **2012**, *20*, 57–63.
10. Mondal, K. Recent advances in soft E-textiles. *Inventions* **2018**, *3*, 23. [CrossRef]
11. Ruckdashel, R.R.; Khadse, N.; Park, J.H. Smart E-textiles: Overview of components and outlook. *Sensors* **2022**, *22*, 6055. [CrossRef]
12. Du, K.; Lin, R.; Yin, L.; Ho, J.S.; Wang, J.; Lim, C.T. Electronic textiles for energy, sensing, and communication. *IScience* **2022**, *25*, 104174. [CrossRef]
13. Islam, R.; Khair, N.; Ahmed, D.M.; Shahariar, H. Fabrication of low cost and scalable carbon-based conductive ink for E-textile applications. *Mater. Today Commun.* **2019**, *19*, 32–38. [CrossRef]
14. Micus, S.; Rostami, S.G.; Haupt, M.; Gresser, G.T.; Meghrazi, M.A.; Eskandarian, L. Integrating electronics to textiles by ultrasonic welding for cable-driven applications for smart textiles. *Materials* **2021**, *14*, 5735. [CrossRef] [PubMed]
15. Sanga, B.; Wattal, R.; Nagesh, D.S. Mechanism of joint formation and characteristics of interface in ultrasonic welding: Literature review. *Period. Eng. Nat. Sci.* **2018**, *6*, 107–119. [CrossRef]
16. Ismar, E.; Kurşun Bahadir, S.; Kalaoglu, F.; Koncar, V. Futuristic clothes: Electronic textiles and wearable technologies. *Glob. Chall.* **2020**, *4*, 1900092. [CrossRef] [PubMed]
17. Kuo, C.-C.; Tsai, Q.-Z.; Li, D.-Y.; Lin, Y.-X.; Chen, W.-X. Optimization of Ultrasonic Welding Process Parameters to Enhance Weld Strength of 3C Power Cases Using a Design of Experiments Approach. *Polymers* **2022**, *14*, 2388. [CrossRef] [PubMed]
18. Le, T.Q.; Nguyen, T.H.; Nguyen, L.H. Optimisation of technological parameters in ultrasonic welding of the polypropylene fabric using Taguchi and FCCCD methods. *Phys. Eng.* **2023**, *3*, 97–111.
19. Shi, W.; Little, T. Mechanisms of ultrasonic joining of textile materials. *Int. J. Cloth. Sci. Technol.* **2000**, *12*, 331–350. [CrossRef]
20. Dils, C.; Hohner, S.; Schneider-Ramelow, M. Use of Rotary Ultrasonic Plastic Welding as a Continuous Interconnection Technology for Large-Area e-Textiles. *Textiles* **2023**, *3*, 66–87. [CrossRef]
21. Pu, J.; Ma, K.; Luo, Y.; Tang, S.; Liu, T.; Liu, J.; Leung, M.; Yang, J.; Hui, R.; Xiong, Y.; et al. Textile electronics for wearable applications. *Int. J. Extrem. Manuf.* **2023**, *5*, 042007. [CrossRef]
22. Radouchova, M.; Janda, M.; Kalas, D.; Blecha, T. Interconnection of Highly Flexible Carbon Threads Suitable for Wearable Strain Monitoring using Ultrasonic Plastic Welding. In Proceedings of the 2023 46th International Spring Seminar on Electronics Technology (ISSE), Timisoara, Romania, 10–14 May 2023; pp. 1–5.
23. Hirman, M.; Navratil, J.; Steiner, F.; Reboun, J.; Soukup, R.; Hamacek, A. Study of low-temperature interconnection techniques for instant assembly of electronics on stretchable e-textile ribbons. *Text. Res. J.* **2022**, *92*, 4269–4287. [CrossRef]
24. Yin, F.; Mao, H.; Hua, L.; Guo, W.; Shu, M. Back propagation neural network modeling for warpage prediction and optimisation of plastic products during injection molding. *Mater. Des.* **2011**, *32*, 1844–1850. [CrossRef]
25. Li, J.; Cheng, J.-H.; Shi, J.-Y.; Huang, F. Brief introduction of back propagation (BP) neural network algorithm and its improvement. In Proceedings of the Advances in Computer Science and Information Engineering; Elsevier: Amsterdam, The Netherlands, 2012; Volume 2, pp. 553–558.
26. Kumar, P.; Misra, J.P. Modelling of machining characteristics during green machining of biomaterials. *J. Inst. Eng. Ser. C* **2020**, *101*, 847–859. [CrossRef]
27. Mongan, P.; Hinchy, E.; O'Dowd, N.; McCarthy, C.T. Optimisation of ultrasonically welded joints through machine learning. *Procedia CIRP* **2020**, *93*, 527–531. [CrossRef]
28. Baraya, M.Y.; Hossam, M. AMT Design of an electromechanical system for measuring and monitoring micro-ultrasonic amplitude of Langevin transducer. *Int. J. Adv. Manuf. Technol.* **2020**, *107*, 2953–2965. [CrossRef]
29. Dassault Systèmes Simulia Corp. *Abaqus 6.12 Analysis User's Manual Volume III: Materials*; Dassault Systèmes Simulia Corp: Paris, France, 2012.
30. Dassault Systèmes Simulia Corp. *Abaqus 6.12 Example Problems Manual Volume II: Other Applications and Analyses*; Dassault Systèmes Simulia Corp: Paris, France, 2012.
31. Moayedi, H.; Aghel, B.; Vaferi, B.; Foong, L.K.; Bui, D.T. The feasibility of Levenberg–Marquardt algorithm combined with imperialist competitive computational method predicting drag reduction in crude oil pipelines. *J. Pet. Sci. Eng.* **2020**, *185*, 106634. [CrossRef]
32. Pradeep Kumar, J. Effect of temperature distribution in ultrasonically welded joints of copper wire and sheet used for electrical contacts. *Materials* **2018**, *11*, 1010. [CrossRef]
33. Feng, M.; Wang, Z.; Meng, D.; Liu, C.; Hu, J.; Wang, P. A review of quality monitoring for ultrasonic metal welding. *Mater. Sci. Technol.* **2024**, *40*, 02670836231215651. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

BiVi-GAN: Bivariate Vibration GAN

HoeJun Jeong [1], SeongYeon Jeung [1], HyunJun Lee [2] and JangWoo Kwon [3,*]

[1] Department of Electric Computer Engineering, Inha University, Incheon 22212, Republic of Korea; lilmae@inha.edu (H.J.); ran22314@inha.edu (S.J.)
[2] Technology Research Center, RMS Technology Co., Ltd., Cheonan 31217, Republic of Korea; hjlee@rmstech.co.kr
[3] Department of Computer Engineering, Inha University, Incheon 22212, Republic of Korea
* Correspondence: jwkwon@inha.ac.kr

Abstract: In the domain of prognosis and health management (PHM) for rotating machinery, the criticality of ensuring equipment reliability cannot be overstated. With developments in artificial intelligence (AI) and deep learning, there have been numerous attempts to use those methodologies in PHM. However, there are challenges to applying them in practice because they require huge amounts of data. This study explores a novel approach to augment vibration data—a primary component in traditional PHM methodologies—using a specialized generative model. Recognizing the limitations of deep learning models, which often fail to capture the intrinsic physical characteristics vital for vibration analysis, we introduce the bivariate vibration generative adversarial networks (BiVi-GAN) model. BiVi-GAN incorporates elements of a physics-informed neural network (PINN), emphasizing the specific vibration characteristics of rotating machinery. We integrate two types of physical information into our model: order analysis and cross-wavelet transform, which are crucial for dissecting the vibration characteristics of such machinery. Experimental findings show the effectiveness of our proposed model. With the incorporation of physics information (PI) input and PI loss, the BiVi-GAN showed a 70% performance improvement in terms of JS divergence compared with the baseline biwavelet-GAN model. This study maintains the potential and efficacy of complementary domain-specific insights with data-driven AI models for more robust and accurate outcomes in PHM.

Keywords: vibration; rotary machine; deep learning; PINN; GAN

1. Introduction

In industrial areas, numerous dynamic systems operate in an integrated manner, among which rotating equipment constitutes one of the most utilized types of machinery. Rotating equipment is predominantly employed in various mechanical conversion devices such as automobile engines, pumps, wind turbines, power generation systems, and gas turbines [1,2]. The failure of such equipment can potentially lead to malfunctions in the associated machinery and a loss of functionality in the overall system. Extensive research has been conducted on the diagnosis and prediction of the reliability and safety of rotating equipment [3–6]. Those approaches have evolved into the field of prognosis and health management (PHM). Particularly, research on artificial intelligence (AI)-based PHM has been actively conducted, owing to advancements in computer technology [7–10]. In the realm of AI-based PHM research, in contrast with traditional PHM approaches, which are structured around physical laws, various forms of data are acquired from the equipment. This data serves as the foundation for analyzing the equipment's health and remaining useful life (RUL). With advancements in deep learning models, analytical methodologies have been proposed. Several studies are based on models that enable generalized analysis without the requirement for expert knowledge [11,12]. Such research endeavors utilize diverse data sources, including thermal images, high-speed camera images, vibration

data, and noise data, to train AI models. Among these, vibration data have traditionally been a critical element in the analysis of rotating equipment within conventional PHM methodologies. Owing to its rich information related to the operation and condition of the equipment, vibration data are also extensively employed in deep learning studies where features contained in the data are of significant importance. However, a salient challenge with these data-driven methodologies is the conflict between the massive volume of data generally required to train AI models and the limited amount of vibration data that can realistically be collected in industrial settings.

As the prevalence of data-hungry deep learning models continues to rise, data augmentation techniques have increasingly become a significant issue in the field of AI. Numerous studies have demonstrated that training AI models on augmented data can help them outperform those trained on the limited amount of original data that can be collected in practice [13]. Vibration data have temporal correlations and can thus be classified as time series data. Studies on methods for augmenting time series data have been vigorously conducted, reflecting the growing recognition of its importance [14]. In basic approaches, various techniques such as cropping, flipping, and jittering are employed in the time series, frequency, and time–frequency domains to artificially augment the data [15]. These methodologies do not focus on understanding the inherent characteristics of the data; rather, they augment the data based on specific rules and probabilities. Recently, with the development of data analytics and learning models, techniques for analyzing the complex rules and information inherent in data have emerged [16,17]. These include decomposition methods, statistical generative models, and learning models. Particularly, deep learning-based generative models such as generative adversarial networks (GANs), variational autoencoders (VAEs), and diffusion models have high applicability across various domains. This is because they can extract the nonlinear features inherent in the data in an end-to-end manner without the requirement for specialized knowledge.

However, the black-box nature of deep learning models poses a challenge, as these generative models may not adequately capture the key physical characteristics relevant to vibration analysis [18,19]. Consequently, there are frequent instances where the generated data fail to reflect the vibration characteristics required for analyzing rotating equipment. To address these challenges raised from the perspective of field application, physics-informed neural networks (PINNs) have emerged as a solution aimed at overcoming the limitations of deep learning models, which are highly dependent on the data themselves [20]. PINNs employ a structure that guides the neural network to learn physical information and characteristics that are difficult for AI to extract by itself. This enables the leveraging of both the advantages of deep learning, which has an excellent ability to extract various features based on data, and the robust, dimensionless characteristics that can be represented by physical equations. In this study, we aim to propose a vibration data augmentation model that incorporates the structure of a PINN. Specifically, the model is designed with a structure specialized for the vibration characteristics of rotating equipment, thus enabling efficient consideration of the relevant physical features. The model proposed in this study has the objective of mapping two types of physical information onto the existing structure of a bidirectional GAN (Bidirectional GAN) [21]. One of the types of physical information aims to incorporate order analysis, a key analytical focus in rotating machinery, which analyzes harmonic frequency components based on the equipment's rotational speed. The other type of physical information aims to utilize cross-wavelet transform, a methodology that analyzes the time–frequency axis correlation between two sets of vibration data.

2. Related Work

2.1. Time Series Data Generation Using Generative Models

Generally, research on methods for generating data to train AI models, particularly deep learning models, has been consistently conducted. Recently, there has been a proliferation of studies focused on utilizing deep learning generative models for data creation [22,23]. The most commonly used are VAEs and GANs. Both VAEs and GANs are similar in terms

of learning the data distribution based on maximum likelihood estimation. However, VAEs belong to the explicit density model category, approximating the prior distribution of the model to estimate the data distribution. In contrast, GANs are part of the implicit density model category, which, instead of explicitly defining the model, iteratively performs sampling to converge to a specific probability distribution. VAEs have the advantage of being able to generate data similar to a given x by directly calculating the probability of x in the training set, as shown in Figure 1.

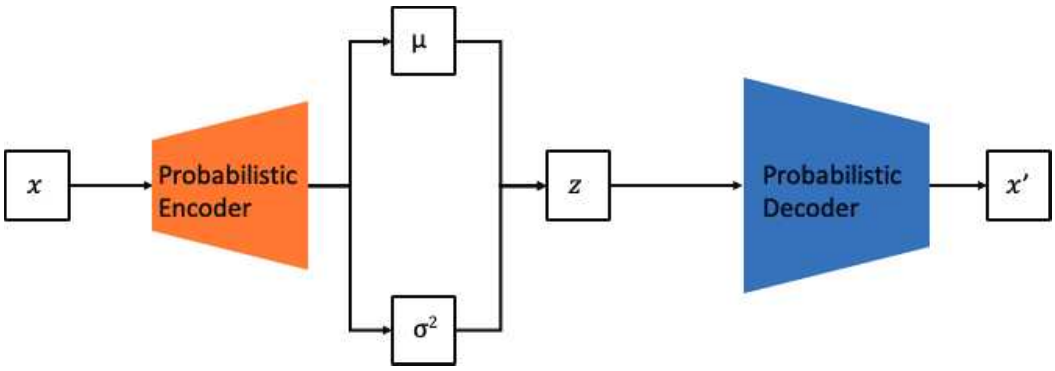


Figure 1. Architecture of VAE.

However, VAEs have the drawback of producing "blurred" generated outputs owing to the injection of noise and the imperfection of the variational bounds. Consequently, much of the recent generative research has been based on GAN models. GANs are composed of two neural networks, commonly referred to as the generator and discriminator, as shown in Figure 2.

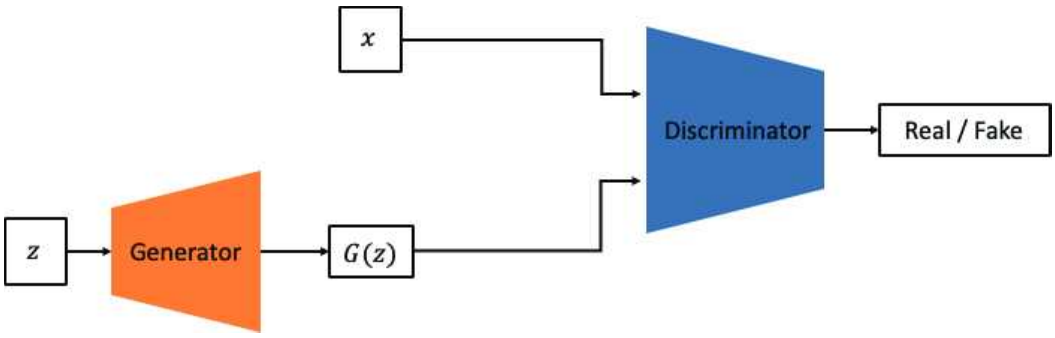


Figure 2. Architecture of GAN.

The generator aims to produce data that are difficult to distinguish from the real data, while the discriminator's role is to differentiate between genuine and generated data [24,25]. Although GANs have a more challenging training process compared with VAEs, they are capable of generating sharper outputs and can cover a broader range of distributions beyond the specific distributions exhibited by the given input data. These advantages have led to their widespread use in recent research. The model proposed in this study also aims to generate vibration data gathered from rotating equipment, leveraging the adversarial training approach inherent to GANs. In this study, among various GAN models, we utilize the Bidirectional GAN with a conditional GAN structure as the base model. Conditional GANs aim to overcome the limitation of the original GAN, which cannot learn different distributions for multiple labels when only sampling z from the latent space as input. To address this, conditional GANs incorporate class information along with z as additional input, enabling the GAN model to learn different distributions for each class. Conditional GANs have evolved in various directions, from the simple act of adding a condition y to using the image itself as a condition, as in the case of pix2pix. In this study, the condition is set to include the presence or absence of defects, as well as the type of defects in the rotating

equipment. This allows the model to learn the distribution of data that occur under various conditions.

The Bidirectional GAN not only learns to generate x from z but also incorporates an additional encoder structure that learns the inverse mapping operation, projecting x back into the latent space as shown in Figure 3 [21].

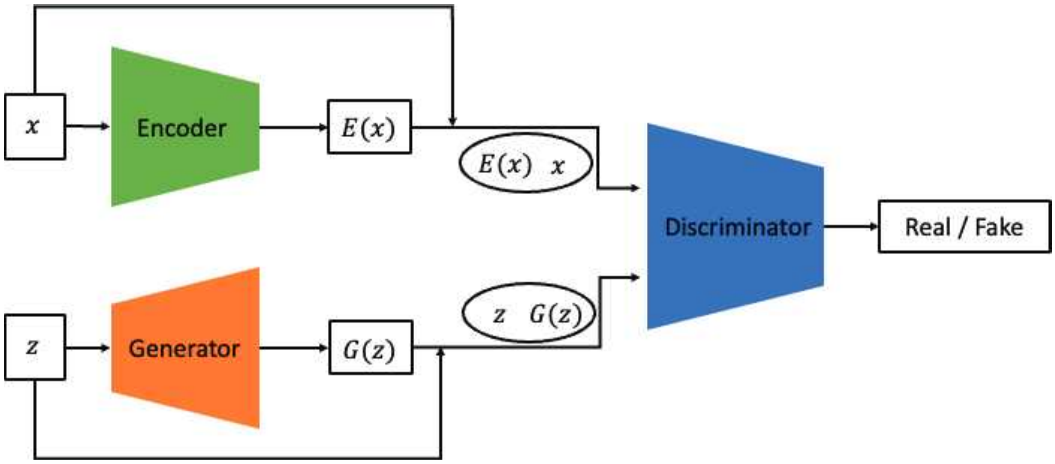


Figure 3. Architecture of Bidirectional GAN.

The discriminator is trained using x and z as inputs, undergoing a process to align p_G and p_E. The authors claim that this structure can overcome the saddle point problem, which is a common issue in GANs. Subsequent research has indeed used the BiGAN structure to synthesize more refined and higher-quality images. In this study, the generator is required to simultaneously produce both the vibration signal itself and the continuous wavelet transform (CWT) image. Therefore, a more stable training method is required. We have configured the model to resolve the saddle point problem and reach a global optimum by employing the inverse mapping technique proposed in BiGAN.

2.2. Frequency Feature Extraction Using Wavelet

The Fourier transform operates under the assumption that any given signal can be represented as a sum of sine waves with varying frequencies and amplitudes. It is a method used to decompose a given signal into its constituent frequency components and intensities, thereby enabling data analysis in the frequency domain. Inherently designed for frequency component analysis, the Fourier transform inevitably suffers from the loss of information along the time axis [26]. To overcome this limitation, the short-time Fourier transform (STFT) was proposed, as shown in Figure 4 [27]. STFT involves dividing a signal that changes over time into short time windows and applying the Fourier transform to each window. This allows for the extraction of frequency components present within each time segment, which are then accumulated. The results of STFT can be represented as an image with a time and frequency axis, making it suitable for signal representation. Consequently, numerous machine and deep learning models have been applied to analyze given vibration signals through STFT analysis.

The wavelet transform overcome the limitation of STFT, which suffers from a trade-off between time resolution and frequency resolution owing to its reliance on sine wave-based frequency decomposition [28]. A wavelet transform involves defining a waveform function with a limited temporal extent and using it to analyze signals as shown in Figure 5.

This allows for an improved time–frequency resolution trade-off. Consequently, the wavelet transform offers the advantage of enhancing time resolution for high-frequency components and frequency resolution for low-frequency components. This enables a more balanced resolution trade-off based on the nature of the signal's frequency components. However, the process of shifting the waveform function and calculating the similarity for the given input signal in a wavelet transform, while scaling the waveform function,

results in a significantly higher computational load compared with STFT. Consequently, the wavelet transform is limited in real-world industrial applications owing to issues related to real-time processing and memory constraints.

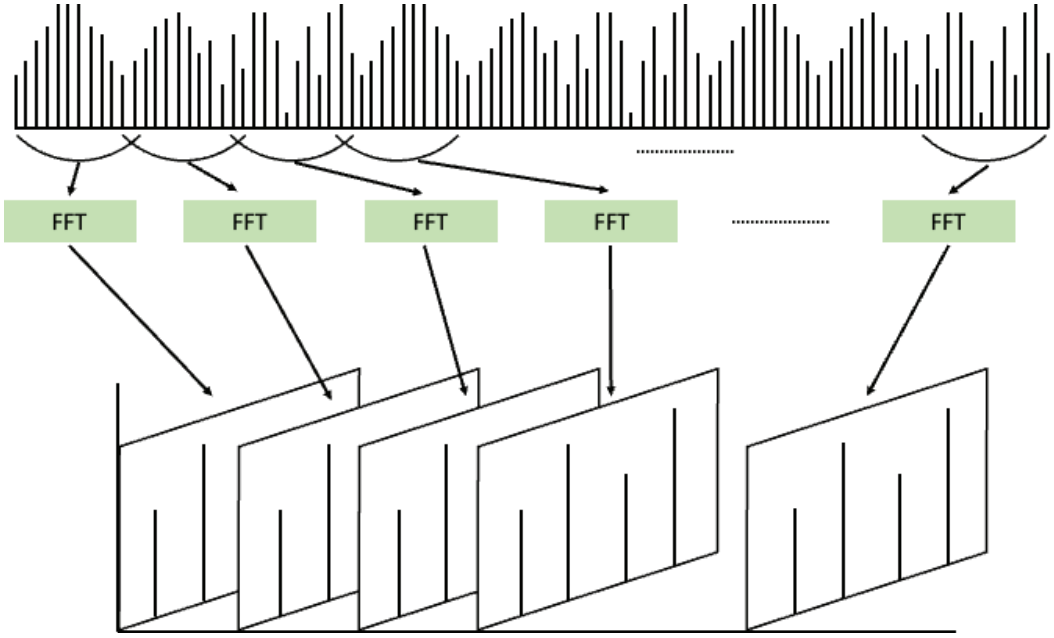


Figure 4. Short-Time Fourier Transform (STFT).

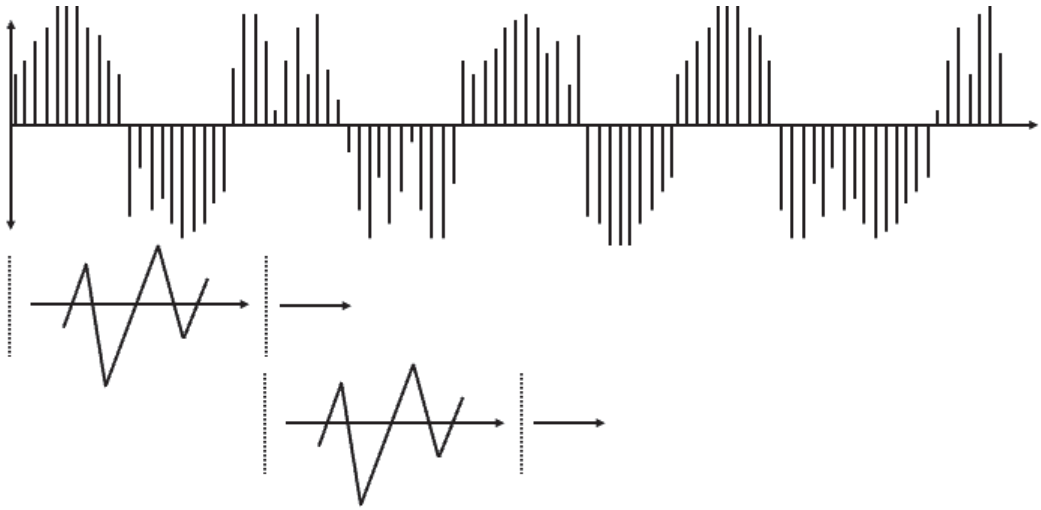


Figure 5. Wavelet Transform.

Certainly, research utilizing similar frequency analysis methods has been conducted using multivariate time series data as well [29,30]. However, in most studies, each signal is often treated as an independent entity, applying separate transformations to each signal and subsequently concatenating them on a channel-by-channel basis [31,32]. Alternatively, these transformations are applied to multiple received inputs and utilized accordingly. In this study, in contrast with analyses that conclude with a frequency analysis of individual signals, the aim is to apply CWT to two signals after performing frequency analysis for both. This is to subsequently analyze the correlation between the two signals within the frequency domain. CWT involves applying the same mother wavelet signal and analysis technique to both signals. Following the transformation, the two values are multiplied to generate the cross-wavelet spectrum. This spectrum showcases the correlation and phase difference between the two signals in the time–frequency domain. The input signals

utilized in this study are vibration data collected from the same vibration source with a phase difference of 90°. Leveraging the advantages of CWT, which effectively represents the frequency-based correlation and its strength, the study aims to analyze the bivariate vibration data and interpret the interactions between them in the frequency domain.

2.3. Learning Physical Characteristics Using PINN

Certainly, AI models, particularly methodologies such as machine learning (ML) and deep learning (DL), profoundly rely on data-driven approaches. Consequently, the dependence on the additional information inherent in the actual data diminishes. However, in domains where the underlying physical system is well-understood, leveraging predefined rules as prior knowledge before training on gathered data can lead to achieving strong performance with a limited dataset. This approach can also prevent issues arising from the application of black-box models to hidden data, which is a concern owing to the nature of such models. Recently, research aiming to combine the advantages of data-driven and physics-based approaches has gained momentum. This approach is referred to as physics-informed machine learning (PIML) or PINN. PINN is structured into two main branches, as shown in Figure 6.

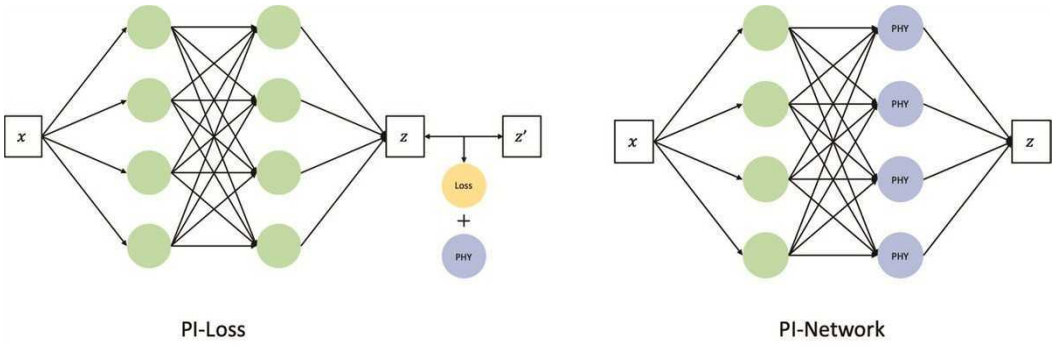


Figure 6. (**Left**): Physics-informed loss model; (**Right**): Physics-informed architecture.

One branch involves enforcing known physical information onto the AI model [33–35]. This can be achieved through methodologies such as incorporating the upper and lower bounds of physics information into the neural network as regularization or utilizing loss functions that leverage the physics information. The other branch involves the neural network itself mapping physical information. This is achieved through a "physics-informed (PI) architecture", where specific nodes or layers are designed to receive or learn the physical information [36,37]. In this study, the proposed approach involves incorporating a PI input that considers frequency characteristics into the Bidirectional GAN-based generative model. Additionally, the model generates cross-wavelet transform images based on learned features during data generation. The differences between these images are used as a loss function, referred to as the PI loss, to leverage PI.

2.4. Bearing Failure Detection via Deep Learning Model

Bearings are one of the critical components of rotating machinery, and the failure of bearings is so significant that it can lead to the shutdown of the entirety of the equipment. Accordingly, numerous studies have been conducted with approaches aiming to perform bearing fault detection. Early research relied on features extracted by experts, utilizing methods like Artificial Neural Networks (ANN), k-Nearest Neighbor, and Support Vector Machines (SVM) for classification and anomaly detection. The features were mainly collected based on frequency domain, including Short-Time Fourier Transform (STFT), Fast Fourier Transform (FFT), Hilbert Transform (HT), and Wavelet Transform (WT) [38,39]. However, these methodologies have limitations due to the necessity of preceding feature engineering, due to heavily relying on expert knowledge in selecting feature extraction techniques, and in determining the structure of models based on these to perform tasks.

With the development of deep learning, many studies have been conducted to utilize the capability of deep learning to extract features directly from data. In the field of bearing fault detection, research utilizing deep learning has been actively conducted [40–42]. Tand et al. studied a method of visualizing data for the analysis of low-speed rotating equipment [43]. Particularly, there have been many studies utilizing inputs from multiple sensor data simultaneously, including methods diagnosing bearing status using multiple sensors and a small number of filters [44], and methods mixing each sensor channel for robust learning [45].

In this study, we utilize data from two sensors attached to the bearing. We model the 90-degree phase difference between the two sensors as a feature. Additionally, we propose BiVi-GAN that synthesizes data using the frequency and temporal characteristics obtained, thereby preserving correlations that cannot be considered in single-sensor data only.

3. Methodology

In this study, we propose a new GAN model, the BiVi-GAN, which employs a Bidirectional GAN architecture combined with PI loss and PI input. The BiVi-GAN, as shown in Figure 7, comprises three sub-networks: the generator, encoder, and discriminator.

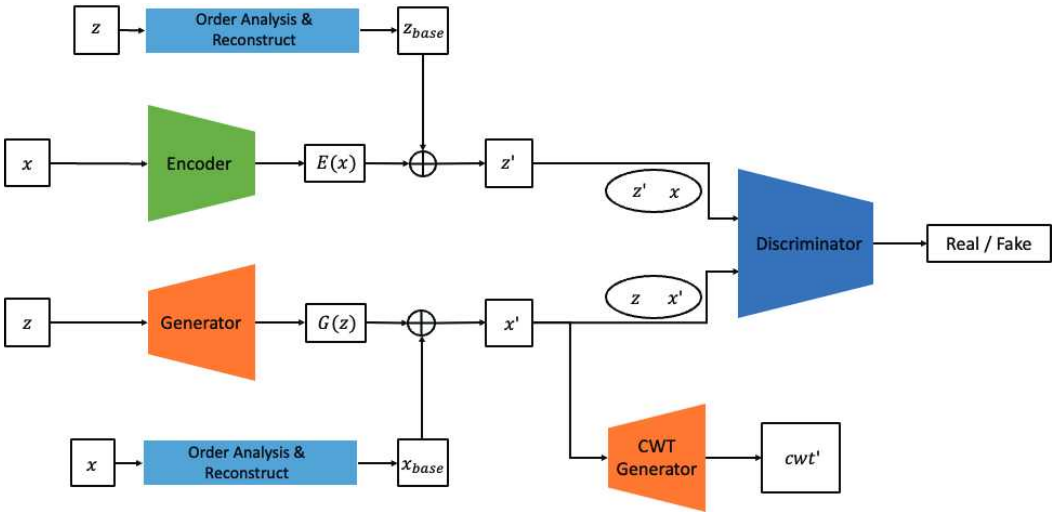


Figure 7. Architecture of BiVi-GAN.

The generator accepts random noise z following a Gaussian distribution and a 1x2x feature x_{phy} extracted from real vibration data x, producing synthetic vibration data x' and synthetic cross-wavelet transform images cwt'. The encoder considers the real data x and a 1x2x feature z_{phy} extracted from random noise z as inputs and generates an estimated value z' for z. The discriminator performs the task of classifying between real and fake by accepting the following as inputs: random noise z, the set of data generated by the generator from z as $\{z, x\}'$, real data x, and random noise z' generated by the encoder based on x, represented as $\{z', x\}$.

The proposed model utilizes two major types of physical information related to vibration. One is the order frequency extracted from the vibration signal x. Order frequency is a feature commonly used in order analysis, a diagnostic tool extensively employed in the field of mechanical engineering that signifies frequency values that are multiples of the equipment's rotational speed. Numerous studies in the field of mechanical engineering have demonstrated that order analysis not only distinguishes between the normal and abnormal states of rotating equipment but also can diagnose the cause and severity of anomalies. Hence, it's evident that the order frequency contains substantial physical information regarding defects in vibration data. In this study, we extract the 1st and 2nd order frequency and directly input them into the generator as the physical information for the given vibration data x. This can be compared to the method of incorporating physical prior

knowledge, a technique frequently employed in PINN series of models. The generator is tasked with learning about frequency characteristics of x that are not described by this prior knowledge, as well as information regarding noises originating both externally and internally from the system.

The other piece of physical information is the cross-wavelet transform data extracted from the vibration signal. This study analyzes the condition of equipment based on data simultaneously collected from two sensors, each with a phase angle of 90°. The phase difference in sensors collecting data from rotating equipment allows for capturing distinct vibration characteristics owing to the rotational trajectory made by the equipment's axis. The method of diagnosing abnormalities in the axis and bearings of rotating equipment based on a 90-degree phase difference has been substantiated through numerous prior studies. In several studies based on rotating equipment, it was common to construct and analyze an orbit, which represents the trajectory of the rotating axis, based on the vibration information acquired from two orthogonal sensors. However, in this study, we use wavelet analysis, which offers richer information than the orbit and simultaneously represents both the frequency and time domains. Among various wavelet-based analytical methodologies, we employ CWT to visualize the frequency response correlation between the two vibration signals. This visualized data are then utilized as the physical information extracted from the given pair of vibration signals. In this study, we employ a strategy where the generator, while producing the vibration signal, concurrently generates a CWT image. Through this strategy, the generator is able to simultaneously learn information in the time domain from direct analysis of the vibration signals, and information in the time–frequency domain by comparing the CWT images. Essentially, in the proposed model, the order frequency information is used as the PI input for both the generator and encoder. To compute the PI loss, we utilize the CWT image to produce the CWT loss, which in turn is used to train both the generator and encoder.

3.1. Physics Guided Generator & Encoder

Typically, a GAN considers the distribution of z as input, with the objective of having the generator learn to produce outputs that closely resemble the desired distribution of x. The goal is for the discriminator to be unable to distinguish between these generated outputs and the real data. In this study, instead of learning a direct function from z to x', we first extract prominent features from x through order analysis. Using these frequency characteristics, we generate a baseline for x' termed x_{base}. This x_{base} serves as an intermediate result, constructed solely based on the extracted features. While it may not be entirely identical to x, it encapsulates prior physical knowledge. Therefore, the BiVi-GAN's generator learns the remaining frequency components and noise that x_{base} fails to represent. That is, major features that can be clarified physically are utilized as prior knowledge, while more elusive phenomena and noises are learned using the data. The encoder follows the structure of BiGAN and is designed to perform a task that is entirely opposite to that of the generator. The encoder considers the real signal x as input and synthesizes z'. In this process, similarly to the generator, it also performs the task of extracting the primary frequency components from z.

While the primary goal of this study is to synthesize vibration data themselves, as an auxiliary task, it also undertakes the generation of cross-wavelet transform images. One reason for this is that during the learning process of the cross-wavelet transform, features are extracted that consider the correlation between the two signals. Consequently, when generating vibration signals, vibrations that consider these features are produced. Each signal is recognized not as an independent entity but as data with correlations. This enables the generation of x' to produce more realistic data.

3.2. Theoretical Convergence

3.2.1. BiGAN with Auxiliary Loss

In this section, we will write generator, encoder, and discriminator as G, E, and D, respectively, and L_1 indicates L-1 Loss. The mini-max problem of the conventional BiGAN structure is expressed in Equation (1) as follows [21]:

$$\begin{aligned} &\min_{G,E}\max_{D} V(D,E,G) = \\ &\min_{G,E}\max_{D} \mathbb{E}_{(x,z)\sim p_{EX}}[\log(D(x,E(x)))] \\ &\quad +\mathbb{E}_{(x,z)\sim p_{GZ}}[\log(1-D(G(z),z))] \end{aligned} \tag{1}$$

The probability distribution of (x, z) follows the distribution, as expressed in Equation (2).

$$\begin{aligned} P_{EX}(x,z) &= P_E(z|x)P_x(x) \\ P_{GZ}(x,z) &= P_G(x|z)P_z(z) \end{aligned} \tag{2}$$

3.2.2. Optimal Discriminator

For fixed G and E, we can identify optimal discriminator D. To identify optimal D, we define D^* as an optimal D.

Theorem 1. $D^* = \frac{p_G(x)\cdot p_{EX}(x,z)}{p_G(x)\cdot p_{EX}(x,z)+p_{data}(x)\cdot p_{GZ}(x,z)}$.

Proof of Theorem 1. We can rewrite the mini-max problem of BiVi in Equation (3) as follows:

$$\begin{aligned} &\min_{G,E}\max_{D} V(D,E,G) \\ &= \min_{G,E}\max_{D} \mathbb{E}_{z\sim p_z}\mathbb{E}_{(x,z)\sim p_{EX}}[\log(D(x,E(x,z)))] \\ &\quad +\mathbb{E}_{x\sim p_{data}}\mathbb{E}_{(x,z)\sim p_{GZ}}[\log(1-D(G(x,z),z))] \\ &\quad +\mathbb{E}_{x\sim p_{data}}\mathbb{E}_{(x,z)\sim p_{GZ}}[L_1(CWT(x),CWT(G(x,z)))] \end{aligned} \tag{3}$$

For fixed G and E, we denote the $\arg\max_D V$ that maximize V as $C(G, E)$. In this case, $C(G, E)$ can be expressed as an integral form in Equation (4) as follows:

$$\begin{aligned} C(G,E) = \quad & \int_z p_z(z) \int_{(x,z)} p_{EX}(x,z)\Big[\log(D(x,z))\Big]d(x,z)dz \\ & + \int_x p_{data}(x) \int_{(x,z)} p_{GZ}(x,z) \\ & \times\Big[\log(1-D(x,z)) + L_1(x,z)\Big]d(x,z)dx \end{aligned} \tag{4}$$

Because distribution of $p_z(z)$ is equal to distribution of $p_G(x)$, $C(G, E)$, can be simplified in Equation (5) as follows:

$$\begin{aligned} C(G,E) = \quad & \int_x \Big[p_G(x) \int_{(x,z)} p_{EX}(x,z)\big[\log(D(x,z))\big] \\ & + p_{data}(x) \int_{(x,z)} p_{GZ}(x,z)\big[\log(1-D(x,z)) \\ & + L_1(x,z)\big]\Big]d(x,z)dx \end{aligned} \tag{5}$$

To obtain optimal D, we can rewrite the equation for G and E as equation for D in Equation (6) as follows:

$$\begin{aligned} C(G,E) &= C'(D) \\ &= \int\int a\cdot b\log(D) + c\cdot d\log(1-D) \\ &\qquad + L_1 d\mu d\phi \end{aligned} \tag{6}$$

where, $a = p_G(x)$, $b = p_{EX}(x,z)$, $c = p_{data}(x)$, and $d = p_{GZ}(x,z)$. For the integral to be maximized, the integrand must be maximal. Let $f(D) = a \cdot b \log(D) + c \cdot d \log(1-D) + L_1$; then, via differentiation, $f'(D) = a \cdot b/D + c \cdot d/(1-D)$. Because f has maximum value at $f' = 0$, we can obtain the following result:

$$\begin{aligned} D^* &= \tfrac{a \cdot b}{a \cdot b + c \cdot d} \\ &= \tfrac{p_G(x) \cdot p_{EX}(x,z)}{p_G(x) \cdot p_{EX}(x,z) + p_{data}(x) \cdot p_{GZ}(x,z)} \end{aligned} \tag{7}$$

Therefore, D has optimal value of $1/2$ when $p_G(x) \times p_{EX}(x,z) = p_{data}(x) \cdot p_{GZ}(x,z)$. □

3.2.3. Convergence of $C(G,E)$

Theorem 2. *$C(G,E)$ has global minimum of $P_G = P_{data}$.*

Proof of Theorem 2. When rearranging the expression for $C(G,E)$ at the optimal D, it is as expressed in Equation (8).

$$\begin{aligned} C(G,E) &= \int_x \Big[p_G(x) \int_{(x,z)} p_{EX}(x,z) \big[\log(D(x,z)) \big] \\ &\quad + p_{data}(x) \int_{(x,z)} p_{GZ}(x,z) \big[\log(1 - D(x,z)) \big] \\ &\quad + L_1(x,z) \Big] d(x,z)dx \\ &= \int_x \int_{(x,z)} p_G(x) p_{EX}(x,z) \big[\log(D^*) \big] \\ &\quad + p_{data}(x) p_{GZ}(x,z) \big[\log(1 - D^*) \big] \\ &\quad + L_1(x,z) \big] d(x,z)dx \end{aligned} \tag{8}$$

When isolating the L_1 portion, it can be expressed in Equation (9) as follows:

$$L_1(x,z) = L_1(CWT(x), CWT(\hat{x})) \tag{9}$$

The L_1 fundamentally approximates the output values by inputting both the original data and the generated data into the same function. Because the wavelet transformation is based on convolution operations via FFT, its inverse transformation can be considered. This indicates that the L_1 corresponding to the wavelet transformation operates mathematically in the same manner as the actual pre-transformation loss, approximating the generated data to the real data. Consequently, approximation through the L_1 has the effect that $P_G \approx P_{data}$. Via the converge of L_1, we can assume that $p_G(x) \approx p_{data}(x) = p(x)$; then, $C(G,E)$ can be rewritten as follows in Equation (10).

$$\begin{aligned} C(G,E) &= \int_x \int_{(x,z)} p_G(x) p_{EX}(x,z) \big[\log(D^*) \big] \\ &\quad + p_{data}(x) p_{GZ}(x,z) \big[\log(1 - D^*) \big] \\ &\quad + L_1(x,z) d(x,z)dx \\ &= \int_x p(x) \Big[\int_{(x,z)} p_{EX}(x,z) \log(D^*) \\ &\quad + p_{GZ}(x,z) \log(1 - D^*) \Big] d(x,z)dx \\ &= \int_x p(x) \cdot [2JS(P_{EX} || P_{GZ}) - \log 4] dx \end{aligned} \tag{10}$$

where the L_1 converges to 0 and consequently vanishes. The convergence of $C(G,E)$ for this case has been proven in the previously presented BiGAN paper [21]. The minimum value for this case is as follows:

$$\begin{gathered} 2JS(P_{EX} || P_{GZ}) - \log 4 \gg -\log 4 \\ \therefore C(G,E) \gg -\log 4 \cdot \int_x p(x)dx = -\log 4 \end{gathered} \tag{11}$$

In conclusion, the adversarial loss and L_1 for $C(G, E)$ converge simultaneously. Owing to the properties of the two converging limits, $C(G, E)$ converge to $-\log 4$.

Therefore, the optimal D is $\frac{1}{2}$, and the generated data and the real data are approximated to have the same distribution as indicated by the above equation. □

4. Experiment

4.1. Experiment Setting

4.1.1. Evaluation Method

When two probability distributions, P and Q, exist, KL divergence and JS divergence are representative quantitative metrics to measure the similarity of the two distributions. The KL divergence, or Kullback–Leibler divergence, calculates the difference between a reference probability distribution and a target probability distribution and is defined as follows [46].

$$KL(p\|q) = \begin{cases} \sum_i p_i \log \frac{p_i}{q_i} = -\sum_i p_i \log \frac{q_i}{p_i} \\ \int p(x) \log \frac{p(x)}{q(x)} dx = -\int p(x) \log \frac{q(x)}{p(x)} dx \end{cases} \tag{12}$$

where P is the reference distribution and Q is the target distribution. The KL divergence has two notable characteristics, and one of them indicates that KL divergence cannot be used as a distance measure between two probability distributions.

$$\begin{gathered} KL(p\|q) \geq 0 \\ KL(p\|q) \neq KL(q\|p) \end{gathered} \tag{13}$$

The inability to use KL divergence as a distance concept is a critical issue when employing it as an evaluation metric for AI models. This is because different values can emerge based on how the reference probability distribution is set. That is, the model performance can be measured differently based on whether the reference probability distribution is from the distribution of values generated by the model or the actual data distribution. Therefore, using KL divergence as a metric to measure the performance of an AI model can be challenging. Consequently, we utilize the Jensen—Shannon divergence, which expands upon the concept of distance inherent in KL divergence [47].

$$JS(p\|q) = \frac{1}{2} KL(p\|\frac{1}{2}(p+q)) + \frac{1}{2} KL(q\|\frac{1}{2}(p+q)) \tag{14}$$

JS divergence is defined as mentioned above, and because it satisfies the commutative property, it is suitable for representing the distance between two distributions. That is, in contrast with KL divergence, it outputs the same value regardless of which distribution is set as the reference point. Therefore, in this study, JS divergence was used as a quantitative evaluation metric for the proposed model.

4.1.2. Dataset

This study aimed to generate 2D vibration data originating from rotating machinery. To evaluate the performance of the proposed BiVi-GAN, we conducted training and testing using the 'Rotating Machinery Fault Type AI Dataset' (Ministry of SMEs and Startups, Korea AI Manufacturing Platform (KAMP), KAIST, 23 December 2022). The dataset, as shown in Figure 8, was acquired at a rotational speed of 1500 RPM using a rotor testbed. The specifications of the equipment used for data collection are listed in Table 1.

Table 1. Rotor testbed setting.

Size	673 mm (W) × 280 mm (D) × 281 mm (H)
Weight	25 kg
Material	Aluminum
Bearing	6202ZZ × 2EA
Motor	DC 12V × 0.25HP (0.2 W), 0~3000 RPM
Main power	220 VAC

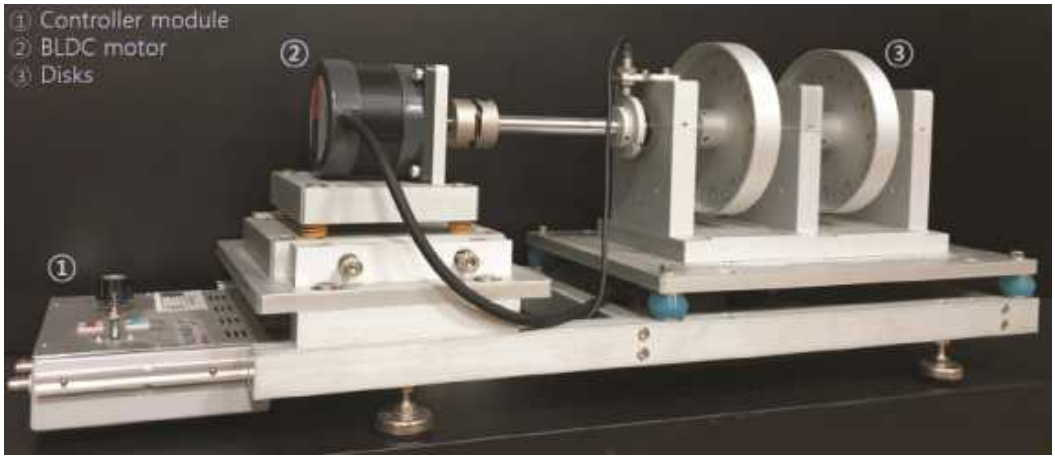


Figure 8. Rotor testbed.

The collected data consisted of four conditions: normal, imbalance, mechanical looseness, and a combination of imbalance and mechanical looseness. Imbalance refers to the most common cause of vibration in rotating machinery. It signifies a condition where the center of mass of the motor does not align with the center of its rotational axis. Mechanical looseness refers to a condition where the equipment is not securely anchored to a flat surface or ground, causing the misalignment or tilting of the equipment. In this study, the four conditions were organized into separate classes, and one-hot encoding was applied. This encoded information was then concatenated with the latent z, forming a class-based GAN structure.

The original data were collected over 140 s from four sensors with a phase difference of 90°. It consisted of a total of 3,772,385 points from four channels. For training, a pair of two sensors with a phase difference of 90° was configured. The window size was set to 1024, and slicing was performed using a stride of 200. In addition to slicing, no other signal processing or preprocessing techniques were applied. The data were used in their original state with considerable noise as input to maintain them in as similar as possible a state to the data collected in a real environment. The training dataset was composed of 7080 slices, with each class being represented equally. Additionally, a test dataset was formed with 2280 slices.

4.1.3. Implementation

BiVi-GAN was implemented using the PyTorch framework, and experiments were conducted in the following computing environment. The generator, discriminator, and encoder of BiVi-GAN were constructed based on one-dimensional convolutional neural networks (1D-CNN) and implemented using a feature pyramid structure for both the encoder and discriminator. However, in the generator, the part responsible for generating cross-wavelet transform images was based on two-dimensional CNN (2D-CNN). It was structured with a network depth identical to the other networks. The generator, encoder, and discriminator networks used convolutional networks with a kernel size of four. Only

the final layer of the discriminator included a sigmoid layer to make a binary discrimination between real and fake samples.

Training was conducted in batches, with each training iteration using a batch size of 128. The generator performed the CWT, and, owing to the inherent nature of GANs, where the generator tended to learn more slowly than the discriminator, there might be a risk of model collapse owing to differences in learning speeds. To compare and analyze the performance of the proposed BiVi-GAN, an ablation study was conducted. A total of six experiments were set up, each consisting of 500 epochs of training and validation under the same conditions. These experiments varied based on whether the CWT loss was used, whether the order frequency was used, and how many order features were chosen to utilize.

4.2. Ablation Study

The proposed BiVi-GAN is built on the foundation of the existing Bidirectional GAN and introduces two key contributions to advance the model. One of the contributions is the utilization of the PI input, which extracts the first-order phase information through order analysis from the input x and incorporates it into the model. The other contribution involves training the model's ability to reconstruct the cross-wavelet transform and use it as an auxiliary task with a PI loss. An ablation study was conducted to assess the impact of these two methodologies on the model's performance and determine the improvement compared with the baseline model. The analysis of the performance variation concerning the utilization of rotational synchrony components involves three conditions: using only the 1x component, using both the 1x and 2x components, and not using any component, thus examining the effect of these configurations on the PI input's performance. Additionally, by examining the conditions of using and not using the cross-wavelet transform loss, the performance changes related to PI loss were analyzed. A total of six experiments were conducted, and the results of each experiment can be quantitatively analyzed using JS divergence. The JS divergence value approaching zero indicated that the data generated by the model were similar to the original data, while a value close to one indicated dissimilarity with the original data. The results of each experiment are listed in Table 2.

Table 2. Ablation study result.

Experiment No.	Order Analysis	Cross-Wavelet Transform	JS Div
1	x	o	0.774
2	x	x	0.821
3	1x	o	0.360
4	1x	x	0.474
5	2x	o	0.230
6	2x	x	0.363

Based on the experimental results, the average JS divergence for the model using the PI input was 0.35, while the model that did not use it had an average JS divergence of 0.79. On average, this represents a 55.26% improvement in performance. Particularly, it should be noted that there was a 47.71% improvement in performance when using only the 1x component and 62.82% improvement when using both the 1x and 2x components. This suggests that using the PI input assisted in effectively capturing the vibration characteristics, contributing to improved performance. Furthermore, when using PI loss with cross-wavelet transform images, the average JS divergence was 0.45, which is approximately a 17.73% improvement compared with the model's average of 0.55 when not using PI loss. Both sets of results provided experimental evidence that the two contributions proposed in BiVi-GAN indeed contributed to improved performance. Finally, when both PI input and PI loss are applied in BiVi-GAN, it achieves a JS divergence of 0.23. This represents a

70% improvement compared with the baseline model, Bidirectional GAN, which had a JS divergence of 0.77. It is evident that applying domain-specific expertise to data-driven deep learning models yielded significantly better performance compared with only using data.

4.3. Validation of GAN Data via Training

To verify whether the vibration data generated through the proposed BiV-GAN are effective as input data for actual learning-based methodologies, experiments were conducted to train one of the most representative deep learning models used in data analysis methodologies based on deep learning. The deep learning model used in the experiment utilizes a CNN model consisting of 3 Conv layers and 1 FC layer. The data were trained in the same learning environment, using the same model and the same hyperparameters as outlined in the following Table 3. The performance of the proposed methodology was validated using the learning dataset by training under three different dataset environments and performing tests on real data to evaluate the performance of the trained model.

Table 3. Train parameter setting.

Parameter	Setting
o	0.774
x	0.821
o	0.360
x	0.474
o	0.230
x	0.363

In the first training, the learning was conducted using only the data actually acquired. In the second training, learning was performed using only the data generated through the proposed methodology, and in the final experiment, both the generated data and the actual data were utilized for the training. A classification task was carried out for a total of four classes (normal, imbalance, mechanical looseness, and a combination of imbalance and mechanical looseness), and to evaluate the quantitative performance, Precision, Recall, and F1 score were used. Precision represents the ratio of actual positives among the positive judgments made by the model, while Recall represents the ratio of positives correctly identified by the model out of the actual positives. The F1 score is obtained through the harmonic mean of Precision and Recall, with all three indicators having values between 0 and 1, where values closer to 1 indicate better model performance. The results of the experiments are listed in Table 4.

Table 4. Data validation experiment result.

Experiment No.	Train Data	Precision	Recall	F1 Score
1	Real Only	0.728	0.715	0.722
2	Generated Only	0.681	0.685	0.683
3	Real + Generated	0.815	0.816	0.816

According to the experimental results, training with only real data resulted in an F1 score of 0.722, while training with only generated data achieved an F1 score of 0.683. When both types of data were used for training, an F1 score of 0.816 was achieved, resulting in an approximate 13-percent improvement in performance, marking the highest performance achieved. These results demonstrate that in cases where there is insufficient or a lack of diversity in the available data, using the proposed BiVi model to synthesize data

and incorporating it into the training can be beneficial, especially for utilizing bearing vibration data.

5. Conclusions

In this study, we proposed BiVi-GAN for generating vibration data in rotating machinery. Our proposed model leveraged the strengths of deep learning in capturing data characteristics and combined them with the physical knowledge of PINN. Consequently, we were able to generate data that closely resembled real-world data. Using the cross-wavelet transform as PI loss and order-analysis results as PI input, we observed a dramatic improvement in performance, particularly when applying PI input. In the future, we aim to conduct a study on methodologies that allow the application of physical information to other tasks. We intend to explore how expertise in machine engineering, where substantial physical knowledge is required, can be effectively complemented by AI technologies.

Author Contributions: Conceptualization, H.J. and S.J.; methodology, H.J. and J.K.; software, S.J.; validation, H.J. and S.J.; formal analysis, H.L.; investigation, H.J.; resources, J.K.; data curation, H.L.; writing—original draft preparation, H.J. and S.J.; writing—review and editing, S.J.; visualization, H.J.; supervision, J.K.; project administration, H.J.; funding acquisition, H.L. and J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government (MOTIE) (20224B10100060, Development of Artificial Intelligence Vibration Monitoring System for Rotating Machinery).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in KAMP (Korea AI Manufacturing Platform) at https://www.kamp-ai.kr/aidataDetail?.

Conflicts of Interest: Author HyunJun Lee was employed by the company RMS Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Shen, C.; Wang, D.; Kong, F.; Tse, P.W. Fault diagnosis of rotating machinery based on the statistical parameters of wavelet packet paving and a generic support vector regressive classifier. *Measurement* **2013**, *46*, 1551–1564. t.2012.12.011. [CrossRef]
2. Ebrahimi, B.M.; Faiz, J.; Roshtkhari, M.J. Static-, Dynamic-, and Mixed-Eccentricity Fault Diagnoses in Permanent-Magnet Synchronous Motors. *IEEE Trans. Ind. Electron.* **2009**, *56*, 4727–4739. [CrossRef]
3. Lei, Y.; Lin, J.; He, Z.; Zuo, M.J. A review on empirical mode decomposition in fault diagnosis of rotating machinery. *Mech. Syst. Signal Process.* **2013**, *35*, 108–126. [CrossRef]
4. Yin, S.; Kaynak, O. Big Data for Modern Industry: Challenges and Trends [Point of View]. *Proc. IEEE* **2015**, *103*, 143–146. [CrossRef]
5. Qiao, W.; Lu, D. A Survey on Wind Turbine Condition Monitoring and Fault Diagnosis—Part I: Components and Subsystems. *IEEE Trans. Ind. Electron.* **2015**, *62*, 6536–6545. [CrossRef]
6. Lei, Y.; Lin, J.M.; Zuo, J.; He, Z. Condition monitoring and fault diagnosis of planetary gearboxes: A review. *Measurement* **2014**, *48*, 292–305. [CrossRef]
7. Samanta, B. Gear fault detection using artificial neural networks and support vector machines with genetic algorithms. *Mech. Syst. Signal Process.* **2004**, *18*, 625–644. [CrossRef]
8. Hajnayeb, A.; Ghasemloonia, A.; Khadem, S.E.; Moradi, M.H. Application and comparison of an ANN-based feature selection method and the genetic algorithm in gearbox fault diagnosis. *Expert Syst. Appl.* **2011**, *38*, 10205–10209. [CrossRef]
9. Zhang, X.; Wang, B.; Chen, X. Intelligent fault diagnosis of roller bearings with multivariable ensemble-based incremental support vector machine. *Knowl.-Based Syst.* **2015**, *89*, 56–85. [CrossRef]
10. Jegadeeshwaran, R.; Sugumaran, V. Fault diagnosis of automobile hydraulic brake system using statistical features and support vector machines. *Mech. Syst. Signal Process.* **2015**, *52–53*, 436–446. [CrossRef]
11. Jia, F.; Lei, Y.; Lin, J.; Zhou, X.; Lu, N. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mech. Syst. Signal Process.* **2016**, *72–73*, 303–315. [CrossRef]

12. Li, Y.; Lekamalage, C.K.L.; Liu, T.; Chen, P.-A.; Huang, G.-B. Learning Representations With Local and Global Geometries Preserved for Machine Fault Diagnosis. *IEEE Trans. Ind. Electron.* **2020**, *67*, 2360–2370. [CrossRef]
13. Mikolajczyk, A.; Grochowski, M. Data augmentation for improving deep learning in image classification problem. In Proceedings of the 2018 International Interdisciplinary PhD Workshop (IIPhDW), Swinoujście, Poland, 9–12 May 2018; pp. 117–122. [CrossRef]
14. Bandara, K.; Hewamalage, H.; Liu, Y.-H.; Kang, Y.; Bergmeir, C. Improving the accuracy of global forecasting models using time series data augmentation. *Pattern Recognit.* **2021**, *120*, 108148. [CrossRef]
15. Cui, X.; Goel, V.; Kingsbury, B. Data Augmentation for Deep Neural Network Acoustic Modeling. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 1469–1477. [CrossRef]
16. Kang, Y.; Hyndman, R.J.; Li, F. GRATIS: GeneRAting TIme Series with diverse and controllable characteristics. *Stat. Anal. Data Min. Asa Data Sci. J.* **2020**, *13*, 354–376. [CrossRef]
17. Keylock, C.J. Constrained surrogate time series with preservation of the mean and variance structure. *Phys. Rev. E* **2006**, *73*, 036707. [CrossRef] [PubMed]
18. Zio, E. Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice. *Reliab. Eng. Syst. Saf.* **2022**, *218*, 108119. [CrossRef]
19. Chao, M.A.; Kulkarni, C.; Goebel, K.; Fink, O. Fusing physics-based and deep learning models for prognostics. *Reliab. Eng. Syst. Saf.* **2022**, *217*, 107961. [CrossRef]
20. Karpatne, A.; Atluri, G.; Faghmous, J.H.; Steinbach, M.; Banerjee, A.; Ganguly, A.; Shekhar, S.; Samatova, N.; Kumar, V. Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2318–2331. [CrossRef]
21. Donahue, J.; Krähenbühl, P.; Darrell, T. Adversarial Feature Learning. *arXiv* **2017**, arXiv:1605.09782.
22. Frid-Adar, M.; Klang, E.; Amitai, M.; Goldberger, J.; Greenspan, H. Synthetic data augmentation using GAN for improved liver lesion classification. In Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018; pp. 289–293. [CrossRef]
23. Patel, M.; Wang, X.; Mao, S. Data augmentation with conditional GAN for automatic modulation classification. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, Linz, Austria, 13 July 2020; pp. 31–36. [CrossRef]
24. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
25. Antoniou; Antreas; Storkey, A.; Edwards, H. Data augmentation generative adversarial networks. *arXiv* **2017**, arXiv:1711.04340.
26. Nussbaumer, H.J. The Fast Fourier Transform. In *Fast Fourier Transform and Convolution Algorithms*; Springer Series in Information Sciences; Springer: Berlin/Heidelberg, Germnay, 1982; Volume 2.
27. Oppenheim, A.V. *Discrete-Time Signal Processing*; Pearson Education: Bangalore, India, 1999.
28. Zhang, D. Wavelet transform. *Fundam. Image Data Min. Anal. Featur. Classif. Retr.* **2019**, 35–44 .
29. Saito, Y.; Takamichi, S.; Saruwatari, H. Text-to-Speech Synthesis Using STFT Spectra Based on Low-/Multi-Resolution Generative Adversarial Networks. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5299–5303. [CrossRef]
30. Lan, T.; Hu, Q.; Liu, X.; He, K.; Yang, C. Arrhythmias Classification Using Short-Time Fourier Transform and GAN Based Data Augmentation. In Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 308–311. [CrossRef]
31. Bashar, S.K.; Bhuiyan, M.I.H. Classification of motor imagery movements using multivariate empirical mode decomposition and short time Fourier transform based hybrid method. *Eng. Sci. Technol. Int. J.* **2016**, *19*, 1457–1464. [CrossRef]
32. Roy, S.; Dora, S.; McCreadie, K.; Prasad, G. MIEEG-GAN: Generating Artificial Motor Imagery Electroencephalography Signals. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8. [CrossRef]
33. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
34. Li, W.; Bazant, M.Z.; Zhu, J. A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches. *Comput. Methods Appl. Mech. Eng.* **2021**, *383*, 113933. 3933. [CrossRef]
35. Cho, I.H. A framework for self-evolving computational material models inspired by deep learning. *Int. J. Numer. Methods Eng.* **2019**, *120*, 1202–1226. [CrossRef]
36. Lagergren, J.H. Nardini, J.T.; Baker, R.E.; Simpson, M.J.; Flores, K.B. Biologically-informed neural networks guide mechanistic modeling from sparse experimental data. *PLoS Comput. Biol.* **2020**, *16*, e1008462. [CrossRef] [PubMed]
37. Yucesan, Y.A.; Viana, F.A.C. Hybrid physics-informed neural networks for main bearing fatigue prognosis with visual grease inspection. *Comput. Ind.* **2021**, *125*, 103386. [CrossRef]
38. Jiang, Z.; Zhang, K.; Xiang, L.; Yu, G.; Xu, Y. A time-frequency spectral amplitude modulation method and its applications in rolling bearing fault diagnosis. *Mech. Syst. Signal Process.* **2023**, *185*, 109832. [CrossRef]
39. Chen, B.; Shen, B.; Chen, F.; Tian, H.; Xiao, W.; Zhang, F.; Zhao, C. Fault diagnosis method based on integration of RSSD and wavelet transform to rolling bearing. *Measurement* **2019**, *131*, 400–411. [CrossRef]

40. Shao, H.; Xia, M.; Wan, J.; de Silva, C.W. Modified Stacked Autoencoder Using Adaptive Morlet Wavelet for Intelligent Fault Diagnosis of Rotating Machinery. *IEEE/ASME Trans. Mechatronics* **2022**, *27*, 24–33. [CrossRef]
41. Han, S.; Shao, H.; Cheng, J.; Yang, X.; Cai, B. Convformer-NSE: A Novel End-to-End Gearbox Fault Diagnosis Framework Under Heavy Noise Using Joint Global and Local Information. *IEEE/ASME Trans. Mechatronics* **2023**, *28*, 340–349. [CrossRef]
42. Goyal, D.; Dhami, S.S.; Pabla, B.S. Non-Contact Fault Diagnosis of Bearings in Machine Learning Environment. *IEEE Sens. J.* **2020**, *20*, 4816–4823. [CrossRef]
43. Tang, H.; Liao, Z.; Chen, P.; Zuo, D.; Yi, S. A novel convolutional neural network for low-speed structural fault diagnosis under different operating condition and its understanding via visualization. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 3501611. [CrossRef]
44. Tang, H.; Tang, Y.; Su, Y.; Feng, W.; Wang, B.; Chen, P.; Zuo, D. Feature extraction of multi-sensors for early bearing fault diagnosis using deep learning based on minimum unscented kalman filter. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107138. [CrossRef]
45. Tang, H.; Liao, Z.; Chen, P.; Zuo, D.; Yi, S. A robust deep learning network for low-speed machinery fault diagnosis based on multikernel and RPCA. *IEEE/ASME Trans. Mechatronics* **2021**, *27*, 1522–1532. [CrossRef]
46. Kullback, S.; Leibler, R.A. On Information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
47. Lin, J. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* **1991**, *37*, 145–151. 1115. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

cancers

MDPI

Review

Machine Learning Meets Cancer

Elena V. Varlamova [1,*], Maria A. Butakova [1], Vlada V. Semyonova [2], Sergey A. Soldatov [1], Artem V. Poltavskiy [1], Oleg I. Kit [3] and Alexander V. Soldatov [1]

[1] The Smart Materials Research Institute, Southern Federal University, 178/24 Sladkova Str., 344090 Rostov-on-Don, Russia; mbutakova@sfedu.ru (M.A.B.); sergeysoldatov@sfedu.ru (S.A.S.); poltavsky@sfedu.ru (A.V.P.); soldatov@sfedu.ru (A.V.S.)
[2] Faculty of Computer Science, HSE University, 20 Myasnitskaya Str., 101000 Moscow, Russia; vvsemyonova_1@edu.hse.ru
[3] National Medical Research Centre for Oncology, 63 14 Liniya Str., 344037 Rostov-on-Don, Russia; onko-sekretar@mail.ru
* Correspondence: var@sfedu.ru

Simple Summary: This review examines the latest technologies using machine learning (ML) methods, including the use of convolutional neural networks, decision trees, and generative adversarial networks to solve problems concerning cancer recognition, planning optimal treatment strategies, as well as predicting the likelihood of patient survival. The authors also discuss the prospects of using machine learning technologies in medicine. In addition, the authors emphasize the need to address issues such as the anonymization of data received from patients.

Abstract: The role of machine learning (a part of artificial intelligence—AI) in the diagnosis and treatment of various types of oncology is steadily increasing. It is expected that the use of AI in oncology will speed up both diagnostic and treatment planning processes. This review describes recent applications of machine learning in oncology, including medical image analysis, treatment planning, patient survival prognosis, and the synthesis of drugs at the point of care. The fast and reliable analysis of medical images is of great importance in the case of fast-flowing forms of cancer. The introduction of ML for the analysis of constantly growing volumes of big data makes it possible to improve the quality of prescribed treatment and patient care. Thus, ML is expected to become an essential technology for medical specialists. The ML model has already improved prognostic prediction for patients compared to traditional staging algorithms. The direct synthesis of the necessary medical substances (small molecule mixtures) at the point of care could also seriously benefit from the application of ML. We further review the main trends in the use of artificial intelligence-based technologies in modern oncology. This review demonstrates the future prospects of using ML tools to make progress in cancer research, as well as in other areas of medicine. Despite growing interest in the use of modern computer technologies in medical practice, a number of unresolved ethical and legal problems remain. In this review, we also discuss the most relevant issues among them.

Keywords: artificial intelligence; machine learning; oncology; radiomics; PET/CT

1. Introduction

Cancer is one of the most common diseases in the world. In 2022, 1,930,000 cases of this disease were detected worldwide. Lung cancer is the leader in regard to the number of deaths. Using computer-aided diagnosis (CAD) can significantly speed up the process of cancer diagnosis. In order to use medical images as input data for CAD, it is necessary to find a way to effectively identify features in images. Using machine learning in CAD systems can improve the accuracy of histopathological diagnoses. Over the past 5 years, AI has demonstrated unprecedented progress in the field of computer vision (CV). Methods

of extracting dependencies from data, implemented with the help of AI, represent great potential for cancer diagnostics. AI technologies are beginning to be widely used in screening, diagnosis, predicting future outcomes, disease monitoring, treatment planning, and clinical oncology research in general. An important novel direction is the development of computer-driven systems for personalized medicine, namely the individual synthesis of the necessary medical substances directly at the bedside. Thus, this approach could be of great importance for the future success of medicine.

2. Diagnostics

Radiomics is the subdiscipline of AI dealing with the computation, identification, and extraction of image features, as well as the generation of prognostic or predictive mathematical models. It is a novel, developing, supplementary source of diagnostic information used for analyzing clinical data based on ML [1]. It is a non-invasive method. In silico research needs data to make a breakthrough for this clinical approach. It is possible to identify various parameter correlations. It only requires the data to be available for a representative sample of patients. This is the reason why the benefits of using AI are extensive.

One of the problems in radiation oncology today is the uncertainty in creating medical diagnostic image descriptions, which directly affect the model for predicting treatment and patient care. This is often expressed by the large variability in the definition of images, represented by the structural marks made by the doctor. In a prior study [2], W.C. Sleeman et al. proposed an approach to change the description of images according to the American Association of Physicists in Medicine's (AAPM) Task Group 263 (TG- 263). Scientists selected datasets with diagnostic images for the first 709 patients with lung cancer and 752 patients with prostate cancer from 40 medical centers. Training datasets were collected as part of the Radiation Oncology Quality Surveillance Program (VAROQS). A comparative analysis was carried out on the Apache Spark platform using machine learning algorithms, such as Naive Bayes (NB), Random Forest (RF), Gradient Boost Tree (GBT), Multilayer Perceptron (MLP), and Support Vector Machine (SVM) techniques. The authors constructed signs of a bony anatomy that were used as additional vectors for the comparative analysis. The optimization of the learning process was performed by reducing the dimension of these features using the singular value decomposition (SVD), which gave an error within 0.1% at a higher speed, in contrast to the traditional principal component analysis (PCA). The data from 50 lung cancer patients and 50 prostate cancer patients, provided by Virginia Commonwealth University (VCU), were used as test datasets for validating the pre-trained models. The best result was demonstrated by the RF algorithm at 98.77 for lung cancer and 95.06 for prostate cancer. In general, using the prepared data, all the methods showed accuracy at the level of 90%, except for NB, and using the clinical data, the results was comparable to the Virginia Commonwealth University (VCU) external test dataset, and here MLP showed itself to be the best of all. The presence of signs of a bony anatomy in all cases increased the accuracy of the measurements, so even when using clinical kits, the results were 95% for lung cancer and 91% for prostate cancer.

An indicator of normal or pathogenic biological processes in the body or reactions to therapeutic intervention is a biomarker. F. Klauschen et al., in a prior study [3], describe how a qualitative analysis of a new intracranial formation plays a crucial role in diagnosis and can determine whether a confirmatory biopsy or resection should be performed in clinical practice. Their research in the field of image analysis is aimed at extracting information from a set of image data for further creation of biomarkers, the schematic diagram of which is shown in Figure 1.

There are several phases of radiomics: image preprocessing, feature evaluation, feature selection, classification, and evaluation. An important part of the study is preprocessing: correction of intensity, unevenness, as well as noise reduction. The ML methods can be successfully used to solve the problems described above. This review describes the studies aimed at visualizing biomarkers for monitoring human responses to treatment, as well as

for diagnosis and prognosis. The use of the imaging method would make it possible to distinguish patients with true progression from patients with pseudo-progression. This would make it possible to change the treatment strategy and prevent ineffective treatment. As a result of the selection of signs, 600,000 positron emission tomography (PET) clusters based on three classes were obtained, and the accuracy of the diagnosis of pseudo-progression and tumor progression was 90%. It was proved that in several types of cancer, the number of tumor-infiltrating lymphocytes (TILs), along with other biomarkers, is a marker of the response to immune checkpoint therapy. In their paper, A. Klauschen et al. discuss automated approaches to calculate the number of TILs. Both classical approaches and non-traditional approaches were considered. Explicable ML allows medical specialists to interpret the result for patients. The ability to scan the whole slide images (WSIs) and the availability of solutions that can process the WSIs are attracting much attention to digital pathology. Breast cancer is an example where TIL scoring is important. The article discusses various predictors of the body's response to therapy. The authors consider segmentation and direct classification methods. Classical segmentation methods require small amounts of input data. Among them, threshold segmentation, the watershed method, and segmentation based on morphology are considered. The main issue in applying the direct approach is that the typical size of the input image for the deepest neural networks (NNs) is 300 × 300 pixels. Currently, the WSIs have dimensions up to 10,000 × 10,000 pixels. The authors discuss other approaches in which restrictive frames are displayed around the cells as an alternative to segmentation. Examples of such methods include Fast-R-CNN, Faster R-CNN, and others. In such methods, a sliding window is used, which moves around the image and transmits location information if the window falls into a certain cell. The result and its reliability are not always easy for a specialist to interpret. There are several approaches that are used to explain the prediction result, such as Guided Backprop, sensitivity analysis, integrated gradients, and DeepLIFT. These methods work in different ways. For example, a gradient-based method shows which regions make the greatest contribution to the prediction. The authors note that in order to use ML approaches in cancer diagnosis everywhere, it is necessary to develop clear criteria and evaluation standards using benchmark datasets.

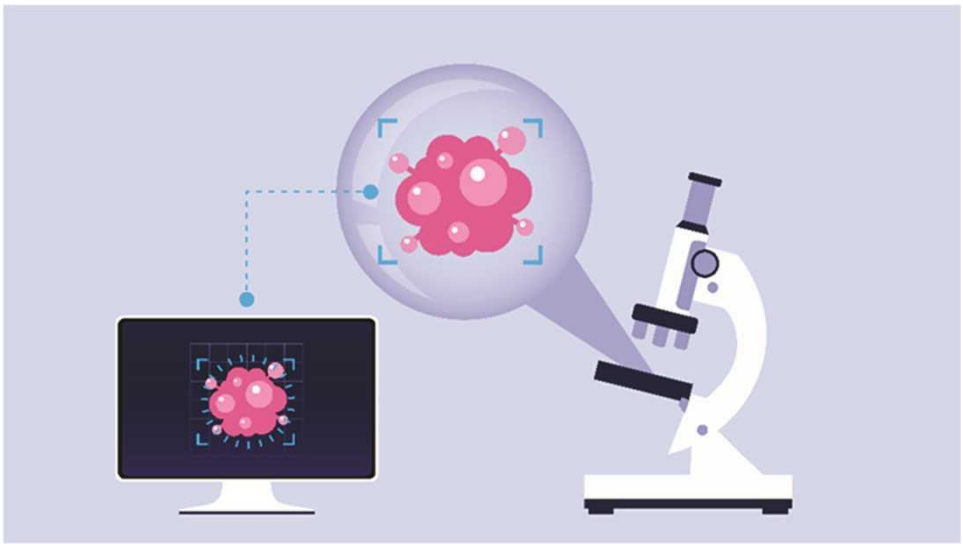


Figure 1. Diagnosis of oncology using computer technology.

Big data phenomena give new opportunities in radiation oncology as well. However, the problem of effectively processing big data retrieved from multiple sources and different locations for radiation oncology clinics remains unsolved. J. Kazmierska et al., in their work [4], tried to identify the barriers and limitations of big data adoption in the clinical data science community in radiation oncology. Multisource big data in radiation oncology

clinics arise from constantly increasing data flows, including electronic health records, patients' laboratory measures, output data, medical images (PET, computed tomography (CT), cone-beam computed tomography (CBCT), magnetic resonance imaging (MRI)), and the features extracted from images. The authors noticed that the clinicians themselves were not able to process such huge datasets without clear rules, and the AI approaches coupled with AI/ML could be really beneficial for scientists.

In [5], R. Seifert et al. discuss in detail the application of AI in various fields of nuclear medicine imaging and the directions of advanced scientific developments. The authors provide examples of applying AI for image analysis, pre- and post-processing of images, as well as for predicting the success of treatment and psychotherapy.

Although the use of AI in nuclear medicine imaging is at an early stage of its development, there is great scientific potential for the use of imaging systems in clinical practice. There are several ways to use image analysis as a part of AI for the segmentation of tumor PETs and for the quantification of a whole-body tumor volume. Currently AI is being used to improve fade correction of PET images, image capture and recovery, automated disease classification, and automated metastatic detection. The authors of the study emphasize three approaches to using image analysis in nuclear medicine. First, ML is employed for the specification of body subregions. Second, AI is employed to quantify the whole-body tumor volume in PET acquisitions. Finally, AI for pathological accumulation segmentation is commonly used in a variety of cancer research, like prostate-specific membrane antigen (PSMA)-PET, bone scintigraphy, or fluorodeoxyglucose (FDG-PET). ML imagining is carried out in nuclear cardiology to determine whether surgical intervention is necessary for the patient. PET and SPECT usually combine imaging scans with CT. The study analyzed the data from 713 patients who underwent myocardial perfusion scintigraphy and received cardiac catheterization. Based on these results, the data parameters that represented information gain were selected from 33 total parameters and a prediction model was then trained. The algorithm showed the results comparable or superior to human operators in predicting the need for surgical intervention. The application of ML makes it possible to integrate many variables that a person is not able to supervise. AI and ML are also actively used in dealing with benign diseases. In addition to classification, AI effectively correlates the dependence of various clinical factors with treatment options.

Individual processing can also be performed by AI, this is especially important in time-consuming processes such as complete image reconstruction. This leads to a decrease in the number of indicators used and prevents the need for simultaneous acquisition of CT, which results in a decrease in radiation exposure. The data collection time is also reduced, which increases the number of studies.

One of the main promising directions of the ML development for cancer diagnostics is the separation of a metastatic tumor in the body, taking into account various parameters of metastases and tumors, the accurate determination of changes in the volume of each metastasis, the study of the tumor for lesions in vivo, the analysis and processing of dynamic PET data, and automatic image noise reduction.

The confidentiality of patient data while organizing the general access to medical diagnostic data for solving specialized problems is an important issue for the healthcare sector in terms of progress and innovation development. The study in [6] presents a distributed structure of Personal Health Train (PHT) based on the FAIR data model (Findable, Accessible, Interoperable, Reusable), which allows ML and data mining in a decentralized manner while maintaining patient privacy. T.M. Deist et al. used the total data coverage at eight medical organizations in five countries, which made it possible to provide information on 23,203 clinical cases. The authors trained the ML Logistic Regression (LR) model on 14,810 patients (treated from 1978 to 2011) and confirmed it on 8393 patients treated from 2012 to 2015. As a result, a comparative analysis has been conducted with an alternative platform for distributed data analysis and ML DataSHIELD. The possible weak points of PHT are also highlighted.

Modern clinical diagnostic studies in medical institutions generate a large amount of heterogeneous data. Precision medicine methods such as next-generation sequencing involve evaluating tens of thousands of parameters when diagnosing one patient. In [7], J.-E. Bibault et al. provided an evidence base for the effectiveness of using Big Data technology for the digital support of modern radiation oncology. A summary analysis of the advantages and disadvantages of the ML algorithms was performed. The considered ML models included Decision Tree (DT), NB, k-means (kM), SVM, NN and deep learning (DL). Scientists have considered various possibilities for obtaining high-quality data when creating predictive models. Discussing the advantages of using the DT algorithm, it can be noted that it is fast and accessible for the user to understand the mechanism of its operation. Nevertheless, for its successful application, it is necessary to observe the rule of mutually exclusive classes and determine a clear order of attribute selection. In addition, there is a risk of retraining. Numeric attributes must comply with the law of normal distribution. The same limitations exist for the NB and k-NN algorithms, but the k-NN algorithm has a significant advantage; it is resistant to possible misplaced values in the dataset and to noise. Scientists note that the k-NN algorithm can be applied to nonlinear classification, and its model is robust. Discussing the SVM and NN models, it is impossible not to note the long time spent on training and operating the models, the risk of retraining, and the difficulties often encountered in interpreting the results. The search for the optimal network architecture often takes place through a series of tests. Nevertheless, these models are resistant to noise and missed dates, limit the risk of error, and can be used to solve both regression and classification problems. These and many other advantages are the reason for their popularity in solving problems of recognizing malignant tumors and predicting the probability of patient survival.

The data regimentation and registration on new cancer cases require a lot of human resources. To solve this problem and to provide the automated input of data on the new cases of pathologies of prostate adenocarcinoma, the authors of [8] proposed a pre-trained ML model based on the SVM method. The data collection for the study took place at the Bas-Rhin registry in two stages. First, 982 cases of pathologies in 2014 were analyzed to build predictions for 785 cases filed in 2015. Then, 2089 cases in 2014–2015 were analyzed, and 926 possible pathologies were presented in 2016. The overall accuracy for the interventions was about 97%, the SVM method was well-demonstrated on the unstructured reports of pathologies, and the main errors were associated with the patients' place of residence. To achieve the results obtained and reduce the number of pathology reports, the authors performed a preliminary analysis of all incoming data. This approach consisted of searching the keyword "simple" in the patient's history, as well as the exclusion of uninformative records and the removal of unnecessary characters. The R software (version 3.5) and rWeka package https://cran.r-project.org/web/packages/RWeka/index.html (accessed on 12 December 2023) were used as the tools for the implementation of the developed algorithms.

Every year, more than 600,000 cases of colon cancer are detected in the world. One of the main aims of applying ML in medicine is to build and improve the accuracy of predictive models for the development of the disease, the effectiveness of the prescribed treatment, or the occurrence of postoperative complications and re-hospitalization. In [9], the authors applied DL methods to diagnose six types of colonic mucosal lesions using convolutional neural networks (CNNs). Scientists used pre-trained deep CNN of the ResNet and EfficientNet architectures to develop an algorithm for automatic segmentation of WSI colon biopsies. The authors compared several methods and approaches for solving the classification problem and concluded that the multiple-label approach was preferable because some WSI fixes may belong to more than one class or none of them. As a result, they achieved an area under the receiver operator characteristic (ROC) (AUC) curve of up to 0.96.

In [10], E.L. Barber et al. proposed improving the accuracy of such methods based on the use of natural language processing (NLP) on unstructured data. They collected a set of medical reports with preoperative CT in women with ovarian cancer to isolate addi-

tional sign vectors. To form the required datasets, the Northwestern Medicine Enterprise (NM-EDW) repositories were used, in which patients were over the age of 18 years old and underwent surgery between 2011 and 2017. The authors reported that all language deviations had been removed to extract the target functions during data preprocessing. Unigrams and bigrams were chosen as the form of extraction since they showed the best results. Further, the authors formed three sets of NLP functions, where in the first case, a token dictionary was used together with LR using the LASSO method to reduce the dimension. The second set used all non-zero coefficients from the first case, applying principal component analysis. For the final set of functions, word2vec was used to generate patient vectors for the dictionary matrix tokens. The authors implemented two models of outcomes according to the degree of postoperative complications. For verification, 5-fold cross-validation was implemented. This reportedly resulted in a slight increase in AUC of 0.01 on average for both cases. The transfer of the obtained results into the final prediction model made it possible to achieve a 20–25% increase in the accuracy of predictions.

It is possible to predict a pathological response in gene overexpression from cancer patients receiving neoadjuvant chemotherapy (NAC). Research was carried out on breast cancer, evaluating intratumor HER2 gene expression [11]. The final model had high sensitivity of 99.3%, specificity of 81.3%, positive predictive value (PPV) of 97.9%, negative predictive value (NPV) of 92.9%, and diagnostic accuracy of 96.4%. The results show that this model may be helpful in clinical decision making. In terms of successful validation in larger multi-institutional studies, these results could be used to better select those patients who could benefit from anti-HER2 treatment.

In [12], Hsiao-Yu Yang et al. built ML prediction models for breast cancer using the breath biopsy principle. This principle helps to make decisions both during treatment and during surgery. The implemented breath biopsy system uses a sensor array to determine the structure of biomarkers and supervised machine learning methods. A carbon dioxide (CO_2) monitoring system receives the purest alveolar air, excluding eating, oral, and smoking contaminations. If the exhaled breath from the lungs has a high enough dose of specific volatile metabolites (or volatile organic compounds, VOCs) to be detected by the sensor array, then those features can serve as biomarkers. In this study, the collected air was analyzed by the Cyranose electronic nose (E-nose) 320, a chemical vapor sensing instrument consisting of 32 chemical sensors based on nanotubes in the standard array which converts some VOCs in the breath to a sensor resistance response. Technically, gas chromatography in combination with mass spectrometric analysis is performed by the E-nose, and the analysis of respiration biomarkers with specific VOCs is collected into datasets. For example, the breast cancer pattern contains pentane VOCs as a disease marker and can be detected by the E-nose with reasonable accuracy of 90% and higher. The datasets were collected for two years (2016–2018), and a total of 899 subjects were screened and estimated. The ML model and parameters consist of the following algorithms: k-nearest neighbors (k-NN), NB, DT, NN, SVM with linear, radial, and polynomial kernels, and RF. The authors reported that the RF model surpassed the others, having the following parameters. The prediction accuracy of breast cancer in the dataset was 91%, and the AUC was 0.99 on a 95% confidence interval (CI) of ROC. PPV and NPV were the same, at 97%, respectively.

It is a well-known fact that early diagnosis of lung cancer increases the patient's chances for recovery, as well as social and labor adaptation. The analysis of metabolomic biomarkers and changes in a metabolite pattern at an early stage provides an informative approach to the assessment of tumor progression and plays a crucial role in distinguishing the tumor stages. Such an analysis has a different degree of sensitivity, which indicates different stages of lung cancer. As a rule, if metabolomic biomarkers are detected as part of the early diagnosis, additional examination is prescribed. The process of discovering and identifying metabolites as promising biomarkers is implemented in the study of Ying Xie [13]. According to the study, six ML techniques were applied to 110 lung tumor patients at stage I and 43 healthy individuals; the best AUC value achieved was 0.923

(95% CI: 0.871–0.975), with a sensitivity of 79.6% and specificity of 93.0%. The data preparation in this study began with the metabolomic chromatographic separation of the collected assays, then liquid chromatography–mass spectrometry (LC-MS) analysis was performed using a Waters ACQUITY UPLC coupled with a 4000 Q-TRAP mass spectrometer. All LC-MS data were preprocessed and normalized (AB Analyst Software package version 1.6.2), multivariate statistical analysis (SIMCA-p14 package (Umetrics AB, Umeå, Sweden)) was applied, and principal component analysis (PCA) was selected for dimensionality reduction problem solving. Based on 46 influential metabolomic biomarkers with statistical significance for stage I lung tumor patients, the top 10 metabolic biomarkers with higher diagnostic value (AUC > 0.800) were chosen. These 10 biomarkers made it possible to clearly separate patients with stage I lung cancer from healthy persons. These data composed a training set for the k-NN, NB, AdaBoost, SVM, RF, and NN ML techniques. The authors reported that the NB, RF, SVM, and NN models showed the best prediction quality (AUC was 1.000), and the sensitivity of these models was from 0.909 to 1.000, which is enough to recognize the early lung tumor stage and appropriate for diagnosis.

Prostate cancer can be identified by investigating a specific gene, prostate cancer gene 3 (PCA3). PCA3 observation shows the overexpression of this gene in tumor tissues in 95% of cases, and thus it is an essential biomarker candidate. The authors of [14] used their original low-cost biosensors for PCA3 detection to avoid complex, specialized approaches that require polymerase chain reaction, optical changes measurement in the emission of light and fluorescence emission methods, and transcription-mediated amplification. The biosensors were fabricated using multi-walled carbon nanotubes and layer-by-layer (LbL) films of chitosan with interdigitated gold electrodes. Then, the PCA3 ssDNA probe equipped with incorporated gold nanoparticles was investigated by electrochemical impedance spectroscopy (using a PGSTAT 204, Autolab system) and UV–vis spectroscopy (using a Hitachi U-2001 spectrophotometer). The spectroscopy measurements were mapped into 2D spectrum plots by applying the interactive document mapping technique while preserving the similarity of the spectral objects in projected space. Next, a digital scanning microscope was used and a dataset for the ML training was acquired. As the training dataset contained a relatively small number of samples, the data augmentation procedure was implemented, and an eight-class classification task was formulated. Before the classification, texture feature extraction was implemented with the following techniques: Gray Level Difference Matrix (GLDM), Fourier descriptors, Complex Network Texture Descriptor (CNTD), Fractal descriptors, Adaptative Hybrid Pattern (AHP), Local Binary Patterns (LBP), Complex Network, Randomized Neural Network (CNRNN), and Local Complex Features and Neural Network (LCFNN). Three types of classifiers were used: SVM with linear kernel; Linear Discriminant Analysis (LDA); and 1-Nearest Neighborhood (1-NN). The authors reported that the maximum accuracy was 99.9 (0.3) using the LCFNN descriptor with SVM and LDA classifiers in the binary classification.

Genomic lesions involve in a variety of morphologic abnormalities, and morphologic evaluation of blood and marrow cells serves as an essential and general approach to studying myelodysplastic syndrome pathogenesis. The diagnosis of myelodysplasia (MDS) is rather difficult due to the large variety of chromosome lesions, diversity of epigenetic alterations, and complexity of processes when the blood-forming cells in the bone marrow experience sufficient transformations and become abnormal. Improving diagnosis and prognosis of MDS can be achieved with modern AI/ML models and statistical and bioinformatics tools applied to large genomic and morphologic data, according to Y. Nagata et al. in [15]. They analyzed bone marrow morphologic alterations in 1079 MDS patients and identified a total of 1929 somatic mutations with whole-exome sequencing (Illumina HiSeq 2000 device). Sequencing errors were reduced and targeted sequencing was performed. To improve data quality, the authors used ANNOVAR sample annotation and compared them with sequenced controls and mutation databases (including dbSNP138, 1000 Genomes, ESP 6500, and Exome Aggregation Consortium (ExAC) databases), in addition to filtering. Using the Consensus Cluster Plus and radomForestSRC packages in R, the authors deter-

mined that the most frequently mutated genes were TET2 (20%), ASXL1 (17%), SF3B1 (13%), SRSF2 (11%), DNMT3A (11%), and RUNX1 (10%). Based on this result and the Bayesian ML procedure, there were eight genetic signatures identified: a DT of eight subtypes for the LR MDS patients (LR-SA through LR-SH). The total accuracy of this approach for the detection of somatic mutations is up to 98.7%.

With the advent of the Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR)-associated protein 9 (Cas9), it has become possible to use computer technologies to edit genes for further use in hematology and oncology. In [16], Han Zhang and Nami McCarthy describe the prospects of using the CRISPR system for genome editing to treat hematological diseases such as lymphoma, myeloma, leukemia, etc. The conducted studies have shown that the use of the CRISPR system makes it possible to identify the potential therapeutic effects of using various drugs in the treatment of lymphomas. Thus, in [17], a target gene for the therapeutic activity of NUTLIN3 was discovered. The use of CRISPR in the suppression of the oncogenic transmembrane protein MUC1-C made it easier to create knockout agents for the treatment of multiple myeloma (MM). However, there is a lot of clinical practice ahead to prove the effectiveness of this method. Therefore, aa team of scientists at Sichuan University in Chengdu working under the guidance of the oncologist Lu You [18] received permission from the ethics commission, and in 2016, for the first time, injected human cells whose genes were modified using the CRISPR/Cas9 technology. The rapidly increasing popularity of this method for the treatment of cancer provides a great number of ways to integrate CRISPR and ML to identify the CRISPR arrays using ML algorithms. In [19], the authors proposed an approach for separating the true CRISPR data arrays from false ones based on the developed CRISPRidentify tool. The developed pipeline consists of two stages: the detection and generation of candidate arrays and universal evaluation of arrays to increase sensitivity and specificity. A. Mitrofanov et al. in [19] identified new training sets, based on which they then trained a classifier to evaluate candidate arrays. Six different datasets were used. A feature vector was generated for each candidate array, which was used to evaluate the pipeline. The Extra Trees classifier from the Python Scikit-learn package https://scikit-learn.org/stable/ (accessed on 12 December 2023), which was integrated into the pipeline, was used to conduct the assessment. As a result of a number of tests, the Extra Trees method surpassed the classifiers based on NB, SVM, k-NN, etc., with a median accuracy of 0.91. In the conducted test, CRISPRidentify was able to detect 147 previously annotated arrays. The developed tool is able to detect the arrays with a limited number of repeating separators, as well as to create dynamic filtering criteria based on the data received during training.

The molecular genetic somatic mutation research in cancer plays an essential role and the presence of mutated genes that belong to the Ras subfamily (e.g., KRAS, NRAS, HRAS) allows for the detection of several cancer types. The Ras subfamily genes can normally exist in the inactive form and encode a cytoplasmic protein involved in intracellular signaling from growth factor receptors. For example, in the vast majority of cases, constant activation of KRAS and NRAS genes leads to malignant mutation of the cells and activates the EGFR-RAS-RAF pathway. This pathway signals tumors in such primary cases as pancreatic adenocarcinoma, lung adenocarcinoma, thyroid carcinoma, and skin cutaneous melanoma. In [20], Way et al. proposed a method of classification and prognosis of Ras activation in cancer using ML and the Pan-Cancer Atlas data. At that time, the data in The Cancer Genome Atlas (TCGA) Pan-Cancer Atlas included 9075 tumors across 33 different cancer types, but now the number of tumor cases included in the Atlas is over 11,000. The data for the ML model training were collected from three sources: the GISTIC2.0 source, cancer cell lines, and Illumina RNA sequencing. The ML model was trained on those data after preprocessing and filtering. The model is elastic net LR with penalties. The elastic net is a regularized regression algorithm that combines both the LASSO and Ridge techniques to improve regularization through removing weak variables altogether, as with LASSO, or reducing them closer to zero as with Ridge. The authors reported high performance of their ML model, with 84% AUC, in predicting Ras pathway activation metrics.

Recent years have witnessed the success of CNNs, driven by the development of DL, dramatically increased computing power of GPUs, and the emergence of large volumes of labeled datasets in visual recognition problems. As a result, DL and CNN have become increasingly popular in CV in various fields [21], including diagnostics and medicine. The most common examples include classification of skin lesions [22], histological classification of breast cancer [23], and mortality assessment by chest radiographs (lung cancer). ML is increasingly used to solve the problems of segmentation and classification of cancer cells in various locations in the body; new systems are being developed and improved that can detect the presence of pathological cells and classify them using a loaded image of scanned glass with histology [24–28].

One of the examples of using the ML models to solve the problem of skin cancer recognition was the development of the "SkinVision" medical service https://www.skinvision.com (accessed on 5 February 2024). The developed application combines AI technologies and the experience of highly qualified oncologists in order to provide recommendations for the prevention of skin cancer and timely detection of diseases. Among the assessed signs are the shape, texture, and color of skin formations. It has been proven that the use of the algorithm in 95% of cases makes it possible to detect the disease at an early stage, but the control of a human specialist is still a prerequisite.

An important application of the ML methods is the study of attenuation correction and image restoration in PET. In [29], Tonghe Wang et al. confirm that in PET, an accurate quantitative assessment of the absorption of indicators is of great clinical interest. Incorrect solution of the problems of inaccurate attenuation and scattering of photons can lead to a deterioration in the accuracy of the PET diagnostics. The authors claim that in recent years, sharing Magnetic Resonance (MR) with PET has become a promising alternative to PET/CT due to the excellent anatomical visualization of soft tissues without ionizing radiation. The automatic classification and manually drawn contour methods are limited by the inaccurate prediction of air and bone. These methods have been replaced by other techniques, such as warped atlases of the MR images labeled with known attenuation factors to the patient-specific MR images. Their effectiveness strongly depends on the registration algorithms. The authors emphasized that image post-preprocessing and the use of noise regularization during reconstruction allowed them to compensate for the differences between neighboring pixels to reduce the noise level in the image. The study suggests two solutions, depending on whether only a PET scanner was used or PET/MR images were obtained. In the second case, the authors analyze the advantages of using the convolutional auto-encoder (CAE) to generate CT tissue labels from the source images. In addition, an RF method is considered for training a set of decision trees. These methods have proven their effectiveness. The MR images are not available in the PET scanner. The most common networks used for processing PET data are GAN and U-Net. The authors consider using the cycle-constant GAN (CycleGAN) method, which showed excellent performance in most of the evaluated indicators and less bias in defeats. In addition, high-resolution PET can be used to visualize and accurately measure the concentration of a radioactive indicator in small structures.

Endometrial cancer (EC) is the most common gynecologic cancer in industrialized countries. About 15% of patients develop recurrence with limited treatment options and poor survival. To improve patient survival and to choose between the most invasive surgical procedures or drug therapy, the reliable identification of patients from different risk groups is required. To minimize the risks of cancer recurrence, new methods of image analysis are needed.

In [30], E. Hodneland reviewed a fully automated approach to primary tumor segmentation in endometrial cancer using three-dimensional CNNs. With this approach, the estimates of tumor volume and the accuracy of CNN-based segmentation are comparable to the segmentation results obtained by radiologists. Automated and accurate CNN tumor segmentation provides new opportunities for full-scale operational radioma tumor profiling, allowing for prognostic markers, which makes oncology treatment more personalized for each patient.

Preoperative MRI in patients with endometrial cancer provides information about the spread of the tumor, which is necessary for the choice of a surgical procedure and drug therapy. In addition, the full range of tumor MRI can provide the radioma signatures of the tumor, which are important for the selection of the optimal treatment regimen for a particular patient. The sample for the CNN training consisted of 139 patients, while expert physicians significantly disagreed on the placement of the primary tumor in 14% of cases. CNN allows for the automatic extraction of tumor volume and tumor texture characteristics, which makes it possible to obtain better prognosis and individualization of therapeutic strategies for endometrial cancer. The results obtained were comparable to the human expert level, which was confirmed during the Wilcoxon signed rank test. Manual segmentation of the tumor to calculate the volume and study the textural characteristics of the tumor is a time-consuming process. E. Hodneland et al. demonstrate the use of two-dimensional U-Net convolutional networks to solve the segmentation tumor issues. There are many powerful frameworks and platforms that allow specialists to implement DL in 3D medical imaging. The automatic segmentation method has the potential to detect systematic changes in tumor volume in response to various treatment strategies. Such potential has high value in clinical drug trials. There were no significant differences in the conducted Intraclass Correlation (ICC) estimates. This suggests that the network performs functions at a level comparable to radiologists.

CV development in medicine has recently opened up many new avenues for solving problems in the field of cardiovascular diseases, where echocardiography is uniquely suited to DL as a simple and effective method for collecting and interpreting data [31]. DL helps pathologists to conduct histological analysis of tissue samples, which is subjective in nature, and different perceptions of pathologists can lead to misdiagnoses [32]. The latest advances in technology have made it possible to create new scanners that capture histological specimens with high resolution and obtain WSIs [33]. This, combined with advances in AI, has led to the research and commercialization of AI-based digital histopathology [34]. James T. Grist et al. describe the methods of RF, l-NN, and AdaBoost, which were used to create the model [35]. The univariate classifier obtained 85% accuracy with AdaBoost. Using the SVM method had the highest precision in discriminating between high- and low-grade tumors. The study showed that the combination of the multiparameter MRI, the ML methods, as well as single-factor analysis can be used to distinguish both highly differentiated and low-differentiated brain tumors in children, as well as to classify tumor types with a high accuracy of about 85%.

The study [33] by Evans et al. provides examples of medical data visualization in cardiology, dermatology, and pathology. The authors note that CV enables solving medical problems such as screening, segmentation of pathologies, predicting outcomes, as well as disease monitoring. The most important DL technique used to solve these problems is CNN. It hardcodes the key feature of images—invariance. The study of medical data is associated with specific problems. Various methods are used to solve them. The authors describe the use of supervised learning, which uses data points and data labels.

Glioma is one of the most common primary brain tumors. ML is used to visualize gliomas and can act as a non-invasive assessment of the molecular characteristics of gliomas. After such an assessment, the data are sent to the laboratory for further clinical trials. It is also noted that the segmentation of tumor volumes in PET can be automated by implementing CNN. Specialists face difficulties in diagnosing children's brain cancer, because the MR scanning method is ineffective in the brain areas with poor magnetic field homogeneity and small lesions. T.C. Booth et al. [36] discuss the methods of diffusion-weighted imaging (DWI) and dynamic susceptibility contrast imaging (DSC), which are used in children brain tumor diagnostics. DWI creates images weighted by the water velocity in a given voxel. DSC displays the dynamics of contrast media using rapid imaging techniques (EPI and PRESTO). Supervised machine learning utilizes mean apparent diffusion coefficient (ADC) or mean cerebral blood volume (CBV) data features and high and low grade of tumor types as classes for mathematical algorithm training and assigning datasets to classes.

The applications of supervised learning to oncological medical imaging have commonly utilized single measures of the tumor microenvironment between tumor types. F-statistics were used to evaluate the discriminant ability of classifiers. The authors in [36] conducted controlled ML using the Orange toolbox in Python (version 3.6) and a single NN.

The Wisconsin diagnostic breast cancer dataset (WDBC) has remained in demand for many years. It consists of a total of 569 records classified into two categories: malignant (212) and benign (357) tumor cell cases. A detailed description of WDCB that contains 10 essential real-valued features (radius, texture, perimeter, etc.) computed for each cell nucleus makes this dataset an excellent candidate for ML model training and estimation. P. Gupta and S. Garg, in their study [37], applied six different models to WDBC, including k-NN, LR, DT, RF, SVM, and deep learning ANN (DL_ANN). They estimated statistical metrics for the above-mentioned models and reported on the following results. All six ML models trained on WDBC had high accuracy of over 95%, but in this case, the preferred ML model was the DL_ANN model, having an accuracy of 98.24%, and the precision, recall, and F1-score for this model were over 98%, too.

R.T. Lewinson et al. developed an ML neural network prediction model for the prediction of cutaneous immune-related adverse events (irAEs) from anti-PD-1 therapy [38]. This model was trained on the dataset of 143 metastatic melanoma and 339 non-small cell lung cancer patients treated with anti-PD-1 therapy. The authors used a three-layer fully connected feed-forward neural network, then trained it with 85% of the collected dataset and tested it with the remaining 15% of the dataset. The model showed promising predictive results, including an AUC of 76.5% and overall accuracy of 78.1%. It was concluded that ML models could be applied to the identification of at-risk patients for early dermatologic interventions.

A.M. Bur et al. [39] proposed models for predicting pathological nodal metastasis in oral cavity squamous cell carcinoma. The datasets for the ML models were extracted from the National Cancer Database, totaling 1961 cases with confirmed tumors and 71 patients treated at a single institute. The former dataset was split into two parts of 1570 training cases and 391 testing cases, respectively. The latter dataset was split into a {51:49} percentage proportion. The authors developed several ML models for processing those datasets to predict pathologic lymph node metastasis. The following ML models were implemented: LR, DF, kernel SVM, and Gradient Boosting Machine (GBM) algorithms. The best result for the NCDB dataset was the DF classification model, with an AUC of 0.712 (sensitivity of 0.753 and specificity of 0.492), and the best result for the SI dataset was the DF classification model, with an AUC of 0.840 (sensitivity of 0.917 and specificity of 0.576).

R. L. Gullo et al. [40] provided an overview of the early prediction of response to NAC with an ML model. One can consider NAC as a potential standard that is widely used to treat several subtypes of breast cancer. Generally, early identification and prediction of breast cancer demonstrate decreasing patient mortality, and MRI is the most applied technique for this purpose. Owing to a great number of such features as texture, shape, color gradings, and others, the MRI images could be one of the essential candidates for the so-called radiomics signature. Along with proper labeling of MRI images in the radiomics signature, ML approaches could be applied to the classification and prediction of tumor changes and patients' health parameters. The authors considered specific ML classifiers for the prediction of residual cancer burden, recurrence-free survival, and disease-specific survival, namely linear SVM, LDA, LR, RF, SGD, DT, AdaBoost, and extreme gradient boosting (XGBoost). The variables between integrated clinical and MRI data frequently have latent and non-linear relationships. Thus, the ML algorithms are supplemented by statistical methods, for example, Bayesian LR. When clinical and MRI features are combined, they demonstrated a better accuracy of 0.86 (95% CI 0.71–0.96) with a sensitivity of 0.88 (95% CI 0.71–1) and a specificity of 0.82 (95% CI 0.56–1). Predictive response to NAC is usually improved when one of the DL techniques is applied. The most used DL architecture is CNN consisting of convolutional, fully connected, dropout, max pooling,

and classifier layers. The accuracy of the three-class prediction of NAC can reach up to 87.7% ± 0.6% with a specificity of 95.1% ± 3.1%, and a sensitivity of 73.9% ± 4.5%.

The balance between prognostic biomarkers of adenocarcinoma and squamous cell carcinoma (SqCC) is not in favor of the latter, since SqCC biomarkers have been studied less intensively. In the study by Y. Koike et al. [41], a total of 135 peripheral SqCC confirmed cases were analyzed to develop ML models for classifying and determining tumor, stroma, and necrosis areas. They used an ML approach to the prognosis of the stroma component significance in the histomorphometric analysis of patients suffering from lung SqCC. The dataset was prepared by converting ematoxylineosin tissue glass slides into high-resolution digital data using the Hamamatsu Photonics NanoZoomer 2.0 system, and then the images were processed with color-based segmentation through kM clustering implemented by MATLAB. As a result, the described approach provided the exact calculation area of the cancer cell component, the cancer cell necrosis component, and the stromal cell component. The authors reported that the ML approach confirmed the prognostic significance of the ratio of stromal cells in the prognosis of lung SqCC.

PET coupled with CT can serve as an integrated tool providing holistic quantification of tumor metabolic activity with tissue density and structural property description. The resulting image radiomics biomarkers and volumes of interest of images are essential quantitative predictors for high-risk human papillomavirus (HPV) infection and were studied by S.P. Haider in [42]. A complete set of PET/CT features accounting for 1317 PET and 1317 CT features per lesion of a total of 190 patients with HPV-associated oropharyngeal squamous cell carcinomas was screened. The authors developed the locoregional progression model as the ML method and utilized the combined set of radiomics and PET/CT features as a predictor for prognosis. The study reported that the developed ML predictor had median (interquartile range) C-index of 0.76 (0.66–0.81; $p = 0.01$).

3. Treatment Planning

Cancer treatment planning is a difficult issue. The treatment for various types of cancer is not only specific, but often has a number of side effects. In addition, assistance in the treatment of cancer requires a multidisciplinary approach and the involvement of various specialists: surgeons and radiologists, persons caring for patients during their rehabilitation, etc. Moreover, the emergence of new scientific discoveries in the field of oncology research leads to a change in existing paradigms of disease treatment. Finally, the types of cancer treatment assistance are divided into initial, curative, supportive care, and help with cancer relapses, as shown in Figure 2. All of the above points create significant difficulties in using AI for cancer treatment.

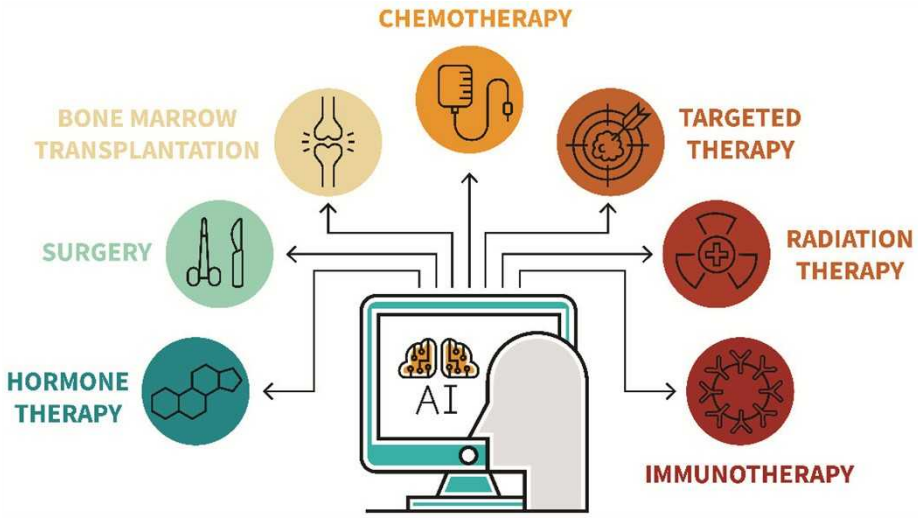


Figure 2. Treatment planning.

On the other hand, the application of ML in healthcare is proving its worth and validity every day. Constantly growing volumes of data, reducing resource costs, improving the quality of prescribed treatment and patient care, timely diagnosis of diseases, and other advantages allow one to gain a positive effect from the introduction of ML and Big Data analysis as essential technologies for helping medical specialists. However, such integrations require highly skilled computer scientists, making it difficult to deploy such support systems everywhere. Moreover, healthcare professionals often lack ML skills. J. Waring et al., in their review paper [43], consider automated ML (AutoML) and a comparative analysis of existing solutions and platforms. The overview is based on 101 sources in AutoML and highlights the major limitation of this technology; it is not efficient enough to work with large-scale data. Both individual elements of AutoML optimization and solutions that allow tracing the entire cycle are considered. The pivot tables of automated function design tools, the AutoML Pipeline optimizers, are presented, and the performance of Neural Architecture Search (NAS) algorithms based on CIFAR- 10 is evaluated.

The use of AI in precision oncology allows for the analysis of tumor genomic data, providing information about potentially effective options in next-generation tumor sequencing assays. It has become possible to predict the type of tumor based on the targeted panel of DNA sequence data. This, in turn, allows specialists to optimize the patient's treatment strategy.

Genomic screening makes it possible to more correctly organize the treatment of patients based on their genetic factors. The investigation of individual genetic factors is enabled by the advent of next-generation sequencing (NGS) technology. A detailed review of NGS and the possibility of its integration with Big Data technologies, ML, and visualization of medical diagnostic data has been conducted in [44]. Timely qualitative analysis of large data sequencing provides early diagnosis of diseases. This is enabled by the accurate forecast model, the Rowan biomarkers, and the identifiers for drug discovery, which specify the strategy for effective cancer treatment for each patient individually. The authors present an overview of the existing NGS platforms with the estimation of the output data amount and the cost of the study, which makes it possible to reflect the trend in the development of precision medicine methods in the field of DNA and RNA sequencing. One of the important components of their research is the expression of medical imaging as a useful tool for detecting, monitoring, and predicting the development of cancer. They conclude that the integration of existing strategies with AI algorithms could significantly expand the capabilities of healthcare.

The paper in [45] discusses the possibility of using ML and DL in matters modulating pharmacological agents to treat cancerous diseases. This article reveals the current potential in solving the problems of developing new and using existing drugs. The importance of in silico structural methods for expanding the possibilities for the synthesis of new drugs is emphasized, among which molecular docking stands out, predicting and assessing priorities in the interaction of a target and a drug. This method is integrated with ML and DL technologies and can significantly improve the accuracy of molecular docking. In addition, the treated and other structural virtual screening techniques have been processed, such as cheminformatics modeling-based ligands using machine learning and proteochemometrics simulation with the use of ML. N.T. Issa et al. analyzed ML capabilities to predict drug–phenotype associations and indicated the applications of large-scale transcriptomics datasets. Based on current advances in natural language processing, Electronic Health Record (EHR) datasets have been evaluated to predict the potential side effects of medication. The paper provides recommendations on the use of ML and AI methods for repurposing drugs. In conclusion, the authors noted the possibilities for the integrated use of AI to solve fundamental problems in creating new and using existing drugs for new types of cancer.

Molecular profiling of tumors is an intensively developing approach that allows targetable identification of malignant somatic alterations and gives rise to a set of diverse strategies in precision oncology. Precision oncology uses genome-scale omics data and next-generation sequencing molecular analysis in addition to the data from large-scale pharmacogenetic screening cell lines and patient-derived xenografts to discover molecular targeting drugs affecting targeted mutated genes. Nevertheless, the problem of targeted drug development remains complex and non-trivial. R. Miao et al. [46] proposed a tumor drug and drug sensitivity prediction model based on the precision oncology approach combined with statistical and ML models. Their approach includes the following steps. The first step was collection from gene expression libraries (Genomics of Drug Sensitivity in Cancer Project, Affymetrix CEL files), the Cancer Cell Line Encyclopedia, and the data from the cancer drug-sensitive genomics pharmacogenomics studies. Next, the data were preprocessed to extract genomic state markers as targets for molecule-specific drugs. A total of 140 drug sensitivity experiments were analyzed in 624 cell lines with 71 gene mutation status and 22,215 probe expression values. The second step was preprocessing the dataset by Robust Multi-Array Averaging, converting the data with two logarithmic operations and single peak testing of the selected gene points on the data. This operation showed that the collected data had a bimodal distribution and were statistically significant in representing a large number of gene variations. After this test, the initial 22,215 probes were reduced to 1595 probes with significant multimodal characteristics and selected as the features for predicting the drug sensitivity. Screened potential genes allowed for choosing targeted drugs, and only 10 of the 53 candidates were approved for clinical application in genomic test instructions and mutation detection of BRCA1/2, EGFR, ERBB2, ALK, and BCR-ABL. The authors experimented with five ML models (RF, Elastic Net, Linear-SVM, PLOY-SVM, and RBF-SVM) and reported that the best result was the RF ML model, which had an average sensitivity of 0.98 and an average specificity of 0.97, and an average accuracy of 0.978. The others had no values mentioned exceeding 0.74.

ML techniques can be successfully applied for prostate low-dose-rate brachytherapy planning instead of the manual conventional planning procedures. In the study by A. Nicolae et al. [47], the authors collected a large dataset with high-quality postoperative prostate images at day 30. After feature extraction from the DICOM file format, the new contours allowed for the application of reinforcement learning, and the initial model was adjusted to improve its performance and accuracy. The developed prostate implant planning algorithm was tested on 41 consecutive patients who underwent I-125 LDR monotherapy and showed high sensitivity to excluding overdose or underdose therapy.

Radiation oncology supplies techniques that must be carefully planned to avoid side effects for patients. Stereotactic body radiation therapy (SBRT) is among them. The treatment planning should be fractioned, dosed, and assessed through a patient's full dose–volume histogram (DVH) profile. X. Pan et al. [48] applied advanced ML techniques for planning prostate SBRT with reduced treatment-related toxicity. They involved 86 patients who underwent prostate SBRT, utilized ensemble ML algorithms, and verified selected models on 26 external cases as the test models. The authors used boosting (Gradient Boosted DT and Adaptive AdaBoost) and RF ensemble learning technologies. The investigation of the three abovementioned ML models showed that the Gradient Boosted DT outperformed the others for each of the urinary and rectal domains at 3 and 12 months after SBRT. The authors reported that the best AUC results were for the gradient boosting decision tree (GDBT), with a cumulative DVH (urinary irritation) of 0.79, cumulative DVH (urinary incontinence) of 0.87, and rectal toxicity of 0.64.

Z. Yang et al. [49] developed a DL framework and complementary statistical methods to predict the relationship between prostate radiation therapy and patients' quality of life. The life quality data were collected after prostate cancer radiation therapy with a 14-question survey with added computed tomography images in de-identified form for a group of 52 patients. Then the dataset was augmented with the curvature-based image registration technique implemented in MATLAB and amounted to 1326 images. The

image regions that had high radiation dosage were marked and outlined. The authors developed a deep NN with three-convolution-layer architecture, exponential linear units, and SoftMax classification layer to classify images of patients with or without symptoms of high radiation dosage. Due to relatively small training (39 cases) and testing (13 cases) datasets, overfitting was noticed. Thus, transfer learning was implemented to improve classification performance with autoencoder implementation to pre-train the CNN on the augmented unlabeled data. The CNN results were compared with statistical methods, including *t*-test, ANOVA, and LR. They reported that the best accuracy of the developed prediction model was 84% for the rectum model and in the range of 23–53% with a median accuracy of 38%.

An ML-based approach for predictive values of gene expression profiles in formalin-fixed paraffin-embedded tumor biopsies was proposed by M. Wiesweg et al. in [50]. The authors performed digital expression gene analysis on 30 potential reference genes using the NanoString nCounter platform and the PanCancer Immune Profiling panel consisting of 770 genes involved in an immune response. The authors performed digital expression gene analysis on potential reference genes (CYLD, CXCL11, MYD88, PRPF38A, HLA-E, STAT6, CD59, and STAT2) using the NanoString nCounter platform and the PanCancer Immune Profiling panel consisting of 770 genes involved in an immune response. A total of 127 patients were screened, and then 55 generated admissible nCounter reads were collected in the training cohort. This dataset underwent RNA feature selection using an ensemble approach with random subsampling and regularized with LASSO/L1 regression. The following ML models were implemented in the Python language: SVM (scikit-learn SVC for classification and SVR for regression), RF (scikit-learn RandomForestClassifier and RandomForestRegressor), LR (scikit-learn LogisticRegression for classification and ElasticNet for regression), and GB (xgboost XGBClassifier and XGBRegressor). It was concluded that ML models could be useful for predicting the positive identification of a group of patients benefitting the immunotherapy.

4. Patient Survival Prognosis

Modern technological advances in computer science include artificial intelligence and promote the use of ML algorithms in medical diagnostic and research, which is a strategic direction in the development of decision support systems in oncology, as shown in Figure 3.

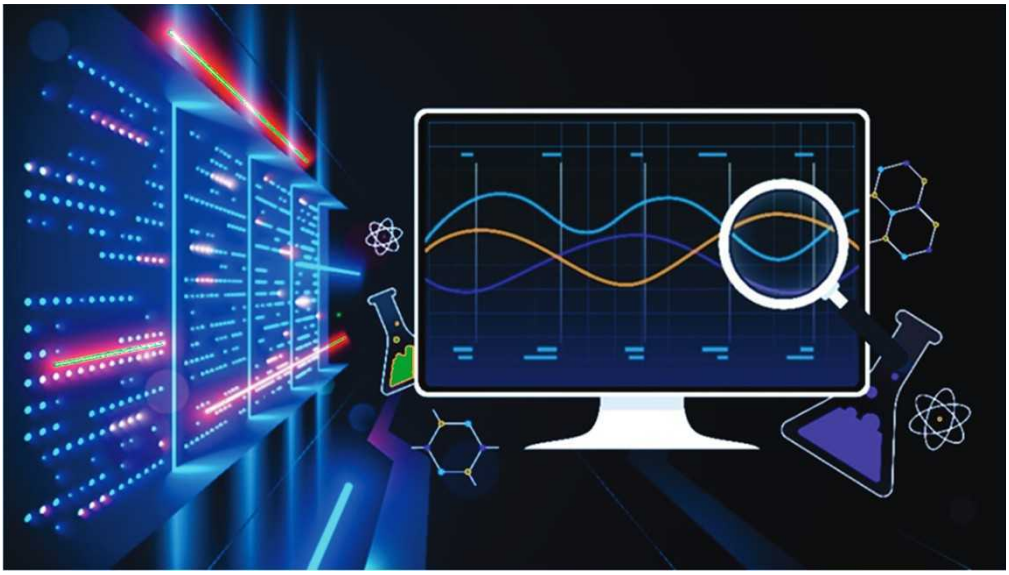


Figure 3. Patient survival prognosis.

This integration can help detect and classify pathologies, create or modify diagnostic test descriptions, and predict disease progression and patient response to therapy. Nev-

ertheless, it is fraught with several problems and limitations in the actual application of the technology in a clinical setting. For a qualitative assessment of possible applications in the study of R. Cuocolo et al. [51], a review of modern literature was carried out, including 90 relevant sources. The summary tables of the ML algorithms are presented. They are used to assess the risks of oncological disease detection, and create descriptions of pathologies, as well as classify and predict types and stages of diseases, indicating a specific goal and type of oncological pathology. All indicated data were accompanied by reduced accuracy, specificity, and sensitivity of the models used. For each of the cases, the limitations for real application were given, and the statistics of the application of ML methods on the training and test dataset were shown, indicating the quantitative and qualitative characteristics of the latter. The possibility of constructing predictive models of patient survival based on ML methods was assessed, as well as the use of AI to develop targeted methods of treating cancer. The data in [51] have a clear understandable structure, giving comprehensive explanations of the current state of the art on the application of ML in oncology in a clinical setting.

Radiomics can also be used to develop a robust radiomics-based classifier that is capable of accurately predicting the overall survival of renal cell carcinoma patients for the prognosis of clear cell renal cell carcinoma. This signature may aid in the identification of high-risk patients who need additional treatment and follow-up regimens. For this model, the AUC, accuracy, sensitivity, and specificity with a 95 percent confidence interval were 0.95–0.98, 0.93–0.98, 0.93–0.96, and 1.0, respectively [52].

One of the determining factors of surgical intervention and the degree of resection in meningioma is the preoperative determination of its consistency. Thus, the study of S. Cepeda [53] aims to develop a predictive model for determining the parameters of meningioma consistency based on preoperative medical diagnostic studies, including MRI and ultrasound elastography (IOUS-E) using ML classifiers. The data from 18 patients operated on in the period 2018–2020 were selected as the initial dataset. All cases were followed by IOUS-E and T 1 WC MRI diagnostics. Preliminary processing of the diagnostic data consisted of the decomposition of the DICOM electrogram format into hue–saturation–brightness (HSB) with subsequent manual tumor segmentation, as well as the removal of MRI radioma signs using the open-source LifEx version 6.0 software and the subsequent automatic segmentation using ITK-SNAP. The softness meningiomas performed value average tissue elasticity (MTE) was chosen as the reference value. As a predictive model, a DT was used, which determined the threshold value based on the generated groups of intraoperative MTE characteristics and made it possible to determine the consistency of the meningioma, where Range version 3.26 was used to optimize the algorithm. The classification system was based on six different ML algorithms, including LR, NB, kM, RF, SVM, and MLP. The models were evaluated using the AUC and reached values from 0.699 to 0.974, with an accuracy of 61–94% and a precision of 60–95% on each algorithm used. The best results were demonstrated by the combination of the Information Gain and ReliefF filters with the NB model, in which the AUC was 0.961, with an accuracy of 95%, corresponding to a classification accuracy of 94%.

The great advantage of using ML in processing data is its ability to evaluate the impact of possible surgical intervention and potential risks of developing distant metastasis. It can provide biological information that cannot be determined from conventional MRI characteristics. Using a retrospective dataset, researchers [54] investigated an ML model based on radiomics, which is favorable for predicting the likelihood of distant metastasis from soft-tissue sarcoma. It was shown that the AUC in the validation set of patients with soft-tumor sarcoma with distant metastasis may be 0.9510. That means that the probability that the mechanism will rank an instance as a true positive is higher than the false positive rate. This may help with clinical decision making to lower the burden of costs from follow-up examinations and anxiety related to false positive results by understanding the most common organs of distant metastasis.

Radiomics has emerged as a promising clinical tool for a variety of clinical problems, including drug development, clinical diagnosis, treatment selection and implementation, and prognosis. Nowadays, radiomics is primarily used to calculate progression-free survival or overall survival, predict tumor sensitivity to treatment, and perform classification tasks that do not fall into one of the above categories. These techniques have the potential to improve patient care in future AI-based radiotherapy workflows by developing personalized medicine and prescribing personalized doses [55].

Radiation therapy coupled with chemotherapy is frequently complemented by radical surgery in gastric cancer treatment. Nevertheless, the question of the toxicity and dosage of radiation therapy constantly arises. In this regard, there is an increasing interest in the use of AI technologies and ML for the prediction and evaluation of the overall survival of patients who undergo radical surgery, radiation therapy, and chemotherapy. M. Akcay et al. [56] assessed the applicability of several ML algorithms to predict the overall survivability of patients with confirmed gastric cancer, peritoneal recurrence patterns, and distant metastasis after treatment. The authors involved several ML algorithms, including LR, MLP, XGBoost, SVM, RF, and Gaussian Naive Bayes (GNB). The dataset was collected from the data of 75 cases of patients with gastric cancer undergoing radiation therapy and chemotherapy at the Department of Radiation Oncology of Eskisehir Osmangazi University Faculty of Medicine (Turkey). The authors avoided an imbalance in the collected dataset and applied the synthetic minority resampling (SMOTE) technique. The ML algorithms and statistical functions were implemented using the Python and Scikit-Learn library. The authors reported that the best prediction performances of overall survivability, distant metastases, and peritoneal metastases were obtained with GNB, with an accuracy of 81% (95% CI, 0.65–0.97, AUC = 0.89), XGBoost, with an accuracy of 86% (95% CI, 0.74–0.97, AUC = 0.86), and RF, with an accuracy of 97% (95% CI, 0.92–1.00, AUC = 0.97), respectively.

ML-based MRI radiomics can be used to improve the accuracy of a prediction model. Using various datasets, the researchers tested a model that was aimed at distinguishing aggressive from non-aggressive papillary thyroid carcinoma. In the testing cohort, a prediction model using only clinical characteristics achieved an AUC of 0.56, which is low and close to a random result. The AUC of the ML-based MRI radiomics was 0.92, indicating improved performance over the clinical characteristics model. A new model that combined clinical characteristics and radiomic features was similar to the model that only used radiomic features. This finding is promising for a non-invasive method of assessing cancer aggressiveness. Assessing the aggressiveness of papillary thyroid carcinoma before surgery may assist clinicians in planning a surgical approach, such as lobectomy, subtotal thyroidectomy, or total thyroidectomy, with or without neck dissection [57].

The main strategy for using ML to solve the problems of radiation oncology in most works published on this topic is to simplify and reduce the problems to the form of a binary classification, which can cause unreliable research results. In addition, there are several problems that can be solved in terms of multiclass classification. In the study by A. Chatterjee et al. [58], the authors provided the facts that enabled them to evaluate the results for several studied classes reliably and the possibility of using surrogate markers in a clinical setting. The authors reported that, unlike binary classification, the AUC and the correlation coefficient could not serve as reliable indicators of performance for multiclass classification in a clinical setting. The validity of their arguments was confirmed using the Monte Carlo model (MC), which showed that the capacity prediction characteristics decreased with the amounts and classes of outcomes. In the second part of the study, devoted to the use of surrogate markers in real conditions, the MC analysis was performed by calculating five scale factors and adding noise to the surrogate marker for each patient from five cohorts with different prescribed treatments. The results were obtained with the statistical differences, separating groups of patients when using a surrogate marker according to the criterion rank-sum Wilcoxon ($p < 0.001$).

Feature-based and DL-based radiomics is being increasingly evaluated in neuro-oncology [1]. AI can be useful for the differentiation of treatment-related changes from

tumor progression in patients with gliomas and brain metastases (AUC of 0.73–0.96), which is a critical clinical question. The addition of molecular markers to histology is recommended to define brain tumor entities but it damages the tissue, and the core biopsy specimen cannot be taken very often. A noninvasive method is a good solution when tissue samples are not accessible. It is possible to predict the isocitrate dehydrogenase (IDH) genotype (AUC of 80–96), loss of heterozygosity of the 1p/19q chromosome arms (AUC of 67–96), and the O6-methylguanine-DNA methyltransferase (MGMT) promoter methylation status (AUC of 80–95). It is molecular markers on which treatment strategies are based. Furthermore, it was demonstrated that the expression levels of the tumor cell proliferation marker (Ki-67) can be used to calculate or predict proliferative activity in gliomas (AUC of 76–92). Aside from that, several studies since 2016 have evaluated radiomics for the determination of grades from the World Health Organization (WHO) classification of central nervous system tumors (AUC of 87–97).

Methylation of the MGMT promoter could also be used to predict overall survival. It has been shown that the age and MGMT methylation index are the most important factors for predicting overall survival in the chosen models. Radiomic features provide additional information beyond the scope of human visual perception, which is considered to be valuable in the prediction of glioblastoma OS and progression-free survival, and there is relatively limited literature on this topic [59].

One significant advantage of in silico studies is that researchers have no restrictions on the parameters or features they can use to find correlations. A radiomics study identified optimal ML methods for the radiomics-based prediction of local failure and distant failure in advanced nasopharyngeal carcinoma [60]. They tested different parameters and features in combination to divide patients into three groups: those who experienced local failure, those who experienced distant failure, and those who did not experience either local or distant failure during the follow-up period. It helped to identify new parameters that are representative enough in this issue to become new imaging biomarkers to assist clinical decision making. Quantitative radiomics has the potential to significantly improve cancer care by facilitating the implementation of personalized medicine.

Neuro-oncology is gradually expanding its field of knowledge using ML models. C.A. Sarkiss and I.M. Germano presented a systematic review [61] devoted to using ML-based algorithms in neuro-oncology. They extracted some information about ML application from 29 studies and the data on a total of 5346 patients who were screened for neuro-oncology diseases, such as glioma, glioblastoma, brain metastases, etc. The studies mentioned above reported on a set of ML algorithms including SVM, RF, Gaussian NB, LDA, DT, gradient boosting (GB), and adaptive GB. Then, the ML algorithms were clustered by the task and application into three categories: outcome predictors, image analysis, and gene expression. The authors reported that the outcome prediction was used in 12 of 29 studies with a sensitivity range of 78–98% and specificity range of 76–95%. The image analysis was conducted in four studies. Finally, the remaining studies were devoted to ML-based gene expression analysis and had an accuracy ranging from 80% to 93%. The study in [62] examined automatic contour scoring using fit indices and supervised machine learning. The issue is relevant because the peer review of the target volume (TV) and organ at risk (OAR) contours during radiation therapy planning is conducted visually and can be time-consuming, and the result can vary greatly among observers. The best result was obtained for stomach images with an overall prediction accuracy of 96% (using the Vector Machine and Ensemble models); the worst result was shown by LR—76% for liver images. The considered method of automatic contour estimation makes it possible to reduce the review time, eliminate gross errors in contour determination, and can also be used for training doctors.

R.O. Alabi et al. [63] compared the nomogram technique with several ML algorithms to estimate the performance of such approaches in the prediction of tongue cancer survival outcomes. They obtained the dataset from the National Cancer Institute through the Surveillance, Epidemiology, and End Results Program of the National Institutes of Health,

including 7649 cases where patients were diagnosed with positive histologically confirmed tongue cancer. Then the dataset was used for labeling and predicting nomogram building. The nomogram accuracy with surgical treatment was 66% and the nomogram accuracy with radiotherapy was 60.4%. The sensitivity and specificity were 1 and 0.62 for the first case and 1 and 0.55 accordingly for the second case. The authors trained seven ML models (LR, NB, SVM, ANN, Boosted Decision Tree, Decision Forest, Decision Jungle) in Microsoft Azure ML Studio 2019 to predict an overall survival month using 5-fold cross-validation, splitting the dataset in {80:20} percentage proportion for training and validation. The best accuracy of 83.1% was achieved with the ML model of Boosted Decision Tree, with an AUC of 0.9. The authors reported that the comparative performance metrics of the ML algorithms were higher than the nomogram approach, and the accuracy, sensitivity, specificity, and F1-score were 88.7%, 1, 0.87, and 0.66, respectively.

In [64], A.V. Karhade et al. demonstrated the applicability of ML techniques to prognosticating survivability with confirmed rare tumors. One of them is spinal chordoma, and due to the rareness and relatively slow change of this tumor type, long-term prediction is not a trivial task. The authors obtained statistics for a long period of time (over 15 years), and estimated survivability for 265 patients, showing a mean value for 5-year survival of 67.5%. Then, the conventional statistics operations were compared with four ML models, including BDT, SVM, Bayes Point Machine, and NN. These algorithms were trained, assessed by a 10-fold cross-validation technique, and a web application using Microsoft Azure (Microsoft Corporation, Redmond, WA, USA), R Studio (version 1.0.153), and Anaconda with the Python distribution (Anaconda, Inc., Austin, TX, USA) was implemented. A.V. Karhade et al. concluded that the Bayes Point Machine showed the best predictive capability, having AUC = 0.801. They also pointed out that the ML models in long-term survival predictions had high potential, but a weak explanatory ability.

It is challenging to avoid subjective risk estimation in patients' mortality from cancers while receiving febrile neutropenia chemotherapy, but the inpatient mortality risk prediction is much harder. X. Du et al. showed in [65] that ML models could provide a non-subjective prognosis in mortality risk prediction. They used a dataset from the HCUP's National Inpatient Sample and Nationwide Inpatient Sample on the patients with diagnosed cancer and picked the data with febrile neutropenia, totaling 126,013 adult admissions. These data were complemented with in-hospital information, and the descriptive statistics were calculated. Then, the authors developed two groups of models: linear models including Ridge LR and linear SVM, and non-linear models including GBT and six-layer NN models. After the model tuning and three-fold cross-validation process, the models demonstrated surprisingly the same AUC, sensitivity, specificity, precision, recall, and F1-score. Those parameters were 0.92, 0.81, 0.88, 0.84, 0.70, and 0.74, respectively. The GBT model was slightly different from the others, but it was no more than one hundredth better.

A prognostic system for endometrial cancer using ML was proposed by A.M. Praiss et al., utilizing the Surveillance, Epidemiology, and End Results (SEER) database and Ensemble Algorithm for Clustering Cancer Data (EACCD) for their research [66]. ECCD made it possible to determine dissimilarities in defining and computing on the SEER data and then applying the hierarchical clustering analysis while visualizing the relationship between survival and prognostic factors. The ML algorithms were trained on data totaling 46,773 patients with endometrial cancer. As a result, this ML model generated a visual prognosis in the dendrogram format classifying patients into several groups according to five-year survival rates. The authors reported that this ML model improved prognostic prediction for patients with endometrial cancer compared to traditional staging algorithms. The scientists used the EACD ML algorithm for endometrial cancer and demonstrated an improved ability to recognize survival compared to the traditional TNM stage. A number of positive changes in prognostic ability were noted based on the classification of existing pathological variables, as well as the inclusion of other factors in the model, such as age and tumor degree.

5. Existing Barriers

Despite the prospects of using ML to solve oncology problems, a number of challenges remain, in particular, the task of data collection and management; the lack of reliable reporting standards; the task of standardizing data; bias inherent in a set of training data; the relative lack of prospective clinical trials. The Existing Minimal Common Oncology Data Elements (mCODE) Initiative is aimed at solving the problem of data collection standardization [67]. To identify signs of patient distress, some institutions are beginning to use a patient-reported outcome measures (PROM) mechanism [68]. Another important problem is the bias of the data on which the models are trained. One of the reasons is the underrepresentation of some population groups (adolescents, women, etc.) during clinical trials, based on which the datasets are formed. The lack of training data for some groups of people can lead to erroneous predictions of the ML model, which is unacceptable for further transfer to clinical practice. It is necessary to ensure a wider inclusion of underrepresented population groups in the training data, which will take a significant amount of time. In addition, the modification of existing solutions will be required, which will mitigate possible systematic errors in datasets [69]. The problem of establishing a standard for reporting on the ongoing research in AI is associated with the high sensitivity of DL methods to source data and difficulties in reproducibility, which may limit the widespread use of AI in oncology. Another problem that has not been fully solved is the insufficient number of studies that compare traditional methods of treating diseases with the methods used with the help of AI. Conducting randomized controlled trials (RCTs) is necessary despite its high cost. Regulatory and legal problems arising with the application of ML methods in medicine have been the subject of discussion for many years. The lack of case law significantly limits the resolution of medical liability issues. In the vast majority of cases, AI implementation in medical practice is used not as a decision-making tool, but as a confirmation of a decision made by a human specialist. Reaching consensus on the regulatory framework for AI-based tools, developing clear standards for evaluating the effectiveness of AI-based disease treatment tools, and conducting new research on the introduction of these tools into medical practice are necessary actions to address the many challenges that limit the potential of using AI in oncology and medicine in general.

6. Conclusions

We analyzed 70 papers describing the use of ML methods for the diagnosis of cancer, prediction of the degree of survival, as well as correction and treatment planning. ML is used to deal with various types of cancer, such as carcinoma, glioma, endometrial cancer, prostate cancer, etc. AI finds its application in such modern areas as genomic screening, precision oncology, medicine personalization, and targeted drug delivery. This shows that the progress in the use of AI in medicine is growing rapidly. Various ML models for solving different challenges were considered, such as DT, k-NN, NB, etc., and the results were analyzed. The peculiarities of using various ML models influencing the choice of a specific algorithm for solving each specific task were emphasized. We discussed the prospects of using ML in oncology as a full-fledged tool for the diagnosis and localization of malignant tumors. However, there are a number of barriers that arise when using AI in medicine, related to local, regulatory, and ethical foundations. In this review, we highlighted the importance of solving these issues.

Author Contributions: Conceptualization: M.A.B. and A.V.S.; project administration: A.V.S. and O.I.K.; funding acquisition: A.V.S.; data curation: E.V.V., V.V.S. and S.A.S.; visualization: M.A.B.; formal analysis: V.V.S.; validation: S.A.S. and A.V.P.; software: A.V.P.; writing—review and editing: E.V.V. All authors discussed the results and contributed to the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: Research was financially supported by the Ministry of Science and Higher Education of the Russian Federation (State assignment in the field of scientific activity, No. FENW-2023-0019).

Institutional Review Board Statement: The review did not require ethical approval.

Informed Consent Statement: This review contains references to publications describing research involving humans. All these publications contain "Informed consent was obtained from all subjects involved in the study". When writing this review, no people were involved in the study.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Lohmann, P.; Galldiks, N.; Kocher, M.; Heinzel, A.; Filss, C.P.; Stegmayr, C.; Mottaghy, F.M.; Fink, G.R.; Jon Shah, N.; Langen, K.J. Radiomics in neuro-oncology: Basics, workflow, and applications. *Methods* **2021**, *188*, 112–121. [CrossRef] [PubMed]
2. Sleeman, W.C., IV; Nalluri, J.; Syed, K.; Ghosh, P.; Krawczyk, B.; Hagan, M.; Palta, J.; Kapoor, R. A Machine Learning method for relabeling arbitrary DICOM structure sets to TG-263 defined labels. *J. Biomed. Inform.* **2020**, *109*, 103527. [CrossRef] [PubMed]
3. Klauschen, F.; Müller, K.R.; Binder, A.; Bockmayr, M.; Hägele, M.; Seegerer, P.; Wienert, S.; Pruneri, G.; de Maria, S.; Badve, S.; et al. Scoring of tumor-infiltrating lymphocytes: From visual estimation to machine learning. *Semin. Cancer Biol.* **2018**, *52*, 151–157. [CrossRef] [PubMed]
4. Kazmierska, J.; Hope, A.; Spezi, E.; Beddar, S.; Nailon, W.H.; Osong, B.; Ankolekar, A.; Choudhury, A.; Dekker, A.; Redalen, K.R.; et al. From multisource data to clinical decision aids in radiation oncology: The need for a clinical data science community. *Radiother. Oncol.* **2020**, *153*, 43–54. [CrossRef] [PubMed]
5. Seifert, R.; Weber, M.; Kocakavuk, E.; Rischpler, C.; Kersting, D. Artificial Intelligence and Machine Learning in Nuclear Medicine: Future Perspectives. *Semin. Nucl. Med.* **2021**, *51*, 170–177. [CrossRef] [PubMed]
6. Deist, T.M.; Dankers, F.J.W.M.; Ojha, P.; Scott Marshall, M.; Janssen, T.; Faivre-Finn, C.; Masciocchi, C.; Valentini, V.; Wang, J.; Chen, J. Distributed learning on 20,000+ lung cancer patients—The Personal Health Train. *Radiother. Oncol.* **2020**, *144*, 189–200. [CrossRef] [PubMed]
7. Bibault, J.-E.; Giraud, P.; Burgun, A. Big Data and machine learning in radiation oncology: State of the art and future prospects. *Cancer Lett.* **2016**, *382*, 110–117. [CrossRef]
8. Fabacher, T.; Godet, J.; Klein, D.; Velten, M.; Jegu, J. Machine learning application for incident prostate adenocarcinomas automatic registration in a French regional cancer registry. *Int. J. Med. Inform.* **2020**, *139*, 104139. [CrossRef] [PubMed]
9. Soldatov, S.A.; Pashkov, D.; Guda, S.; Karnaukhov, N.; Guda, A.; Soldatov, A. Deep Learning Classification of Colorectal Lesions Based on Whole Slide Images. *Algorithms* **2022**, *15*, 398. [CrossRef]
10. Barber, E.L.; Garg, R.; Persenaire, C.; Simon, M. Natural language processing with machine learning to predict outcomes after ovarian cancer surgery. *Gynecol. Oncol.* **2021**, *160*, 182–186. [CrossRef]
11. Bitencourt, A.G.V.; Gibbs, P.; Rossi Saccarelli, C.; Daimiel, I.; Lo Gullo, R.; Fox, M.J.; Thakur, S.; Pinker, K.; Morris, E.A.; Morrow, M.; et al. MRI-based machine learning radiomics can predict HER2 expression level and pathologic response after neoadjuvant therapy in HER2 overexpressing breast cancer. *EBioMedicine* **2020**, *61*, 103042. [CrossRef]
12. Yang, H.-Y.; Wang, Y.-C.; Peng, H.-Y.; Huang, C.-H. Breath biopsy of breast cancer using sensor array signals and machine learning analysis. *Sci. Rep.* **2021**, *11*, 103. [CrossRef] [PubMed]
13. Xie, Y.; Meng, W.-Y.; Li, R.-Z.; Wang, Y.-W.; Qian, X.; Chan, C.; Yu, Z.-F.; Fan, X.-X.; Pan, H.-D.; Xie, C.; et al. Early lung cancer diagnostic biomarker discovery by machine learning methods. *Transl. Oncol.* **2021**, *14*, 100907. [CrossRef] [PubMed]
14. Rodrigues, V.C.; Soares, J.C.; Soares, A.C.; Braz, D.C.; Melendez, M.E.; Ribas, L.C.; Scabini, L.F.S.; Bruno, O.M.; Carvalho, A.L.; Reis, R.M.; et al. Electrochemical and optical detection and machine learning applied to images of genosensors for diagnosis of prostate cancer with the biomarker PCA3. *Talanta* **2021**, *222*, 121444. [CrossRef] [PubMed]
15. Nagata, Y.; Zhao, R.; Awada, H.; Kerr, C.M.; Mirzaev, I.; Kongkiatkamon, S.; Nazha, A.; Makishima, H.; Radivoyevitch, T.; Scott, J.G.; et al. Machine learning demonstrates that somatic mutations imprint invariant morphologic features in myelodysplastic syndromes. *Blood* **2020**, *136*, 2249–2262. [CrossRef] [PubMed]
16. Zhang, H.; McCarty, N. CRISPR -Cas9 technology and its application in haematological disorders. *Br. J. Haematol.* **2016**, *175*, 208–225. [CrossRef] [PubMed]
17. Valente, L.J.; Aubrey, B.J.; Herold, M.J.; Kelly, G.L.; Happo, L.; Scott, C.L.; Newbold, A.; Johnstone, R.W.; Huang, D.C.; Vassilev, L.T.; et al. Therapeutic Response to Non-genotoxic Activation of p53 by Nutlin3a Is Driven by PUMA-Mediated Apoptosis in Lymphoma Cells. *Cell Rep.* **2016**, *14*, 1858–1866. [CrossRef]
18. Cyranoski, D. CRISPR gene-editing tested in a person for the first time. *Nature* **2016**, *539*, 479. [CrossRef]
19. Mitrofanov, A.; Alkhnbashi, O.S.; Shmakov, S.A.; Makarova, K.S.; Koonin, E.V.; Backofen, R. CRISPRidentify: Identification of CRISPR arrays using machine learning approach. *Nucleic Acids Res.* **2021**, *49*, e20. [CrossRef]
20. Way, G.P.; Sanchez-Vega, F.; La, K.; Armenia, J.; Chatila, W.K.; Luna, A.; Sander, C.; Cherniack, A.D.; Mina, M.; Ciriello, G.; et al. Machine Learning Detects Pan-cancer Ras Pathway Activation in The Cancer Genome Atlas. *Cell Rep.* **2018**, *23*, 172–180.e3. [CrossRef]

21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
22. Pratiwi, R.A.; Nurmaini, S.; Rini, D.; Rachmatullah, M.N.; Darmawahyuni, A. Deep ensemble learning for skin lesions classification with convolutional neural network. *IAES Int. J. Artif. Intell.* **2021**, *10*, 563. [CrossRef]
23. Duggento, A.; Conti, A.; Mauriello, A.; Guerrisi, M.; Toschi, N. Deep computational pathology in breast cancer. *Semin. Cancer Biol.* **2021**, *72*, 226–237. [CrossRef] [PubMed]
24. Madabhushi, A.; Lee, G. Image analysis and machine learning in digital pathology: Challenges and opportunities. *Med. Image Anal.* **2016**, *33*, 170–175. [CrossRef] [PubMed]
25. Wang, S.; Yang, D.M.; Rong, R.; Zhan, X.; Fujimoto, J.; Liu, H.; Minna, J.; Wistuba, I.I.; Xie, Y.; Xiao, G. Artificial Intelligence in Lung Cancer Pathology Image Analysis. *Cancers* **2019**, *11*, 1673. [CrossRef] [PubMed]
26. Qaiser, T.; Tsang, Y.W.; Taniyama, D.; Sakamoto, N.; Nakane, K.; Epstein, D.; Rajpoot, N. Fast and accurate tumor segmentation of histology images using persistent homology and deep convolutional features. *Med. Image Anal.* **2019**, *55*, 1–14. [CrossRef]
27. Khened, M.; Kori, A.; Rajkumar, H.; Krishnamurthi, G.; Srinivasan, B. A generalized deep learning framework for whole-slide image segmentation and analysis. *Sci. Rep.* **2021**, *11*, 11579. [CrossRef]
28. Song, Z.; Zou, S.; Zhou, W.; Huang, Y.; Shao, L.; Yuan, J.; Gou, X.; Jin, W.; Wang, Z.; Chen, X.; et al. Clinically applicable histopathological diagnosis system for gastric cancer detection using deep learning. *Nat. Commun.* **2020**, *11*, 4294. [CrossRef]
29. Wang, T.; Lei, Y.; Fu, Y.; Curran, W.J.; Liu, T.; Nye, J.A.; Yang, X. Machine learning in quantitative PET: A review of attenuation correction and low-count image reconstruction methods. *Phys. Medica* **2020**, *76*, 294–306. [CrossRef]
30. Hodneland, E.; Dybvik, J.A.; Wagner-Larsen, K.S.; Šoltészová, V.; Munthe-Kaas, A.Z.; Fasmer, K.E.; Krakstad, C.; Lundervold, A.; Lundervold, A.S.; Salvesen, Ø.; et al. Automated segmentation of endometrial cancer on MR images using deep learning. *Sci. Rep.* **2021**, *11*, 179. [CrossRef]
31. Papolos, A.; Narula, J.; Bavishi, C.; Chaudhry, F.A.; Sengupta, P.P. U.S. Hospital Use of Echocardiography. *J. Am. Coll. Cardiol.* **2016**, *67*, 502–511. [CrossRef]
32. Esteva, A.; Chou, K.; Yeung, S.; Naik, N.; Madani, A.; Mottaghi, A.; Liu, Y.; Topol, E.; Dean, J.; Socher, R. Deep learning-enabled medical computer vision. *NPJ Digit. Med.* **2021**, *4*, 5. [CrossRef] [PubMed]
33. Evans, A.J.; Bauer, T.W.; Bui, M.M.; Cornish, T.C.; Duncan, H.; Glassy, E.F.; Hipp, J.; McGee, R.S.; Murphy, D.; Myers, C.; et al. US Food and Drug Administration Approval of Whole Slide Imaging for Primary Diagnosis: A Key Milestone Is Reached and New Questions Are Raised. *Arch. Pathol. Lab. Med.* **2018**, *142*, 1383–1387. [CrossRef] [PubMed]
34. Srinidhi, C.L.; Ciga, O.; Martel, A.L. Deep neural network models for computational histopathology: A survey. *Med. Image Anal.* **2021**, *67*, 101813. [CrossRef]
35. Grist, J.T.; Withey, S.; MacPherson, L.; Oates, A.; Powell, S.; Novak, J.; Abernethy, L.; Pizer, B.; Grundy, R.; Bailey, S.; et al. Distinguishing between paediatric brain tumour types using multi-parametric magnetic resonance imaging and machine learning: A multi-site study. *Neuroimage Clin.* **2020**, *25*, 102172. [CrossRef]
36. Booth, T.C.; Williams, M.; Luis, A.; Cardoso, J.; Ashkan, K.; Shuaib, H. Machine learning and glioma imaging biomarkers. *Clin. Radiol.* **2020**, *75*, 20–32. [CrossRef]
37. Gupta, P.; Garg, S. Breast Cancer Prediction using varying Parameters of Machine Learning Models. *Procedia Comput. Sci.* **2020**, *171*, 593–601. [CrossRef]
38. Lewinson, R.T.; Meyers, D.E.; Vallerand, I.A.; Suo, A.; Dean, M.L.; Cheng, T.; Bebb, D.G.; Morris, D.G. Machine learning for prediction of cutaneous adverse events in patients receiving anti–PD-1 immunotherapy. *J. Am. Acad. Dermatol.* **2021**, *84*, 183–185. [CrossRef]
39. Bur, A.M.; Holcomb, A.; Goodwin, S.; Woodroof, J.; Karadaghy, O.; Shnayder, Y.; Kakarala, K.; Brant, J.; Shew, M. Machine learning to predict occult nodal metastasis in early oral squamous cell carcinoma. *Oral Oncol.* **2019**, *92*, 20–25. [CrossRef]
40. Lo Gullo, R.; Eskreis-Winkler, S.; Morris, E.A.; Pinker, K. Machine learning with multiparametric magnetic resonance imaging of the breast for early prediction of response to neoadjuvant chemotherapy. *Breast* **2020**, *49*, 115–122. [CrossRef]
41. Koike, Y.; Aokage, K.; Ikeda, K.; Nakai, T.; Tane, K.; Miyoshi, T.; Sugano, M.; Kojima, M.; Fujii, S.; Kuwata, T.; et al. Machine learning-based histological classification that predicts recurrence of peripheral lung squamous cell carcinoma. *Lung Cancer* **2020**, *147*, 252–258. [CrossRef] [PubMed]
42. Haider, S.P.; Sharaf, K.; Zeevi, T.; Baumeister, P.; Reichel, C.; Forghani, R.; Kann, B.H.; Petukhova, A.; Judson, B.L.; Prasad, M.L.; et al. Prediction of post-radiotherapy locoregional progression in HPV-associated oropharyngeal squamous cell carcinoma using machine-learning analysis of baseline PET/CT radiomics. *Transl. Oncol.* **2021**, *14*, 100906. [CrossRef] [PubMed]
43. Waring, J.; Lindvall, C.; Umeton, R. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artif. Intell. Med.* **2020**, *104*, 101822. [CrossRef]
44. Dlamini, Z.; Francies, F.Z.; Hull, R.; Marima, R. Artificial intelligence (AI) and big data in cancer and precision oncology. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 2300–2311. [CrossRef] [PubMed]
45. Issa, N.T.; Stathias, V.; Schürer, S.; Dakshanamurthy, S. Machine and deep learning approaches for cancer drug repurposing. *Semin. Cancer Biol.* **2021**, *68*, 132–142. [CrossRef] [PubMed]
46. Miao, R.; Chen, H.H.; Dang, Q.; Xia, L.Y.; Yang, Z.Y.; He, M.F.; Hao, Z.F.; Liang, Y. Beyond the limitation of targeted therapy: Improve the application of targeted drugs combining genomic data with machine learning. *Pharmacol. Res.* **2020**, *159*, 104932. [CrossRef]

47. Nicolae, A.; Semple, M.; Lu, L.; Smith, M.; Chung, H.; Loblaw, A.; Morton, G.; Mendez, L.C.; Tseng, C.L.; Davidson, M.; et al. Conventional vs machine learning–based treatment planning in prostate brachytherapy: Results of a Phase I randomized controlled trial. *Brachytherapy* **2020**, *19*, 470–476. [CrossRef] [PubMed]
48. Pan, X.; Levin-Epstein, R.; Huang, J.; Ruan, D.; King, C.R.; Kishan, A.U.; Steinberg, M.L.; Qi, X.S. Dosimetric predictors of patient-reported toxicity after prostate stereotactic body radiotherapy: Analysis of full range of the dose–volume histogram using ensemble machine learning. *Radiother. Oncol.* **2020**, *148*, 181–188. [CrossRef]
49. Yang, Z.; Olszewski, D.; He, C.; Pintea, G.; Lian, J.; Chou, T.; Chen, R.C.; Shtylla, B. Machine learning and statistical prediction of patient quality-of-life after prostate radiation therapy. *Comput. Biol. Med.* **2021**, *129*, 104127. [CrossRef]
50. Wiesweg, M.; Mairinger, F.; Reis, H.; Goetz, M.; Kollmeier, J.; Misch, D.; Stephan-Falkenau, S.; Mairinger, T.; Walter, R.F.H.; Hager, T.; et al. Machine learning reveals a PD-L1–independent prediction of response to immunotherapy of non-small cell lung cancer by gene expression context. *Eur. J. Cancer* **2020**, *140*, 76–85. [CrossRef]
51. Cuocolo, R.; Caruso, M.; Perillo, T.; Ugga, L.; Petretta, M. Machine Learning in oncology: A clinical appraisal. *Cancer Lett.* **2020**, *481*, 55–62. [CrossRef] [PubMed]
52. Nazari, M.; Shiri, I.; Zaidi, H. Radiomics-based machine learning model to predict risk of death within 5-years in clear cell renal cell carcinoma patients. *Comput. Biol. Med.* **2021**, *129*, 104135. [CrossRef] [PubMed]
53. Cepeda, S.; Arrese, I.; García-García, S.; Velasco-Casares, M.; Escudero-Caro, T.; Zamora, T.; Sarabia, R. Meningioma Consistency Can Be Defined by Combining the Radiomic Features of Magnetic Resonance Imaging and Ultrasound Elastography. A Pilot Study Using Machine Learning Classifiers. *World Neurosurg.* **2021**, *146*, e1147–e1159. [CrossRef] [PubMed]
54. Tian, L.; Zhang, D.; Bao, S.; Nie, P.; Hao, D.; Liu, Y.; Zhang, J.; Wang, H. Radiomics-based machine-learning method for prediction of distant metastasis from soft-tissue sarcomas. *Clin. Radiol.* **2021**, *76*, 158.e19–158.e25. [CrossRef] [PubMed]
55. Dercle, L.; Henry, T.; Carré, A.; Paragios, N.; Deutsch, E.; Robert, C. Reinventing radiation therapy with machine learning and imaging bio-markers (radiomics): State-of-the-art, challenges and perspectives. *Methods* **2021**, *188*, 44–60. [CrossRef] [PubMed]
56. Akcay, M.; Etiz, D.; Celik, O. Prediction of Survival and Recurrence Patterns by Machine Learning in Gastric Cancer Cases Undergoing Radiation Therapy and Chemotherapy. *Adv. Radiat. Oncol.* **2020**, *5*, 1179–1187. [CrossRef] [PubMed]
57. Wang, H.; Song, B.; Ye, N.; Ren, J.; Sun, X.; Dai, Z.; Zhang, Y.; Chen, B.T. Machine learning-based multiparametric MRI radiomics for predicting the aggressiveness of papillary thyroid carcinoma. *Eur. J. Radiol.* **2020**, *122*, 108755. [CrossRef]
58. Chatterjee, A.; Vallières, M.; Seuntjens, J. Overlooked pitfalls in multi-class machine learning classification in radiation oncology and how to avoid them. *Phys. Medica* **2020**, *70*, 96–100. [CrossRef]
59. Lu, Y.; Patel, M.; Natarajan, K.; Ughratdar, I.; Sanghera, P.; Jena, R.; Watts, C.; Sawlani, V. Machine learning-based radiomic, clinical and semantic feature analysis for predicting overall survival and MGMT promoter methylation status in patients with glioblastoma. *Magn. Reason. Imaging* **2020**, *74*, 161–170. [CrossRef]
60. Zhang, B.; He, X.; Ouyang, F.; Gu, D.; Dong, Y.; Zhang, L.; Mo, X.; Huang, W.; Tian, J.; Zhang, S. Radiomic machine-learning classifiers for prognostic biomarkers of advanced nasopharyngeal carcinoma. *Cancer Lett.* **2017**, *403*, 21–27. [CrossRef]
61. Sarkiss, C.A.; Germano, I.M. Machine Learning in Neuro-Oncology: Can Data Analysis From 5346 Patients Change Decision-Making Paradigms? *World Neurosurg.* **2019**, *124*, 287–294. [CrossRef] [PubMed]
62. Terparia, S.; Mir, R.; Tsang, Y.; Clark, C.H.; Patel, R. Automatic evaluation of contours in radiotherapy planning utilising conformity indices and machine learning. *Phys. Imaging Radiat. Oncol.* **2020**, *16*, 149–155. [CrossRef] [PubMed]
63. Alabi, R.O.; Mäkitie, A.A.; Pirinen, M.; Elmusrati, M.; Leivo, I.; Almangush, A. Comparison of nomogram with machine learning techniques for prediction of overall survival in patients with tongue cancer. *Int. J. Med. Inform.* **2021**, *145*, 104313. [CrossRef] [PubMed]
64. Karhade, A.V.; Thio, Q.; Ogink, P.; Kim, J.; Lozano-Calderon, S.; Raskin, K.; Schwab, J.H. Development of Machine Learning Algorithms for Prediction of 5-Year Spinal Chordoma Survival. *World Neurosurg.* **2018**, *119*, e842–e847. [CrossRef] [PubMed]
65. Du, X.; Min, J.; Shah, C.P.; Bishnoi, R.; Hogan, W.R.; Lemas, D.J. Predicting in-hospital mortality of patients with febrile neutropenia using machine learning models. *Int. J. Med. Inform.* **2020**, *139*, 104140. [CrossRef] [PubMed]
66. Praiss, A.M.; Huang, Y.; St Clair, C.M.; Tergas, A.I.; Melamed, A.; Khoury-Collado, F.; Hou, J.Y.; Hu, J.; Hur, C.; Hershman, D.L.; et al. Using machine learning to create prognostic systems for endometrial cancer. *Gynecol. Oncol.* **2020**, *159*, 744–750. [CrossRef] [PubMed]
67. Osterman, T.J.; Terry, M.; Miller, R.S. Improving Cancer Data Interoperability: The Promise of the Minimal Common Oncology Data Elements (mCODE) Initiative. *JCO Clin. Cancer Inform.* **2020**, *4*, 993–1001. [CrossRef]
68. Howell, D.; Molloy, S.; Wilkinson, K.; Green, E.; Orchard, K.; Wang, K.; Liberty, J. Patient-reported outcomes in routine cancer clinical practice: A scoping review of use, impact on health outcomes, and implementation factors. *Ann. Oncol.* **2015**, *26*, 1846–1858. [CrossRef]
69. Chua, I.S.; Gaziel-Yablowitz, M.; Korach, Z.T.; Kehl, K.L.; Levitan, N.A.; Arriaga, Y.E.; Jackson, G.P.; Bates, D.W.; Hassett, M. Artificial intelligence in oncology: Path to implementation. *Cancer Med.* **2021**, *10*, 4138–4149. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Method for Identification of Aberrations in Operational Data of Maritime Vessels and Sources Investigation

Jie Cai [1,*], Marie Lützen [2], Adeline Crystal John [3], Jakob Buus Petersen [3] and Niels Gorm Maly Rytter [1]

[1] Department of Technology and Innovation, University of Southern Denmark, Campusvej 55, 5230 Odense, Denmark; ngry@iti.sdu.dk
[2] Department of Mechanical and Electrical Engineering, University of Southern Denmark, Campusvej 55, 5230 Odense, Denmark; mlut@sdu.dk
[3] Vessel Performance Solution ApS, Diplomvej 381, 2800 Kgs Lyngby, Denmark; acj@vpsolutions.dk (A.C.J.); jbp@vpsolutions.dk (J.B.P.)
* Correspondence: jiec@iti.sdu.dk; Tel.: +45-6550-2850

Abstract: Sensing data from vessel operations are of great importance in reflecting operational performance and facilitating proper decision-making. In this paper, statistical analyses of vessel operational data are first conducted to compare manual noon reports and autolog data from sensors. Then, new indicators to identify data aberrations are proposed, which are the errors between the reported values from operational data and the expected values of different parameters based on baseline models and relevant sailing conditions. A method to detect aberrations based on the new indicators in terms of the reported power is then investigated, as there are two independent measured power values. In this method, a sliding window that moves forward along time is implemented, and the coefficient of variation (CV) is calculated for comparison. Case studies are carried out to detect aberrations in autolog and noon data from a commercial vessel using the new indicator. An analysis to explore the source of the deviation is also conducted, aiming to find the most reliable value in operations. The method is shown to be effective for practical use in detecting aberrations, having been initially tested on both autolog and noon report from four different commercial vessels in 14 vessel years. Approximately one triggered period per vessel per year with a conclusive deviation source is diagnosed by the proposed method. The investigation of this research will facilitate a better evaluation of operational performance, which is beneficial to both the vessel operators and crew.

Keywords: ship operations; data analysis; noon reports; autolog data

1. Introduction

The operational data that come from vessels are one of the most important sources with practical usage. Among others, in the pre-fixture stage of voyage planning, these data facilitate shipping companies finding the most navigated routes [1], which optimizes the sailing distance calculation and budget estimation. In addition, these data are normally adopted to monitor and evaluate vessel sailing performance by shipping companies [2].

In the endeavors of the shipping industry to meet the Paris agreement [3] for the improvement of its environmental footprint, vessel performance has become an essential issue that needs to be further optimized. Generally, the differences between practical vessel performance and a predefined baseline (defining ideal ship performance) should be minimized. Therefore, such operational data should be reliable enough for a proper ship performance estimation. However, they are normally subject to many sources of uncertainty, including faulty sensors, manual data collection, and even deliberate data manipulation, resulting in large deviations for the estimation of true ship performance. In this work, the period of recorded data variation that deviates from expected variations is defined as the aberration. To discover proper insights from such error-prone data and

reap benefits, it is necessary to develop practical methods to detect aberrations and work towards automatic sources and, eventually, root cause identification.

In shipping, there exist various ways to collect operational data from vessels. One is the noon report, which is filled out daily by the vessel crew. With the advancement of sensing and sensor technologies, autolog systems are utilized to automatically collect data onboard from various sensors [4–6]. Such operational data can be used to better understand and optimize navigation, energy consumption, emissions, and maintenance [7,8]. With the rapid development of new technology, including the Internet of Things (IoT), real-time connectivity, and artificial intelligence (AI) [9,10], looking at operational data and performance monitoring brings focus on quality diagnostics and root cause analysis, with the aim of the sustainable operation of shipping in the near future. The direct utilization of operational data is still limited due to the lack of standardized and reliable data sources. The application and implementation of machine learning (ML) is challenging due to the above-mentioned issues and other data quality problems. Hence, there is a necessity to further diagnose these phenomena.

Recent years have witnessed a considerable amount of research on vessel operational data. Cai et al. [1] developed voyage routes from automatic identification system (AIS) data based on clustering methods. Cai and Lützen [11] applied a sliding window to detect pattern changes in specific fuel oil consumption (SFOC) in noon reports. Dalheim and Steen [12] developed a method to monitor the change-point of vessel operational data from sensors based on hypothesis testing of samples inside a moving window, assuming that the underlying samples can be modeled by a deterministic linear trend model. The same authors in [13] proposed a stepwise framework to deal with the possible quality problems that occur in sensors and device data from vessel operations. Similar research and utilization of vessel operational data can also be found in Shelmerdine [14], Le Tixerant et al. [15], and Han and Yang [16].

However, there is still a lack of research on the identification of aberrations, as defined above, and especially a lack of effective methods to reveal the corresponding underlying reasons. These aberrations should be flagged, and the sources should be identified automatically in order to facilitate vessel operations. Therefore, the objective of this research is to investigate a practical method to identify aberrations in vessel operation based on both manual and sensor data that is capable of triggering alerts and, subsequently, concluding the underlying sources of such aberrations.

This paper is organized as follows: In Section 2, operational data are described, and the data of two vessels from the container shipping company Hapag-Lloyd [17] are compared. In Section 3, the methodology to identify data aberrations in vessel operations is elaborated upon, including baseline models and traditional indicators for ship performance, new proposed indicators, and the sliding window method. Section 4 contains the results of a case study applying the proposed method to the new indicator using data from a commercial vessel. In Section 5, the discussion and method limitations are further presented. Finally, the conclusions are drawn in Section 6.

2. Data Sources: Noon Reports and Autolog Data

In this section, the two types of data sources in vessel operations, noon reports and autolog data, are first described. Both of them will be used for the identification of data aberrations in Section 3. The statistical characteristics of the two data sources recording the same operational process are compared based on two vessels from the container operator Hapag-Lloyd. The results provide a glimpse of data source differences in practice. However, generic results cannot be derived due to the limited number of cases. The effect of using different data sources on the accuracy of the method is not studied in this paper.

Operational information from a vessel is recorded in different ways for regulatory and monitoring purposes. The noon report is one of the classical ways to record data. A noon report summarizes relevant sailing information and is manually prepared by vessel crew approximately every 24 h and shared with onshore offices. The noon report

includes information such as sailing time, sailed distance, course, engine run time, power produced, total fuel consumption, and weather conditions. These values are used in various performance analyses, for example, to monitor the degradation of vessel performance or to measure emissions.

Another method of collecting vessel operational data is using autolog systems that record information automatically through in-service monitoring systems and sensors deployed onboard. Autolog data from sensors and equipment onboard have, over the past few years, become more popular in the area of digital ship operations [2]. During a voyage, the operational data can be constantly logged at a high frequency, thereby capturing instantaneous variations in different parameters. The autolog system has been gradually improved and embraced by industry with the development of IoT technology, and it is expected to provide more reliable and accurate data than noon reports. The sample frequency of autolog data is flexible and normally set to be in minutes. However, the data are prone to be erroneous due to factors such as malfunctioning, unexpected sensor shifting, and asynchronization.

For comparison of the two data sources, the traditional performance indicator SFOC (see Section 3.1.1) is used. Note that the frequency of the autolog dataset has been down-sampled to hours (between 1 and 2 h) for better comparison, while the frequency of the noon report is roughly 1 day. Figures 1 and 2 show the results of the SFOC variations from noon and autolog data for vessel A and vessel B, respectively. The SFOC values in each sample are normalized by the SFOC baseline from shop tests performed by the engine manufacturers. Figures 3 and 4 illustrate the statistical distribution of the different operational data sources. Results show that, for vessel A, the noon report has a lower median value (from 99.2% to 105.2%) and a similar standard deviation (from 7.41% to 7.06%) compared to the autolog data. A simple hypothesis test is conducted as well, which is to test if the two mean values (here we use the mean value of the SFOC percentage: 100.83% and 106.0% for the noon report and autolog, respectively) are statistically equal with the null hypothesis as $H_0 : \mu = \mu_0$, where μ is the mean value of the SFOC from the noon report and μ_0 is the expected mean value of the SFOC from the autolog data. Since the population standard deviation is unknown, the sample standard deviation value is used, which is 7.41% in this case. The number of recorded data is 734. Thus, the t value is calculated as -18.9. If the level of significance (α) is selected as 0.05, the significant value ($t_{\alpha/2}$) can be then estimated from the student's t-distribution table, which is 1.963. Obviously, such an assumption is rejected, which means that the mean values of the two recorded datasets from vessel A cannot be considered the same statistically.

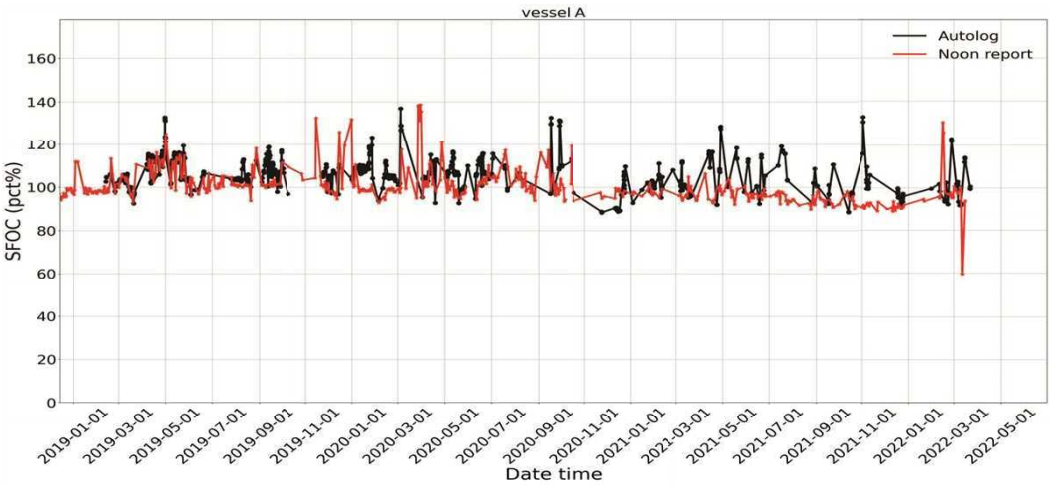


Figure 1. Comparison of the SFOC variation (in percent) of vessel A between noon report and autolog data in a given time period.

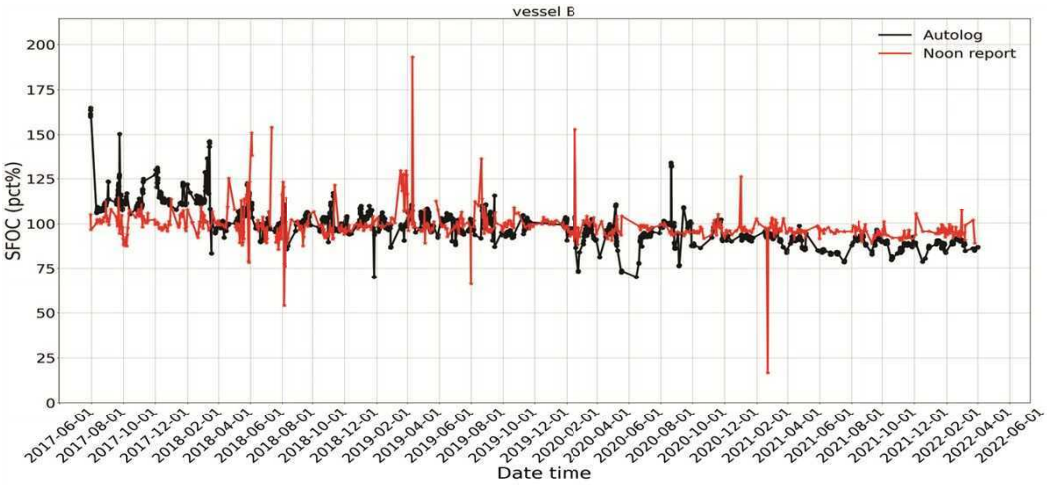


Figure 2. Comparison of the SFOC variation (in percent) of vessel B between noon report and autolog data in a given time period.

For vessel B, the noon report has a lower standard deviation (from 8.2% to 9.7%) and a nearly identical median value (from 98% to 98.3%) compared to the autolog data. More unexpected spikes are observed in the noon reports than in the autolog data for both vessels. The statistical distribution also shows that autolog data have a larger spread between the sample minimum and sample maximum but with fewer outliers, as indicated by the much longer whiskers in the left diagrams of both Figures 3 and 4. The same hypothesis test conducted for vessel A is conducted for vessel B to test if the mean values of the SFOC percentage are the same. Surprisingly, the calculated t value is 0.26, and the null hypothesis can be accepted, which means that the mean values recorded from the two datasets can be considered the same statistically.

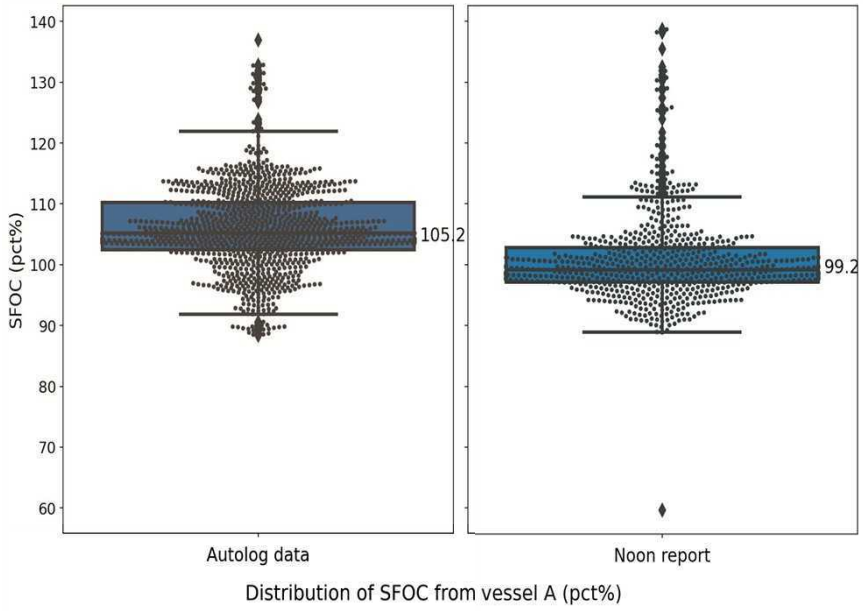


Figure 3. Statistical distribution of vessel A in the given time period.

It can be concluded that the two data sources represent similar vessel operational behavior but with statistical discrepancies due to different data acquisition methods and many uncertainties, such as sensor derivations under extreme environments and human errors. In the case of noon reports, e.g., vessel A, human factors may be much larger, causing more outliers, higher spikes, and statistical differences between mean values. The smaller standard deviation indicates that noon reports cannot sufficiently represent operational

details due to the 24 h time interval. The autolog data have higher variation, so they are capable of capturing realistic operational details with fewer outliers and fewer spikes but introduce more noise. It should be noted that the frequency effect has not been investigated, which may introduce other discrepancies.

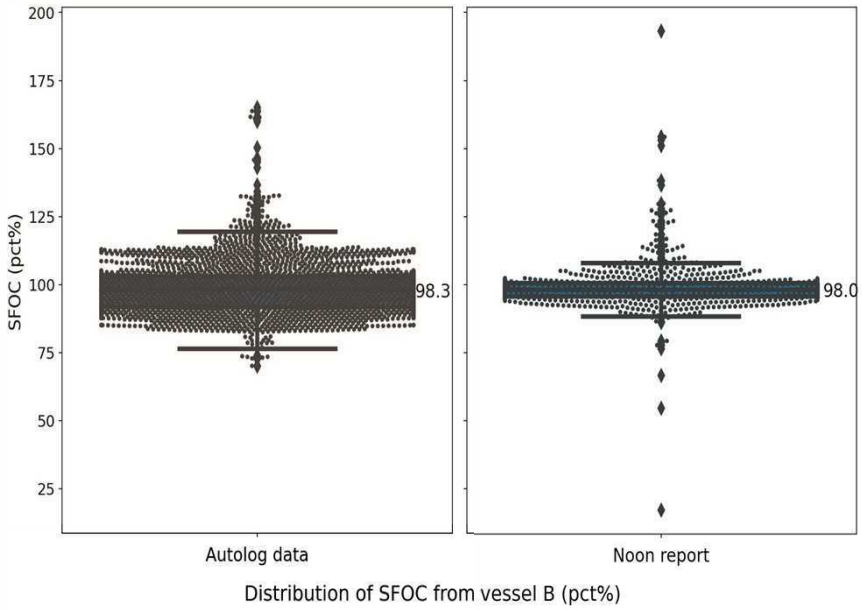


Figure 4. Statistical distribution of vessel B in the given time period.

3. Methodology

In this section, the practical method for data aberrations is proposed. The model can be applied to operational datasets from either noon reports or autolog data. Note that, in the following work, if there is no explicit indication, both types of operation data are applicable. In order to better introduce this method, the baselines for performance analysis and relevant traditional indicators in operations are first introduced, as seen in Section 3.1. Then, new indicators for data aberrations are developed in Section 3.2, which are the parameter errors calculated by our Trinity model (see Section 3.1.2). Using these new indicators, the sliding windows are applied to identify data aberrations, as illustrated in Section 3.3.

3.1. Performance Baselines and Traditional Indicators

When analyzing performance from vessels, traditional indicators, such as RPM, power, speed, SFOC, and resistance, are usually adopted, and the baseline models for them are vital to mutually defining ideal performance in operation, given different factors such as sea states, draft, and fouling levels. These models are normally determined by physical laws and binding data such as shop tests from the engine manufacturer and sea trials delivered by the shipyard. The baseline models used in this analysis and the relevant indicators are elaborated below.

3.1.1. Engine Baselines

The indicator SFOC is the measure of the mass of fuel consumed by the engine per unit time to produce a unit of power. It is unit-engine-specific, and normally given as the function of the engine load in the percentage of maximum continuous rating (MCR). Figure 5 shows an example of the main engine SFOC of vessel C's model from the tanker shipping company TORM [18]. The main engine SFOC model is an important baseline that can be used to estimate excess fuel consumption and evaluate engine power based on the fuel consumed in operations.

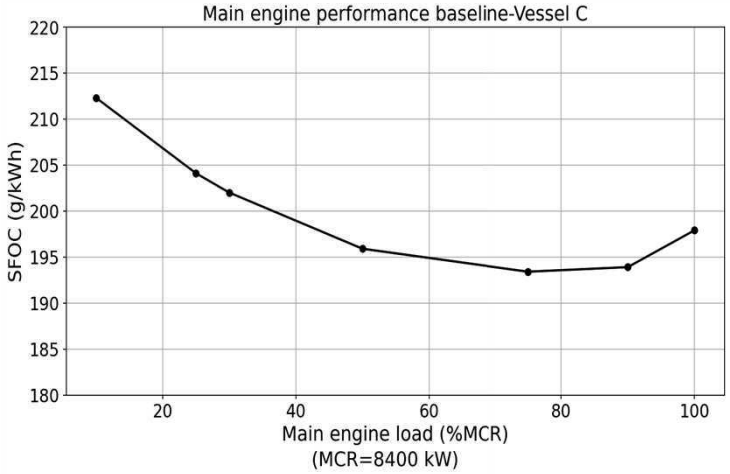


Figure 5. An example of SFOC diagram of the main engine from vessel C obtained from shop test [18].

Figure 6 shows another baseline model, which is the main engine load diagram, presenting the RPM–power limits of the engine of vessel A. *BL_continuous* denotes the continuous running condition of the engine when ample air is present in the combustion chamber to secure acceptable combustion and the maximum allowable loads are not exceeded. The *BL_overload* represents the overload operation limit of the engine, which is possible only for limited periods (e.g., 1 h per 12 h or when required in an emergency situation) [19]. The *BL_heavy_propeller* baseline represents the ideal power for continuous operation. The light propeller curve *BL_light_propeller* is the relation of power and RPM for the propeller with a clean hull and in calm weather. This curve shifts towards the *BL_heavy_propeller* because of heavy weather and a fouled hull, and the distance between these two curves is given by the light running margin (LRM). Details regarding the LRM are elaborated upon in Section 3.1.2.

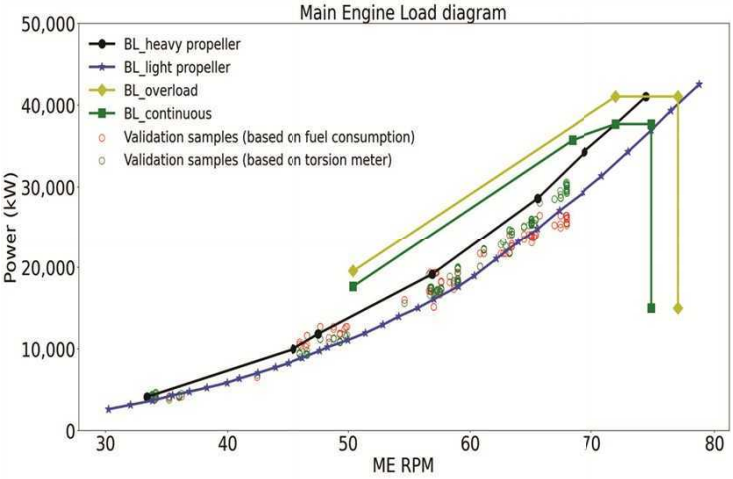


Figure 6. An example of the main engine load diagram from vessel A with validation points from operation [18].

The RPM–power baseline can be used to estimate vessel performance with the recorded power values in operations. However, in practice, there are two different methods to measure vessel power: one is the calculated value based on the measured fuel consumption from the fuel flow meter and the engine SFOC (the red dots in Figure 6), and the other is the direct measurement from torsion meters deployed on the shaft line (to measure the torque) and the RPM (the green dots in Figure 6). The problem is which one people should choose for a better estimation. Ideally, the powers obtained from these two methods should be the same since they reflect the same vessel operational process. However, there exist

deviations in reality, as described previously, resulting in different estimates of the vessel performance. Hence, one source must be a more consistently reliable measurement than the other, and the choice of the right source will further affect the evaluation accuracy of operational performance.

3.1.2. Speed–Power–RPM Baselines

According to the physical laws and specific designs of vessels, there is a fixed relation between speed, power, and RPM, which will form the basis of the new proposed indicators (seen in Section 3.2) in this work. This speed–power–RPM baseline of the vessel is, hereafter, referred to as the Trinity model. Note that the Trinity model is vessel design-specific and is determined based on several factors, including the principal dimensions of the vessel, the design speed, power, and RPM, and, most importantly, the vessel's own sea trial performance. In addition, the baseline models are only for vessels with a fixed-pitch propeller. These curves form the basis of the calculation of expected relations (speed–power–RPM) accounting for specific loading conditions, draft, and weather during operations.

The process of developing this model is briefly explained here. The Trinity model estimate starts with the evaluation of the power (P) required by vessels to maintain the sailing speed (V) given the resistance (R_T) of the vessel, as seen in Equation (1).

$$P = R_T \cdot V \tag{1}$$

In practice, the vessel resistance is normally estimated by semi-empirical methods that are widely used in the literature since they are relatively simple and reasonably accurate [20–22]. Equation (2) expresses the semi-empirical equation of vessel resistance, where S is the hull-wetted surface area at the draft, ρ is the density of sea water, V is the measured speed, and C_T is the total resistance coefficient, which can be influenced by factors such as wind, wave, current, types of propeller, water depth, and hull biofouling. Therefore, if the estimated resistance is provided, the expected power can be estimated (Equation (1)).

$$R_T = \frac{1}{2} \cdot C_T \cdot \rho \cdot S \cdot V^2 \tag{2}$$

To build the relation between power and RPM at the design point, the LRM is used, which can be expressed as Equation (3) [19], where RPM_{light} and RPM_{heavy} are the propeller RPM at the light and heavy load conditions, respectively. The relation of the power and RPM is constructed by the relation $P \propto (LRM \cdot RPM)^3$ according to the so-called "cubic law" based on hydrodynamic principles [23].

$$LRM = \frac{RPM_{light} - RPM_{heavy}}{RPM_{heavy}} \cdot 100 \tag{3}$$

In this fashion, the speed–power–RPM Trinity model of the newly built vessel in calm sea conditions is constructed as a baseline. This model can easily be calibrated to different weather conditions and degrees of hull and propeller fouling. Due to confidentiality requirements from the shipping company, we cannot provide further information on specific resistance coefficients, hull degradation due to fouling, or other relevant factors. Therefore, the "expected" value of the different parameters can be estimated based on the Trinity model with different input subsets in vessel operations. Note that, in this analysis, the vessel sea trial performance is used as the baseline of evaluating hull degradation over time. Figure 7 shows an example of the speed–power and speed–RPM plots for vessel A at the draft of 14.5 m.

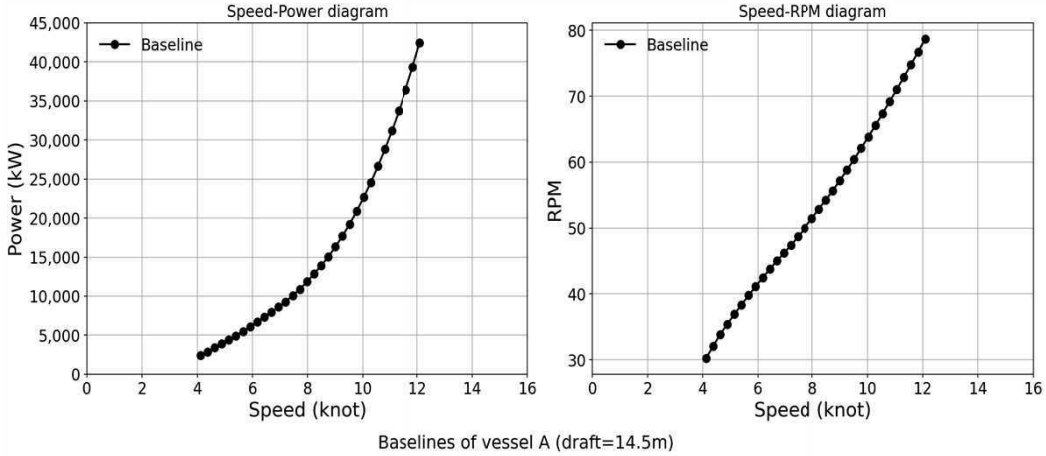


Figure 7. Baselines of the vessel A for speed, power, and RPM.

3.2. New Indicators for Data Aberrations

3.2.1. Motivation

As previously described in Section 3.1.1, there exist two measured power values: one based on the fuel consumption and the other based on the torsion meter. The proper choice of recorded power will further affect the accuracy of performance estimation. Meanwhile, operational data are subject to many sources of uncertainty, such as faulty sensors and manual data collection, introducing even more challenges to accurately estimate ship performance from the data-driven standpoint. Therefore, one big motivation is to propose new indicators to counteract such uncertainties, which will, in turn, facilitate the selection of the more accurate parameters for performance estimation, such as the more accurate power value, and thereby diagnose which of the two measured values can better represent the actual performance of vessels. In this work, the new proposed indicator is the error between the reported values from operational data and the expected values based on baseline models and relevant sailing conditions, which is furthered explained in the next section.

3.2.2. Process to Develop New Indicators

The specific procedure of calculating the new indicator is given in Figure 8. For each data sample in an operation, the expected vessel resistance and the expected power are first calculated through Equation (2) and Equation (1), respectively. The Trinity model is calibrated accounting for the reported weather conditions, in terms of wind speed, wave height, and currents, and the assumed hull deterioration [24]. Note that it is necessary to assume such hull deterioration, otherwise the performance loss of the vessel will be always determined by the reported power values. The assumed deterioration is statistically estimated based on the trend of hull performance in the previous reports, taking into account possible noise. Thus, the expected speed–power–RPM curves are obtained for each sample in the operational dataset. All errors of speed, RPM, and power should be evaluated, as all these errors are used for the final source identification. Using these curves, a look-up of expected value (speed, power, and RPM) can be conducted through linear interpolation using the corresponding operational measurements. For instance, based on the measured RPM value, the expected speed and power values can be looked up from the Trinity model. Note that one parameter is always assumed to be correct when the expected values of the other two are estimated. In this fashion, the errors of the speed, power, and RPM for each sample in the dataset can be calculated.

For brevity reasons, the power value based on the measured fuel consumption from the fuel flow meter and the engine performance baseline SFOC (Figure 5) is defined as the fuel-consumption-based estimation method, denoted as the FC method. The power that is the direct measurement from torsion meters using the torque and RPM is the torsion-meter-based estimation method, denoted as TM method.

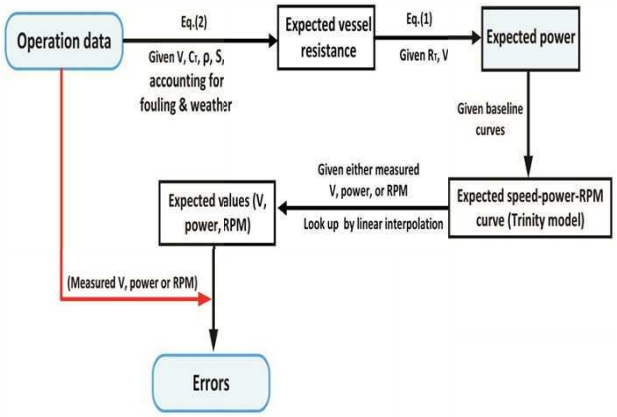


Figure 8. Flowchart of the procedure for the new indicators (parameter errors).

As there are two estimates of hull deterioration based on the different measured value of power, there are two sets of calibrated curves for each record. Therefore, for the calculation of errors of each parameter, there are two corresponding values when using the two different hull deterioration values. Using the proposed procedure as mentioned above (Figure 8), all the parameter errors can be calculated (see the flowchat in Figure 9). Therefore, it will produce twelve (or six pairs of) errors in total, as seen in Table 1. The same pattern of notation for all parameter errors will be used in this research, for instance, ER_RPM_Speed_TM denotes the calculated RPM error with the given value of speed using the performance deterioration from the TM method.

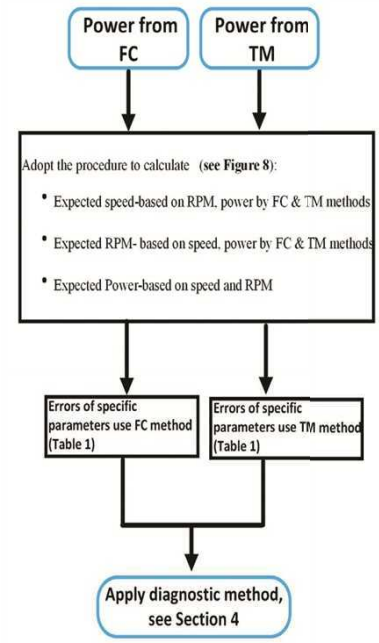


Figure 9. Flowchart of the parameter errors (speed, RPM, and power) calculated from two different power sources (FC and TM) using the new process (Figure 8, Table 1, Section 4).

Since it is not known which of the two signals is better, neither stand-alone error (either with power from TM or with power from FC) can be used to diagnose a faulty signal; instead, we adopt the pair of calculated errors for further analysis. The sliding CV [11] method (see Section 3.3) is applied to detect data aberrations from the error pairs. For instance, the two power errors with given RPM (ER_Power_RPM_FC and ER_Power_RPM_TM) are further analyzed to identify pattern changes and come up with conclusive results.

Table 1. All possible errors based on the new procedure under two different power sources (FC or TM).

Power Source	Error	Given Parameter
FC or TM	Power error	RPM
	Power error	Speed
	RPM error	Power
	RPM error	Speed
	Speed error	RPM
	Speed error	Power

3.3. The Method to Identify Aberrations

In our previous research [11], an original sliding method was proposed to identify unstable periods for a single data variation. In this method, historical CV values of performance indicators bounded by a sliding window with a fixed width and steps were used for the identification of unstable periods. In this study, the same concept is used but with a focus on the detection of aberrations based on a pair of new indicators that can be applied to both noon reports and autolog data. The CV values are calculated within each sliding window to trace changes.

3.3.1. Sliding Window

Figure 10 illustrates the sliding window along the timeline of samples with a window width of n and a step size. In this example, the width of five samples is used, and every time it moves forward by two samples, so that there is an overlapping of each sliding window. In practice, the determination of the window width and the step size should be conducted by a sensitivity study for each vessel based on the historical variation in operational data. According to our previous study [11], for most of the vessels from the shipping company TORM [18], a width of thirty samples and a step size of one sample have been adequate. In each sliding window, the CV value is calculated, expressed as the ratio of the standard deviation to the mean value, denoting the extent of data variability around the mean. This method is quite straightforward and the application is simple, as shown in Figure 11. The CV values of the two power errors (denoted as CV_error_FC and CV_error_TM) are directly calculated. The discrepancy of two corresponding CVs in absolute format is then compared, which will trigger alarms when it is violated by a given threshold. Sources can be found by analysis, and crew feedback will be signalled in the operational system (see Section 4.3).

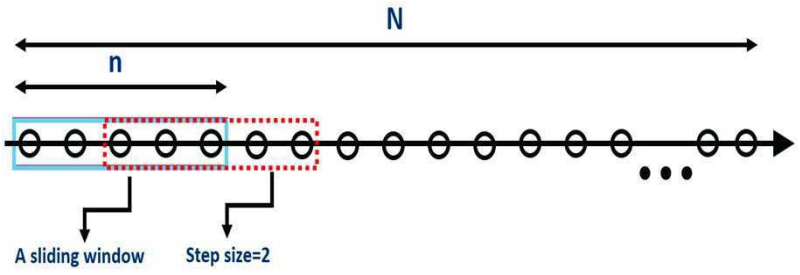


Figure 10. Illustration of the sliding window along the time series data samples.

This method can be applied to time series data in different sample frequencies for both online and offline pattern changes identification. Since the sliding window has a width of n samples, the length of dataset (N) should have at least n samples, $N > n$, so that the change along the timeline can be seen and the historical data can be used to determine the threshold and window width for each vessel. When a sample is added, a new calculation is conducted with the moving forward of a sliding window. In this way, the shift can be detected, accounting for the effect of the past n samples. In this regard, a time delay equal to the width of the sliding window may be introduced before a source and the feedback have

been given due to triggered alarms. This method is generic in application, i.e., independent of the types of operational data sources such as noon reports or autolog data from sensors.

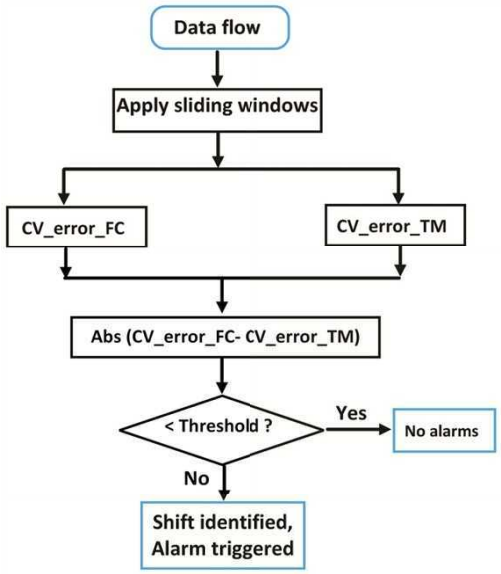


Figure 11. Flowchart of the application of the practical method on the new indicators from two power models.

4. Case Studies and Results

To assess the effectiveness of the proposed method and the new indicators, the time series data from vessel A are selected, and the new indicators in terms of power errors are used for analysis. A sensitivity study is first carried out to determine the effect of value signs when performing CV calculations.

4.1. Sensitivity Study

By nature, the new indicators (Section 3.2) can take negative values when expected values are smaller than measured values. This largely affects the CV in a sliding window and the application of the practical method. Figure 12 shows an example of the absolute power errors with given RPM from both autolog and noon report data, calculated by the FC method and the TM method, respectively. As shown, the power errors are unequally distributed on both sides of the X-axis. An exceptional case would be that the denominator (mean value) of the CVs approaches zero since the original values are equally distributed on both sides of the X-axis. As a consequence, the corresponding CV would approach infinite. As demonstrated in Figure 13, the CVs with signs are visualized, and spikes of CVs are observed due to infinite values. Therefore, before applying the method, a pre-processing of the power errors needs to be performed, which transforms all values into positives so as to rescale the CVs in a reasonable range for the proposed identification method.

When applying the practical method, another sensitivity study is necessary to determine the proper width and the moving step of the sliding windows, as already illustrated in Section 3.3.1. In our previous study [11], an example has been made of the vessels from the shipping company TORM, where a width of thirty samples and a step size of one sample are selected. Therefore, similar work is not repeated in this paper but the same results are adopted for a further case study. Note that, for different ships, the recommendation is to perform such a sensitivity analysis before utilizing the method.

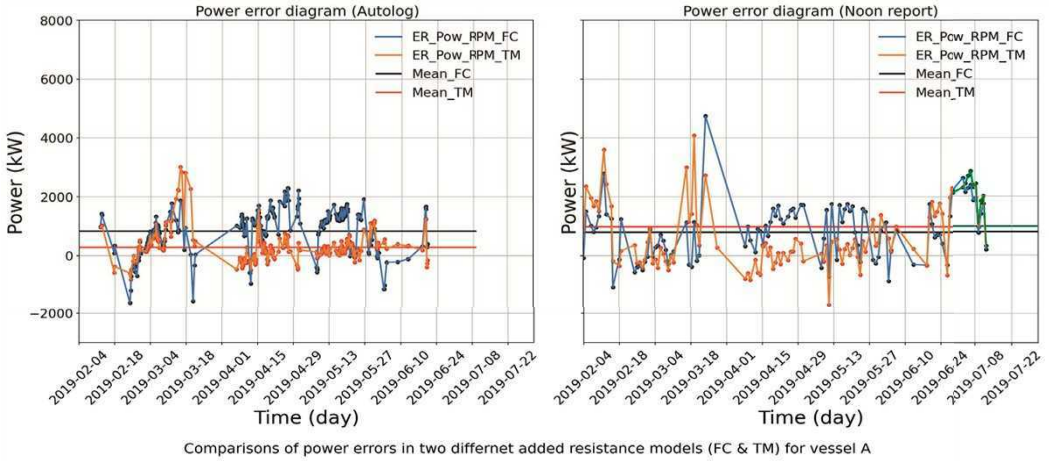


Figure 12. Comparisons of the power errors with signs from FC method and TM method, respectively, for vessel A.

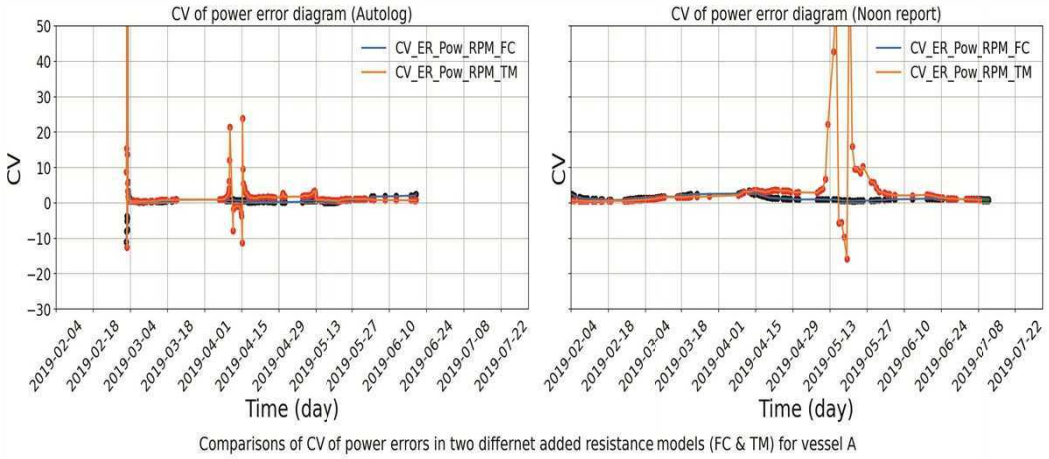


Figure 13. CV of the power errors with signs from FC method and TM method, respectively, for vessel A.

4.2. Identified Aberration Periods

The operational data in vessel A from Hapag-Lloyd are used to identify aberrations by the proposed method. The absolute power errors given the RPM in the two ways for autolog data are presented. Figure 14 illustrates the CV variation in the power errors given the RPM for the period between February and July in 2019. Here, the orange curve denotes the new indicator values calculated by the TM method and the blue curve denotes values by the FC method. Since it is expected that both indicators follow the same trend, any remarkable deviation in the values implies a change or shift in the pattern of the values. The discrepancy between the CV for the two methods can be clearly visualized at a proper scale, and the periods violating a set threshold are highlighted. An alarm will be automatically triggered in this situation. Note that the threshold value is based on the observation of historical data variation and industry experience, which may introduce subjective bias. Periods where the difference between the CV is above threshold indicate that one of the values is not so reliable. The isolation of the source of the deviation is explained in the next section, which will facilitate the selection of more accurate power values.

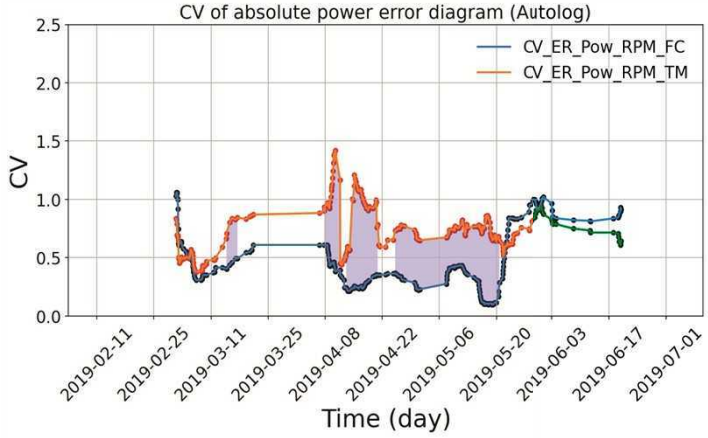


Figure 14. Application of the practical method on power errors (absolute values) for vessel A.

4.3. Deviation Source Analysis

Once an aberration period has been identified, the sources for the identified changes can be further analyzed using baselines (Section 3). For this purpose, the aberrant period from 18 April to 20 May in 2019 (see Figure 14) is selected for detailed analysis.

Figure 15 shows the calculated RPM errors in the detected aberration period estimated by the procedure in Section 3.2. The two horizontal lines in the images are the mean values of the indicator in the given period. The TM method has a lower mean error in the RPM value compared to the FC method, which suggests that the reported power value measured by the torsion meter is more aligned to the previously reported power values from the torsion meter in the source report than those from the fuel flow meter. As presented in this figure, the mean error of RPM based on the speed estimated from the FC method is 2.58 rpm, while the one from the TM method is 1.51 rpm. The mean error of the RPM based on the power estimated from the FC method is 1.77 rpm, while the one from the TM method is 0.30 rpm. Hence, further verification is needed of the fuel consumption values in reports in this period.

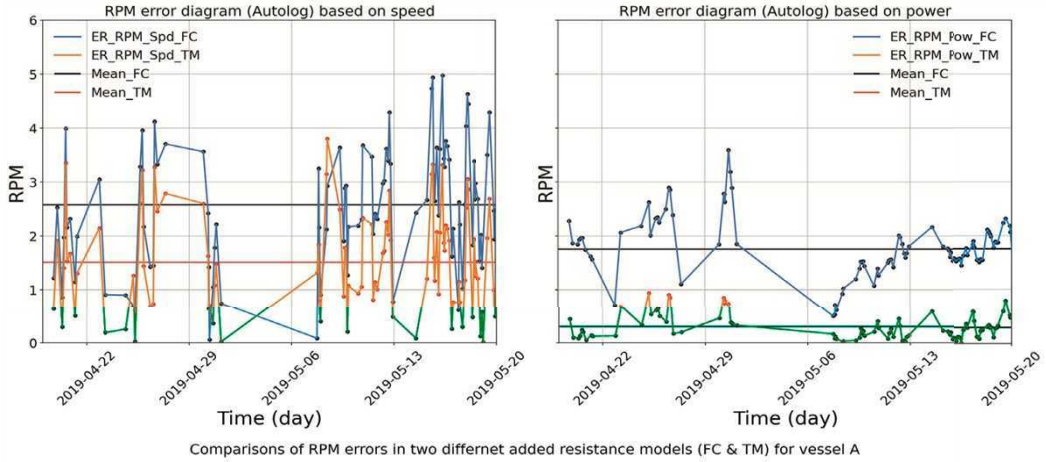


Figure 15. Comparisons of RPM errors from FC method and TM method, respectively, for vessel A.

Figure 16 is the comparison of the last 10 measured samples from the end of the aberration period in vessel A, which further demonstrates the above conclusion. The speed–RPM dimension is used, as the speed–RPM values for each sample are identical for the two ways and are reliable. The black curve is the original speed–RPM baseline from the light propeller curve, while the blue and the red curves denote the calibrated baselines using FC and TM methods for the last sample, respectively. The measured samples fit better to the curve calibrated by the TM method. The overall analysis thus indicates that the

power measured by the torsion meter is more reliable in this identified period and the fuel consumption value has a higher discrepancy and is less reliable to use. Note that the source analysis for the identified aberration periods in this paper is only a demonstration case that was manually conducted to find more reliable values in operations. It is case-dependent for individual vessels. In another vessel, it might be the fuel meter that is more reliable or some errors may occur in either the fuel meters or torsion meters. Thus, extensive domain knowledge in shipping is needed to conduct such an analysis, which limits its industry application. Further work is needed to generalize different aberration periods so that the data aberration source can be automatically identified.

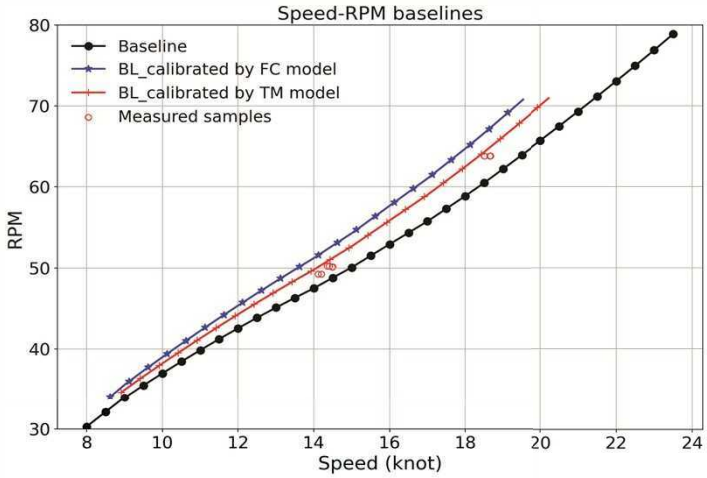


Figure 16. Comparisons between baselines and the last 10 measured samples from the pattern change in vessel A.

5. Discussion

The practical method has been tested on both autolog and noon report data from four different commercial vessels belonging to Hapag-Lloyd, covering 14 vessel years. For confidential reasons, only the final results are summarized here. There are 76 aberration periods found in the case of autolog data and 17 for the noon data based on the new indicators and sliding CV method. Deeper analyses were carried out for all detected periods, where 15.3% of all identified aberrations in autolog data were found to have a conclusive source, with 28.6% in the case of noon reports. This implies that, out of the 76 identified aberration periods for autolog data, 9 were found to have a conclusive source of deviation from either the fuel consumption or the torsion meter. In the case of the noon reports, 4 out of the 17 had a conclusive deviation source. The remainder of the aberration periods were inconclusive due to ambiguity and contradictory conclusions from the methods or too short aberration periods. It is considered that the detection of an aberration period is an important finding as these findings help the selection of more accurate operational parameters for a performance estimation.

To put it into practical terms, there are approximately five aberration periods detected per vessel per year, and there is approximately one triggered period per vessel per year with a conclusive deviation source diagnosed by the proposed method. As no or very little research has been conducted in this field by the shipping industry so far, to the best knowledge of the authors, this is believed to be a significant result. The effectiveness of the proposed method to identify such deviations is of great interest as a start. The derived results in this research are advantageous to implement in diagnostic tools, and the results would be beneficial to facilitate vessel operators and crew in real-time operations.

However, limitations exist for this method and the investigation of sources. The Trinity model may have errors and implicit biases, which need to be kept in mind. In the sliding CV method, the threshold value to trigger alarms is manually defined based on experience and historical data variation, which may introduce subjective bias. A sensitivity study is

needed for each vessel or for the same type of vessels to determine the width of the sliding window, which requires manual effort. The method adopts the moving average concept, which means that data quality issues, such as spikes, outliers, and missing values, will have considerable influence on its effectiveness. The sliding CV method is better suited to diagnose a shift or jump but cannot easily identify constant offset values. Finally, the effect of data source differences on the accuracy of the method has not been analyzed, and needs to be further studied. For instance, over a long period of vessel sailing, possible deviations of onboard sensors, such as sensor drift, sensor offset, sensor sensitivity, and sensor hysteresis, may bring uncertainties to the investigation of sources of identified aberrations.

6. Conclusions

The comparison between the noon reports and autolog data concludes that the two data sources reflect similar vessel operational behavior but with statistical discrepancies. A practical method to identify aberrations in the time series data from ship operations based on new indicators is presented using the sliding window and CV values. This can be applied for various types of operational data, including noon reports and autolog data. Case studies have been conducted using data from commercial vessels that show how this method can be utilized. Source identification analysis has been performed to discover reasons for the triggered changes. The major contributions of this paper are summarized, as shown in the following.

- The noon report and autolog data represent similar vessel operational behavior but with statistical discrepancies due to different data acquisition methods and many uncertainties, such as sensor derivations under extreme environments and human errors.
- New indicators are proposed from the Trinity model of speed, power, and RPM, where weather and hull deterioration during operation is considered.
- The practical method based on the proposed new indicators can be applied to identify data aberrations.
- Despite its time-consuming and manual conduction, the practical deviation source analysis method provides an effective way to select more accurate parameters for ship performance estimation in the complex maritime environment, especially with so many sources of uncertainty during operational data recording.

There are still many deviations that cannot be explicitly explained because of ambiguity, variation tendencies, and noise, which is another big challenge. Further research that focuses on these limitations and drawbacks of the practical method is needed. Applying the method on data—both historical and real-time—can reveal consistent aberration periods that earlier would have been overlooked as noise. The awareness of this shift is a valuable asset to vessel operators and crew, who have better access to analyze the cause of the shift. It has also been demonstrated that the method can be used to isolate the source of the shift in selected cases. Keeping in mind the call for responsible and environmentally sustainable shipping practices, such tools may be immensely powerful in improving performance analyses in the near future. Additionally, this diagnostic tool could facilitate data verification organizations in implementing stricter validation rules for submitted data. Such practical methods in this research are the first step towards bringing a clearer view into the nuances of vessel operational data.

Author Contributions: Conceptualization, J.C., M.L. and A.C.J.; methodology, J.C. and A.C.J.; software, J.C. and A.C.J.; validation, J.C., M.L. and A.C.J.; formal analysis, J.C. and A.C.J.; investigation, J.C.; writing—original draft preparation, J.C.; writing—review and editing, J.C., M.L., A.C.J., J.B.P. and N.G.M.R.; visualization, J.C.; supervision, M.L., J.B.P. and N.G.M.R.; project administration, J.B.P.; funding acquisition, M.L., J.B.P. and N.G.M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research work, belonging to the ShippingLab project "Digital vessel operation", was funded by the Innovation Fund Denmark (IFD) under grant No. 8090-00063B, the Danish Maritime Fund, the Lauritzen Fonden, and the Orient's Fond.

Data Availability Statement: The noon report data and the autolog data used in this research are unavailable due to confidentiality reasons of the companies.

Acknowledgments: Our gratitude is expressed to the tanker shipping company TORM and container shipping company Hapag Lloyd for their vessel data.

Conflicts of Interest: Authors Adeline Crystal John and Jakob Buus Petersen were employed by the company Vessel Performance Solution ApS, 2800 Kongens Lyngby, Copenhagen, Denmark. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Cai, J.; Chen, G.; Lützen, M.; Rytter, N.G.M. A practical AIS-based route library for voyage planning at the pre-fixture stage. *Ocean Eng.* **2021**, *236*, 109478. [CrossRef]
2. Perera, L.P.; Mo, B. Ship performance and navigation information under high-dimensional digital models. *J. Mar. Sci. Technol.* **2020**, *25*, 81–92. [CrossRef]
3. Agreement, P. Paris Agreement. In Proceedings of the Report of the Conference of the Parties to the United Nations Framework Convention on Climate Change (21st Session, 2015: Paris), Retrieved December, HeinOnline, Paris, France, 30 November–13 December 2015; Volume 4, p. 2017.
4. Ou, Z.; Zhu, J. AIS database powered by GIS technology for maritime safety and security. *J. Navig.* **2008**, *61*, 655–665. [CrossRef]
5. IMO. International convention for the safety of life at sea. *Int. Marit. Organ.* **2012**, *142*. Available online: http://library.arcticportal.org/1696/ (accessed on 3 July 2012).
6. Svanberg, M.; Santén, V.; Hörteborn, A.; Holm, H.; Finnsgård, C. AIS in maritime research. *Mar. Policy* **2019**, *106*, 103520. [CrossRef]
7. Rong, H.; Teixeira, A.; Soares, C.G. Data mining approach to shipping route characterization and anomaly detection based on AIS data. *Ocean Eng.* **2020**, *198*, 106936. [CrossRef]
8. Silveira, P.; Teixeira, A.; Guedes-Soares, C. AIS based shipping routes using the Dijkstra algorithm. *TransNav: Int. J. Mar. Navig. Saf. Sea Transp.* **2019**, *13*, 565–571. [CrossRef]
9. De Mauro, A.; Greco, M.; Grimaldi, M. What is big data? A consensual definition and a review of key research topics. In *Proceedings of the AIP Conference Proceedings*; American Institute of Physics: College Park, MD, USA, 2015; Volume 1644, pp. 97–104. [CrossRef]
10. Hermann, M.; Pentek, T.; Otto, B. Design principles for industrie 4.0 scenarios. In Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, USA, 5–8 January 2016; pp. 3928–3937. [CrossRef]
11. Cai, J.; Lützen, M. A Practical identification method for data variation of noon reports in vessel operations. In Proceedings of the International Conference on Naval Architecture and Ocean & Marine Engineering, Haikou, China, 21–23 October 2022; p. 1. [CrossRef]
12. Dalheim, Ø.Ø.; Steen, S. A computationally efficient method for identification of steady state in time series data from ship monitoring. *J. Ocean Eng. Sci.* **2020**, *5*, 333–345. [CrossRef]
13. Dalheim, Ø.Ø.; Steen, S. Preparation of in-service measurement data for ship operation and performance analysis. *Ocean Eng.* **2020**, *212*, 107730. [CrossRef]
14. Shelmerdine, R.L. Teasing out the detail: How our understanding of marine AIS data can better inform industries, developments, and planning. *Mar. Policy* **2015**, *54*, 17–25. [CrossRef]
15. Le Tixerant, M.; Le Guyader, D.; Gourmelon, F.; Queffelec, B. How can Automatic Identification System (AIS) data be used for maritime spatial planning? *Ocean Coast. Manag.* **2018**, *166*, 18–30. [CrossRef]
16. Han, P.; Yang, X. Big data-driven automatic generation of ship route planning in complex maritime environments. *Acta Oceanol. Sin.* **2020**, *39*, 113–120. [CrossRef]
17. Hapag-LIoyd. Hapag-LIoyd. 1847. Available online: https://www.hapag-lloyd.com/en/home.html (accessed on 20 March 2024).
18. TORM. TORM SHIPPING. 1889. Available online: https://www.torm.com/ (accessed on 20 March 2024).
19. Diesel, M. Turbo. In *Basic Principles of Ship Propulsion*; MAN SE: Munich, Germany, 2011.
20. Kristensen, H.O.; Lützen, M. Prediction of resistance and propulsion power of ships. *Clean Shipp. Curr.* **2012**, *1*, 1–52.
21. Aas-Hansen, M. Monitoring of Hull Condition of Ships. Master's Thesis, Norges Teknisk-Naturvitenskapelige Universitet, Trondheim, Norway, 2010.
22. Faltinsen, O. *Sea Loads on Ships and Offshore Structures*; Cambridge University Press: Cambridge, MA, USA, 1993; Volume 1.
23. Psaraftis, H.N.; Lagouvardou, S. Ship speed vs power or fuel consumption: Are laws of physics still valid? Regression analysis pitfalls and misguided policy implications. *Clean. Logist. Supply Chain* **2023**, *7*, 100111. [CrossRef]
24. Hasselaar, T.W.F. An Investigation into the Development of an Advanced Ship Performance Monitoring and Analysis System. Ph.D. Thesis, Newcastle University, Newcastle upon Tyne, UK, 2011.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Improving Graph Convolutional Network with Learnable Edge Weights and Edge-Node Co-Embedding for Graph Anomaly Detection

Xiao Tan [1], Jianfeng Yang [1,*], Zhengang Zhao [2], Jinsheng Xiao [1] and Chengwang Li [3]

[1] School of Electronic Information, Wuhan University, Wuhan 430072, China; haiantanxiao@whu.edu.cn (X.T.); xiaojs@whu.edu.cn (J.X.)
[2] School of Software Engineering, University of Science and Technology of China, Suzhou 215123, China; gavin@ustc.edu.cn
[3] College of Sciences, China Jiliang University, Hangzhou 310018, China; licw980824@126.com
* Correspondence: yjf@whu.edu.cn; Tel.: +86-1350-719-1603

Abstract: The era of Industry 4.0 is gradually transforming our society into a data-driven one, which can help us uncover valuable information from accumulated data, thereby improving the level of social governance. The detection of anomalies, is crucial for maintaining societal trust and fairness, yet it poses significant challenges due to the ubiquity of anomalies and the difficulty in identifying them accurately. This paper aims to enhance the performance of the current Graph Convolutional Network (GCN)-based Graph Anomaly Detection (GAD) algorithm on datasets with extremely low proportions of anomalous labels. This goal is achieved through modifying the GCN network structure and conducting feature extraction, thus fully utilizing three types of information in the graph: node label information, node feature information, and edge information. Firstly, we theoretically demonstrate the relationship between label propagation and feature convolution, indicating that the Label Propagation Algorithm (LPA) can serve as a regularization penalty term for GCN, aiding in training and enabling learnable edge weights, providing a basis for incorporating node label information into GCN networks. Secondly, we introduce a method to aggregate node and edge features, thereby incorporating edge information into GCN networks. Finally, we design different GCN trainable weights for node features and co-embedding features. This design allows different features to be projected into different spaces, greatly enhancing model expressiveness. Experimental results on the DGraph dataset demonstrate superior AUC performance compared to baseline models, highlighting the feasibility and efficacy of the proposed approach in addressing GAD tasks in the scene with extremely low proportions of anomalous data.

Keywords: graph anomaly detection; label propagation; graph convolutional neural networks; semi-supervised learning

1. Introduction

In real-world settings, anomalies are both ubiquitous and potentially detrimental, yet challenging to identify. Fraudulent activities pose significant threats to the robust functioning of society, not only in economic terms but also by adversely impacting societal trust and fairness. Anomaly detection, also referred to as fraud detection, is assuming an increasingly crucial role in contemporary society. Its applications range from detecting fake news [1] to identifying fraudulent reviews [2].

In academia, "anomaly" is commonly abstracted as a set of data that deviates from the majority [3]. Despite the development of numerous techniques in recent years for uncovering outliers and anomalies in unstructured sets of multi-dimensional points, the scarcity of anomalous behavior, high labeling costs, and privacy protection policies result in a paucity of data samples labeled as anomalous [3].

We observe that graphs in the current graph anomaly detection domain mainly exhibit the following three characteristics.

1. Nodes are Labeled as well as Featured. Nodes in graphs often possess both label and feature properties. Node labels can propagate and aggregate along the edges of the graph, while node features can propagate along the edges and undergo transformations through neural network layers. Both label and feature propagation are beneficial for predicting node behavior. The effective combination of these two propagation aspects warrants attention.
2. Edges Contain Rich Information. In the field of GAD, the topology and structure of the graph contain a wealth of information. The attributes and directions of edges imply that even in isomorphic graphs, there are multiple types of connections (i.e., edges) between different nodes. For instance, relationships between users may include transfer and recipient relationships, loan and borrower relationships, as well as familial relationships. This information can assist in better classification, thereby identifying anomalous nodes. However, this information is often overlooked by other methods.
3. The Proportion of Anomalous Data is Extremely Low. While fraud incidents do occur, the majority of users in reality are normal users. The proportion of fraudulent users is significantly smaller than that of normal users, making fraud detection a classic example of imbalanced data. In machine learning model training, achieving a relatively balanced distribution of various sample classes is crucial for optimal performance. When there is a substantial difference in the proportions of different classes, the trained model tends to exhibit characteristics more aligned with the majority class, thereby neglecting features associated with minority class samples.

The inherent suitability of graph structures for analyzing relationships and connectivity patterns in networks has propelled GAD to the forefront. GAD abstracts traditional streaming data into graph structures, where nodes represent entities and edges signify relationships between entities. The interconnected nodes form edges that convey structural information about the graph, information that can be stored using an adjacency matrix [4]. GAD aims to detect anomalies by leveraging the structural information inherent in the network, combining it with classic anomaly detection approaches. Considering fraud scenarios, fraudulent activities often exhibit associations and collaboration. The connections between fraud groups are typically closer than those between normal users, and the behaviors demonstrate a certain level of similarity. This also explains why graph analysis can be employed to detect anomalous behaviors in practical scenarios. According to the different basic graph algorithms, GAD algorithms can be classified into several types, including community detection, feature discovery, pattern mining, and GCN. This paper focuses on the GCN-based GAD algorithm.

Due to divergent definitions of "anomaly" in academic and industrial contexts, coupled with the variability in identifying anomalous data across different scenarios, the precise characterization of what constitutes an "anomaly" proves challenging [3]. This paper abstains from delving into the root cause behind the classification of specific data as anomalous or enumerating the features that render data anomalous. Instead, our focus remains exclusively on the handling of datasets with anomaly ground-truth. Leveraging information from node features, edge features, and graph structures, we seek to infer potential label values for unlabeled nodes based on existing positive and negative labels. The objective is to simplify the model into a semi-supervised learning classification task rooted in graph neural networks, and the main task is to determine whether a node on the graph is an anomaly or not. This classification approach is also often employed in GAD tasks [5].

We propose a unified model capable of addressing the aforementioned three challenges in graphs to make full use of the information in the nodes and edges to improve the performance of the algorithm. The primary contributions of this paper include the following.

1. We analyzed the theoretical relationship between the feature convolution process of GCN and the label propagation process of LPA, demonstrating that using LPA as an auxiliary regularization term in the GCN loss function can train edge weights and thereby improve GCN performance. Building on this theoretical analysis, we introduced a learnable edge weight into GCN and trained it using LPA. This method modifies the convolution process of GCN, making full use of node labels.
2. Considering the heterogeneity of edges, we analyzed the impact of edge information on GAD tasks. Firstly, we embedded edge direction and attribute information to generate edge features. Then, we aggregated these edge features with node features to generate co-embedding. This method conducted feature extraction, fully utilizing the information on edges.
3. Based on the co-embedding, we designed separate trainable GCN weights for node features and co-embedding features, projecting them into different feature spaces, greatly enhancing the model's generalization ability.

We conducted experiments on the publicly available DGraph [6] dataset, selecting MLP, Node2Vec [7], LPA [8], GCN [9], GAT [10], GATv2 [11], GraphSAGE [12], and UniMP [13] as baseline models, with AUC serving as the evaluation metric, which refers to the area under the ROC curve. The ROC (Receiver Operating Characteristic) curve is a graphical visualization that provides an intuitive representation of a model's performance. As a numeric value, the AUC can directly and intuitively exhibit the goodness or poorness of a model's performance. The results indicate that our proposed method outperforms the baselines in terms of AUC. Our model proves feasible in addressing GAD tasks, providing valuable insights for applications in scenarios with scarce data.

The remaining sections of this paper are organized as follows. Section 2 discusses related work in the field of anomaly detection. Section 3 defines the mathematical model of the problem, analyzing the theoretical connection between GCN's convolution on node features and LPA's propagation on node labels, providing a mathematical basis for incorporating node labels into the GCN model. Section 4 introduces our GCN-based GAD model, which is achieved through three enhancements to the GCN network. Section 5 presents experimental results on public datasets, comparing the performance with baseline models, indicating that our method exhibits the best AUC performance on datasets with extremely low proportions of anomalous nodes. Finally, in Section 6, we summarize the key contributions of this paper, as well as directions for future improvement.

2. Related Work

Anomaly detection algorithms typically view the task as a node classification problem, where data are categorized into normal and abnormal classes [3–5]. Our algorithm conceptually relates to semi-supervised methods in graph-based learning and aligns with recent advancements in applying convolutional neural networks to graph-structured data.

2.1. Limitations Encountered by Current Algorithms

Traditional anomaly detection methods can be divided into rule-based methods and probability-based methods. Rule-based methods aim to formalize expert knowledge rules and replicate them through computational means, while probability-based methods rely on statistical knowledge and assume that data distribution conforms to certain statistical distributions. Traditional anomaly detection methods are highly interpretable, but they need to design different detection rules for different scenarios, which requires the introduction of a large amount of expert knowledge. This method has high time and labor costs, and the generalization of the method is weak.

2.2. Anomaly Detection Based on Semi-Supervised Methods

Machine learning anomaly detection methods can be categorized into supervised, semi-supervised, and unsupervised learning. In cases where the number of negative samples is small, resulting in a large disparity between positive and negative sample

proportions, and the patterns of negative samples are difficult to learn, supervised learning methods are not applicable. Unsupervised learning does not require considering prior knowledge and is suitable for most cases where most objects are normal. However, due to insufficient input information, unsupervised learning is prone to the curse of dimensionality, leading to low accuracy and high false-positive rates in existing patterns.

Semi-supervised learning does not rely on external interaction and automatically leverages unlabeled samples to enhance learning based on a training set composed entirely of normal data [14,15]. It allows for joint learning between a small number of labeled samples and a large number of unlabeled samples [16], leading to improved learning performance and generalization ability [17,18]. By better capturing data distribution characteristics, it is particularly suitable for scenarios with limited labeled data and expensive manual annotation. Specifically, semi-supervised learning can be categorized into inductive learning (or pure semi-supervised learning) and transductive learning [14]. In the former, the assumption is that the unlabeled samples in the training data do not represent the data to be tested, while in the latter, the unlabeled samples considered during the learning process are precisely the ones to be predicted [17].

In the mentioned methods, LPA and GNN methods are the focal points of our attention, they can be summarized in Table 1. From the table, we observe that LPA is typically associated with transductive learning, implying its inability to generalize to new nodes; instead, it solely classifies the remaining nodes in the original graph. This is because, during the model learning process, nodes with unknown labels in the original graph may also be utilized for training. In other words, certain unlabeled data obtain their label information from other unlabeled data. GNN is associated with inductive learning [19]. Additionally, GNNs can leverage node features and support graph embedding, whereas LPA cannot. LPA, however, can make full use of node label information, which GNNs may not fully exploit. Consequently, both algorithms alone may not effectively utilize the information present in the nodes of the graph. Our primary research goal is to cleverly combine these two algorithms.

Table 1. Comparison of semi-supervised anomaly detection methods.

	Label Propagation	**Graph Neural Network**
Graph Embedding	no	yes
Representation Learning	no	yes
Use Features	no	yes
Use Labels	yes	no
Transductive	yes	yes
Inductive	no	yes/no

LPA was initially proposed by Zhu and Ghahramani [8], and implemented using bipartite graphs. Several anomaly detection algorithms have been proposed based on LPA. Li et al. [20] used adaptive label propagation for group anomaly detection in networks. Wang et al. [21] introduced a collaborative multi-label propagation strategy, leveraging a user-item interaction graph for acceleration. Wang et al. [22] improves graph-based label propagation algorithm with group partition for fraud detection.

With GNN algorithms, numerous effective anomaly detection algorithms have also been proposed. Building upon the dynamic heterogeneous graph, Zhang et al. [23] introduced the hierarchical multi-modal fusion GNN (HMF-GNN), which was applied to the detection of health insurance claim records. This model addresses the challenge of lost neighbor information. DAGAD [24] provides a promising solution to alleviating the problem of GAD with anomalous sample scarcity and class imbalance.

However, the aforementioned methods have not been able to fully integrate the information provided by node features and labels.

2.3. Anomaly Detection Based on Graph Convolutional Networks

According to whether the node attributes in the graph are multivariate, graphs can be classified into homogeneous and heterogeneous graphs [25], this paper exclusively discusses homogeneous graphs. For graph data, anomalies can be considered as elements such as nodes, edges, subgraphs, etc., that differ from the majority of objects in the graph. We can categorize GAD into different types based on the properties of the graph (static graph, static attribute graph, and dynamic graph), and the nature of anomalies (node-based, edge-based, and graph-based) [4,26]. The focus of this paper is on the detection of anomalous nodes in static attribute graphs. In attribute graphs, besides the structural information between nodes, each node also encompasses attribute information [27]. These attribute details aid in uncovering hidden features of nodes, thereby enhancing the accuracy of node anomaly detection.

Our approach is primarily inspired by GCN-based algorithms. Kipf and Welling [9] introduced the concept of convolution to graph structures, proposing GCN. Building on the work of GCN, Hamilton et al. [12] improved the mean aggregation method and introduced GraphSAGE, a method for inductive representation learning, enabling algorithmic application to large-scale graphs through minibatch sampling. However, GAD algorithms constructed based on the above methods, such as GEM [28] and DualFraud [29], have not taken into consideration the benefits brought by node labels.

Other methods, like GCN-LPA [30], and UniMP [13], cleverly combine GCN with LPA. Huang et al. [19] demonstrated that this combination can avoid excessive smoothing, thereby enhancing the performance of GCN. However, the aforementioned methods have not fully utilized the information embedded in edges.

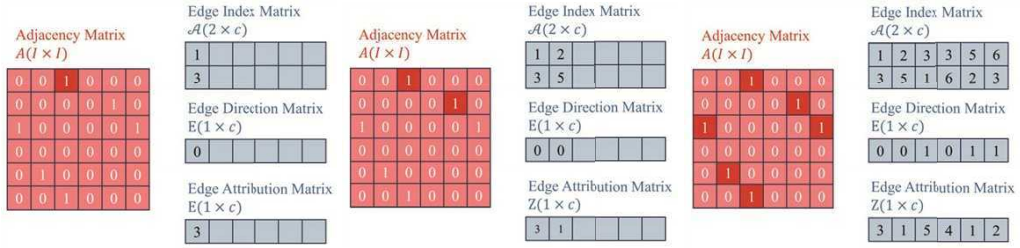
Research on combining GCN with LPA for recommender systems (RS) has proven the feasibility of this approach in graph node classification [30–32]. Li et al. [33] combined LPA with GNN and proposed a GAD method applicable to e-commerce shopping networks.

3. Problem Formulation

We start with some notation. Assume that there is a graph, directed or undirected, $G = (V, E)$, in which $V = \{v_1, \ldots, v_I\}$ denote the sets of subjects to be detected. Each node has a P-dimensional feature, the i-th node's feature is represented by x_i. The arrangement of all features in a row by node orders the results in the feature matrix $X \in \mathbb{R}^{I \times P}$. $A \in \{0,1\}^{I \times I}$ is the binary adjacency matrix, the ij-th entry $a_{ij} \in \{0,1\}$ of A means whether there is an edge from v_i to v_j, and vice versa. Number the elements of A from the first row to the n-th row as $1, 2, \ldots, n+1, n+2, \ldots, nn-1, nn$; therefore, A can be flattened into a two-dimensional array $\mathcal{A} \in \mathbb{N}^{2 \times c}$, called the edge index matrix, where the first row represents the source node and the second row represents the target node. c is the number of non-zero elements in the matrix A, as well as the number of edges in the graph. Partition A into two triangular matrices, A_{upper} and A_{lower}. For an undirected graph, the first $c/2$ elements of $\mathcal{A}$, correspond to the ith non-zero element in A_{upper}; the last $c/2$ elements of $\mathcal{A}$ correspond to the i-th non-zero element in A_{lower}. D is the diagonal degree matrix, and $S = D^{-1/2}AD^{-1/2}$ is the normalized adjacency matrix.

For graph G, each edge comprises two elements: edge direction and attribution. Matrices E and Z, are organized based on the corresponding order of edges. That is to say, E and Z shares the same topology as $\mathcal{A}$. The matrix $E \in \{0,1\}^{1 \times c}$ stores the directions of the edges, where 0 and 1 represent incoming edges and outgoing edges. The matrix $Z \in \{1, \ldots, n\}^{1 \times c}$ stores the attributes of the edges, where different values of element Z_i represent different attributes of the edges. Figure 1 shows the process of mapping the adjacency matrix A to the edge index matrix $\mathcal{A}$, as well as the construction of the edge direction matrix E and the edge attribution matrix Z. In (a), the first non-zero element in A is read, which is located in the first row and third column. Therefore, it represents that the corresponding edge has a source node of index 1 and a target node of index 3. Consequently, the first column of the edge index matrix $\mathcal{A}$, denoted as $\mathcal{A}_1$, has 1 in the row above (representing the source node) and 3 in the row below (representing the target

node). For the edge direction matrix, since this element is located in the upper triangular matrix A_{upper}, the first element of the edge direction matrix E_1 is 0, indicating an incoming edge. The first element of the edge attribution matrix Z_1 stores attribute 3 of this edge. Following this procedure, we iterate through the other non-zero elements in A to construct $\mathcal{A}$, E, and Z.



(**a**) (**b**) (**c**)

Figure 1. The process of mapping the adjacency matrix A to the edge index matrix $\mathcal{A}$, as well as the construction of E and Z. (**a**) The first non-zero element in A is mapped to $\mathcal{A}$. (**b**) The second non-zero element in A is mapped to $\mathcal{A}$. (**c**) The remaining non-zero elements in A are mapped to $\mathcal{A}$.

We can transform the directed graph G into an undirected graph UG through a simple symmetric operation. This involves processing the elements in matrices A with the transformation a_{ij} or $a_{ji} = \begin{cases} a_{ji} \text{ or } a_{ij}, & \text{if } a_{ij} \text{ or } a_{ji} \neq 0 \\ 0, & \text{otherwise.} \end{cases}$

Consequently, A becomes symmetric matrices.

In GAD tasks, the node set V is divided into a disjoint set of unlabeled nodes and labeled nodes, while only the first m nodes ($m \ll n$) have labels from a label set $L = \{1, 2\}$. Here, 1 represents normal nodes, and 2 represents anomalous nodes. The label is stored by a one-hot-encoding matrix $Y = [y_1, \ldots, y_n]^T$.

Finally, V is divided into three sets in proportion: training set V_{train}, validation set V_{valid}, and test set V_{test}. Our objective is to learn the features of nodes with labels, propagate their labels, and then infer the labels of the remaining nodes, given G, X, Y. Training is performed on V_{train}, validation is conducted on V_{valid}, and testing is carried out on V_{test}.

Table 2 summarizes the symbols used in this paper along with their corresponding meanings.

Table 2. Symbols used in this paper.

Symbols	Meanings	Symbols	Meanings
G	graph	$\mathcal{A}$	edge index matrix
V	node matrix	E	edge direction matrix
v_i	the i-th node	Z	edge attribution matrix
Y	label matrix	H	representation of nodes
y_i	the i-th label	Θ	GCN trainable weight
A	adjacency matrix	W	learnable edge weight
a_{ij}	edge from v_i to v_j	I_f	feature influence
D	diagonal degree matrix	I_l	label influence
S	normalized adjacency matrix		

3.1. Label Propagation Algorithm

LPA is based on the assumption that two connected nodes are correlated because of the homophily and influence. LPA utilizes node labels as its input.

We construct a one-hot-encoding label matrix $Y^{(k)} = \left[y_1^{(k)}, \ldots, y_n^{(k)}\right]^T$ to store labels. The i-th row $y_i^{(k)T}$ stores the predicted label of node y_i after k times label propagation. The label matrix $Y^{(0)}$ is initialized with the rule $y_i^{(0)} = \begin{cases} y_i & i \leq m, \text{ labeled nodes} \\ zero - vector & m < i \leq n, \text{ unlabeled nodes} \end{cases}$.

In the label propagation process during the k-th iteration, all nodes propagate their labels to neighboring nodes with Equation (1). From the perspective of nodes, node v_i will choose the community of its neighbor with the highest membership count as its new community:

$$Y^{(k+1)} = D^{-1}AY^{(k)}. \tag{1}$$

Labels are propagated from each other nodes through a normalized adjacency matrix $D^{-1}A$. Subsequently, all nodes are reset back to their initial labels $y_i^{(0)}$, before proceeding to the $(k+1)$-th iteration:

$$y_i^{(k+1)} = y_i^{(0)}, \forall i \leq m. \tag{2}$$

LPA is an iterative computation process and does not guarantee convergence. Approximately five iterations are often sufficient to achieve convergence [8], striking a balance between accuracy and performance. This will be the chosen number of iterations in our study. Consequently, for node v, the probability of category l is calculated as:

$$P(Y_v = c) = \frac{1}{\sum_{e(v,u)} A_{v,u}} \sum_{e(v,u)} A_{v,u} P(Y_u = c), \tag{3}$$

where $e(v,u) \in E$ represents the edge from v to its neighbor u. Figure 2 illustrates an LPA process executed on an unweighted, undirected graph.

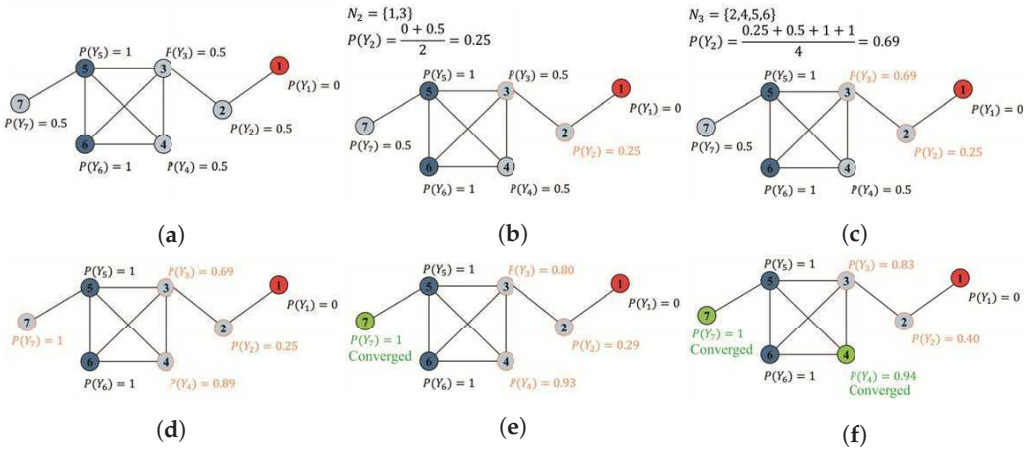


Figure 2. Label propagation process in three iterations: a demo. In the first iteration, the algorithm starts computation from the first unlabeled node (i.e., node 2). For node 2, $N_2 = \{1,3\}$. Here, N_i represents the neighboring nodes of node i. The next unlabeled node is then computed (i.e., node 3), using the updated value of node 2. After the first iteration, all nodes are reset back to their initial labels according to Equation (2). After multiple iterations (three iterations are shown here), some node values will approach convergence, representing the final label probabilities for nodes. (**a**) Initial state. (**b**) Update for node 2 in the 1st iteration. (**c**) Update for node 3 in the 1st iteration. (**d**) After a round of updates for all unlabeled nodes (iteration 1). (**e**) After iteration 2. (**f**) After iteration 3.

In case of an undirected graph, the final representation of node i is the weighted average of its neighboring nodes $j \in N(i)$, represented by $y_i^{\infty} = \sum_{j \in N(i)} a_{ij} y_j^{\infty}$.

3.2. Graph Convolutional Neural Networks

GCN [9] is one of the most popular GAD models based on the Laplacian smoothing assumption. GCN utilizes node features as its input. In general, GCN employs a sum aggregator as an aggregation function, the discussions in this paper are based on it. We begin our derivation from the spatial representation of GCN. (The connection between LPA and GCN is observed in their spatial representations, both of which have the form of $AH\Theta$. The original derivation of GCN starts from the spectral domain of graph convolution, and the spatial representation of GCN is an approximation using Weisfeiler-Lehman on the spectral

domain of graph convolution. However, for LPA, the original paper directly provides a spatial representation without a spectral domain expression. Reference [34] introduces energy to represent information propagation and then considers energy Laplacian, but its form differs significantly from the spectral domain expression of GCN.)

The feature propagation scheme of GCN in layer l can be formulaed as:

$$H^{(l+1)} = \sigma\left(\tilde{S}H^{(l)}\Theta^{(l)}\right) \tag{4}$$

where σ is an non-linear activation function, $\Theta^{(l)}$ denotes the GCN trainable weight in the l-th layer, and $H^{(l)}$ is the l-th layer representations of nodes with $H^{(l)} = X$ for the input layer. To avoid neglecting the node's own features, self-loops are added when updating the node itself. This involves combining the node's own features with the features of its neighbors during the node update process. The normalized adjacency matrix will be $\tilde{S} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, where $\tilde{A} = A + I$, and $\tilde{D}$ is the diagonal degree matrix for $\tilde{A}$ with entries $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. Of particular note is the adjacency matrix $\tilde{S}$, which here remains fixed, and depends on the topological structure of the constructed graph. Meanwhile, the weight Θ mentioned is the trainable weight of GCN, which is different from our learnable edge weight.

By stacking GCN, features for nodes in an l-dimensional space can be obtained. In GAD tasks, the final layer of GCN is set to an output dimension of 2, allowing for the classification of all nodes into normal/abnormal binary categories. Generally, stacking two layers of GCN already yields satisfactory results [9], and this will be the number of GCN layers used in this paper. At this point, the features of nodes after two layers of GCN are represented as:

$$H^{(2)} = \sigma\left(\tilde{S}\sigma\left(\tilde{S}X\Theta^{(0)}\right)\Theta^{(1)}\right) \tag{5}$$

Similar to LPA, in the case of an undirected graph and sum aggregator, the final representation of a node i is the weighted sum of its neighboring nodes $j \in N(i)$, represented by $h_i^{\infty} = \sum_{j\in N(i)} a_{ij}x_j^{\infty}$.

3.3. Relationship between LPA and GCN

According to analysis before, on the label side L, LPA can propagate and aggregate node labels along edges; along feature side V, GCN can convolute and transform node features along edges. We can also observe that Equations (1) and 4 share similarities in their forms. In this section, we attempt to answer two questions: What is the relationship between the label propagation of LPA and the feature convolution of GCN? Why can LPA serve as edge weights for training GCN?

Consider two nodes v_1 and v_2 in a graph; v_1 is unlabeled while v_2 is labeled. We study the relationship between LPA and GCN from the perspective of label influence and feature influence. Especially, if the initial label (feature) of v_2 changes, how does the output label (feature) of v_1 change. The label (feature) influence can be measured by the gradient (Jacobian) of the output label (feature) of v_1 with respect to the initial label (feature) of v_2 [35,36]. The label influence of v_2 on v_1 after k iterations of LPA is:

$$I_l(v_1, v_1; k) = \frac{\partial y_1^{(k)}}{\partial y_1}. \tag{6}$$

Due to the iterative nature of the LPA algorithm, the influence of x_2 on x_1 accumulates, and can eventually be expressed as:

$$I_l(v_1, v_2; k) = \sum_{j=0}^{k-1} \frac{\partial y_1^{(k)}}{\partial y_2^{(j)}}. \tag{7}$$

On the other hand, denote $x_1^{(k)}$ as the k-th layer embedding of v_1 in GCN, and x_2 represents the initial embedding of v_2. The feature influence of v_2 on v_1 after k-layers of GCN is the L1-norm of the expected Jacobian matrix [30], as:

$$I_f(v_1, v_2; k) = \left\| \mathbb{E}_{\Theta^{(\cdot)}} \left[\frac{\partial \mathbf{x}_1^{(k)}}{\partial \mathbf{x}_2} \right] \right\|_1, \tag{8}$$

where the expectation is taken under the transformation matrix $\Theta^{(\cdot)}$. The feature influence should be normalized as:

$$\tilde{I}_f(v_1, v_2; k) = \frac{I_f(v_a, v_b; k)}{\sum_{v_i \in V} I_f(v_1, v_i; k)}. \tag{9}$$

Assume that the ReLU function serves as the activation function σ of GCN, and β denotes the fraction of unlabeled nodes. Consequently, the relationship between the label propagation of LPA and the feature convolution of GCN can be concluded, which can be expressed as:

$$\mathbb{E}_{\Theta^{(\cdot)}}[I_l(v_1, v_2; k)] = \sum_{j=0}^{k-1} \beta^j \tilde{I}_f(v_1, v_2; j). \tag{10}$$

The details of the proof for Equation (10) can be found in Reference [30]. With this, the first question posed in this section is answered: the relationship between label propagation and feature convolution is that if the label of node v_2 has a sufficiently large influence on the label of node v_1, then the initial features of node v_2 will also have an equally large impact on the output features of node v_1. In other words, the influence of all nodes labeled as c on the label of node v_1, i.e.,

$$\sum_{\forall v_2, \text{ label of } v_2 \text{ is } c} I_l(v_1, v_2; k) \tag{11}$$

is proportional to the probability of node v_1 being classified as c by LPA as Equation (3), i.e.,

$$P\left(\hat{y_1}^{lpa} = c\right). \tag{12}$$

From the above derivation, we can conclude that label influence can serve as a substitute for feature influence, which is the answer to the second question. This provides a theoretical basis for us to use LPA to assist in training edge weights for GCN and to establish a GCN model where edge weights are learnable.

4. Our Unified GCN-Based GAD Model

This section will introduce our unified GCN-based GAD model, primarily consisting of three clever enhancements built upon the GCN network: learnable edge weights, edge-node co-embedding, and separate trainable GCN weight. Learnable edge weights introduce node label information through LPA, edge-node co-embedding incorporates information from edges, and separate trainable GCN weights enhance the model's expressive ability. We aim to leverage the improvements mentioned above to fully utilize node feature information, node label information, and edge information in the GAD graph, thereby enhancing the algorithm's performance in scenarios with extremely low proportions of anomalous data. Figure 3 illustrates the overall architecture of our unified GCN-based GAD model. In contrast, Figure 4 illustrates the structure of a classical GCN network.

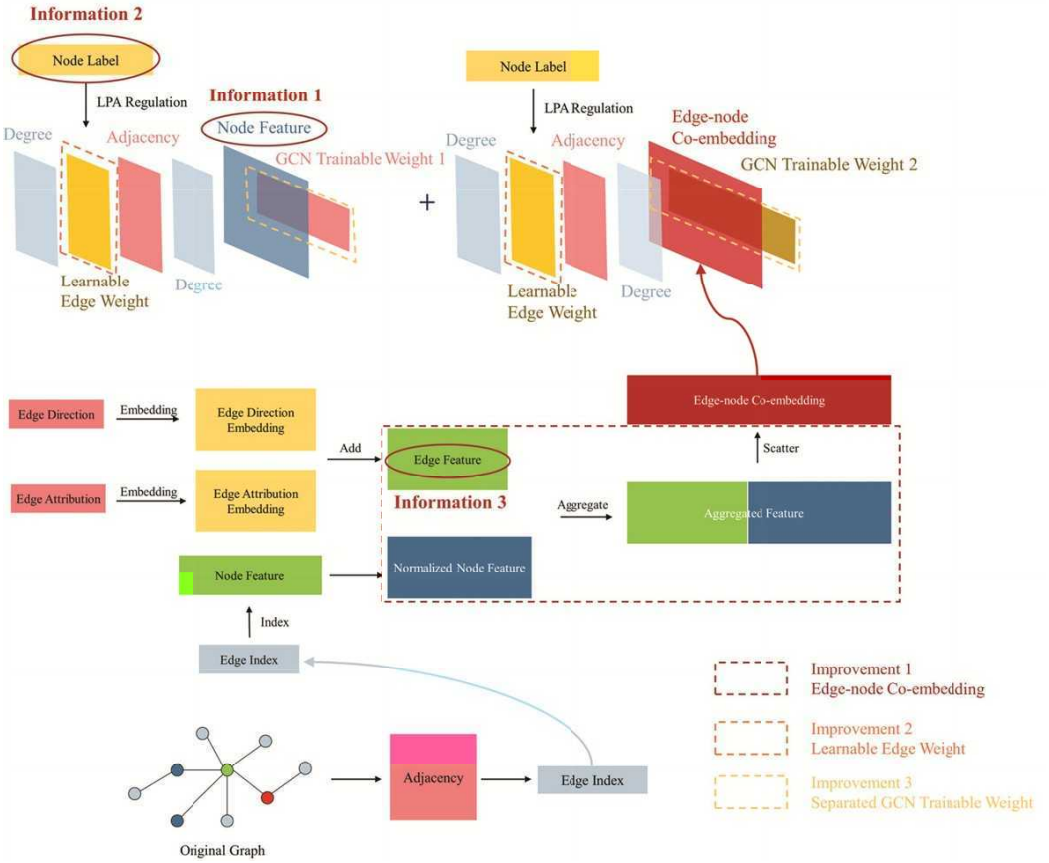


Figure 3. Overall architecture of our unified Graph Convolutional Network (GCN)-based Graph Anomaly Detection (GAD) model.

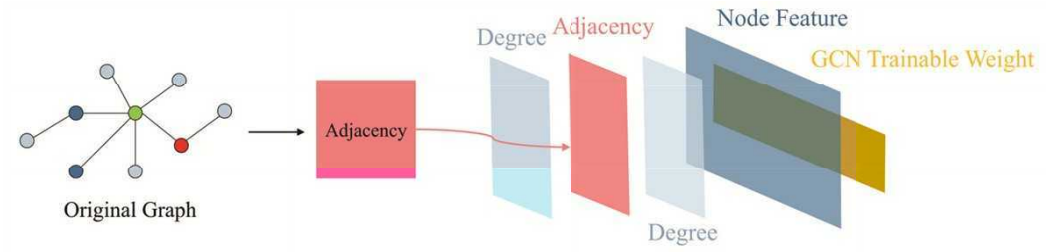


Figure 4. Architecture of classical Graph Convolutional Network (GCN).

4.1. Learnable Edge Weight

To ensure narrative coherence, we first introduce the concept of learnable edge weights.

Building upon traditional GCN networks, as in Equation (4), a learnable edge weight parameter $W^{(l)}$ is introduced. With this addition, the GCN network can be rewritten as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} W^{(l)} \circ \tilde{A}\tilde{D}^{-\frac{1}{2}} H^{(l)}\Theta^{(l)}\right), \tag{13}$$

where $W_{(l)}$ represents the edge weight parameter for layer l, $\circ$ is the symbol of the Hadamard product. It is important to note that the matrix $\tilde{A}$ in the equation is constant and solely dependent on the adjacency relations of the input nodes, with its values remaining unchanged throughout the training process. Θ represents the weights of the graph neural network (not our learnable edge weight). Figure 5 illustrates our first improvement, which is introducing node label information by adding learnable edge weights. The number of columns in the GCN trainable weight layer, denoted as C, is variable. It is set to 2 in the final GCN layer (meeting the binary classification requirements of our GAD task), while in other layers, it is set to the number of hidden units.

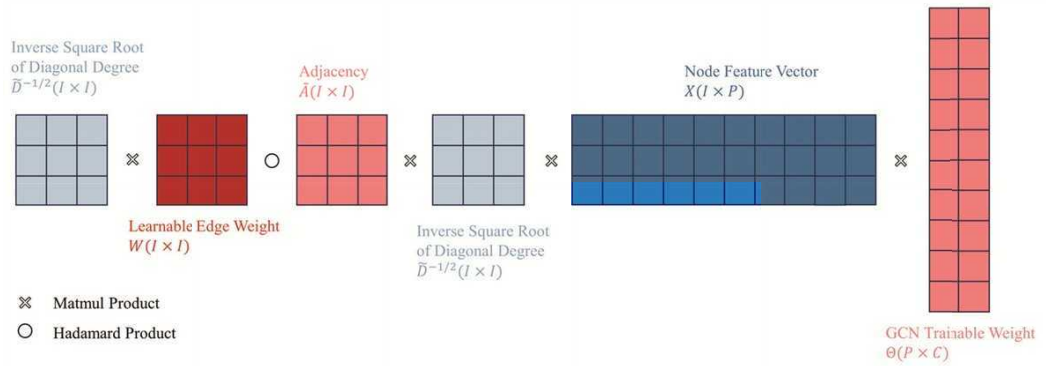


Figure 5. First improvement: Introducing node label information by adding learnable edge weights (only one layer of Graph Convolutional Network (GCN) is displayed).

The next question becomes how to train the weight matrix $W \circ \tilde{A}$ to optimize its performance. Combining Equations (11) and (12), we can conclude that when the loss function of LPA $\mathcal{L}_{LPA}(W \circ \tilde{A})$ is minimized, the optimal edge weight $W \hat{\circ} \tilde{A}$ is achieved, as:

$$W \hat{\circ} \tilde{A} = \arg\min_{W \circ \tilde{A}} \mathcal{L}_{LPA}(W \circ \tilde{A}). \tag{14}$$

Cross entropy can measure the distance between predicted values $\hat{y}_1^{lpa}$ and ground truth y_1. Therefore, the above expression can be rewritten as:

$$W \hat{\circ} \tilde{A} = \arg\min_{W \circ \tilde{A}} \frac{1}{\text{number of labeled nodes}} \sum_{\text{labeled nodes } v_1} J\left(\hat{y}_1^{lpa}, y_1\right), \tag{15}$$

where J is the cross entropy loss.

As a GAD task, as analyzed in Section 1, fraudulent activities exhibit correlations and organizational characteristics. Fraudulent groups often have close connections, and their behaviors show certain similarities. This implies that neighboring nodes in the graph often have the same labels, and their feature vectors are also similar (as the node feature vectors in the GAD task are derived from real user behavior). Given that neighboring nodes have similar labels and feature vectors, cross entropy can effectively encourage adjacent nodes to be similar, learning an effective mapping from feature vectors to labels. Therefore, the loss function for traditional GCN is defined as follows: cross entropy is also used to compute the distance, and the loss of predicted labels by GCN $J\left(\hat{y}_1^{gcn}, y_1\right)$ minimized to learn the optimal transformation matrix Θ^* in the GCN:

$$\Theta^* = \arg\min_{\Theta} \frac{1}{\text{number of labeled nodes}} \sum_{\text{labeled nodes } v_1} J\left(\hat{y}_1^{gcn}, y_1\right). \tag{16}$$

The final question is how to determine the loss function for the unified model. For a graph semi-supervised learning task, we can utilize both node features and graph structure information. Therefore, we can optimize both loss functions together and incorporate L2 regularization [9]:

$$\mathcal{L} = \mathcal{L}_{\text{labeled}} + \lambda \mathcal{L}_{\text{reg}} + \gamma \|\omega\|_2^2, \tag{17}$$

where $\mathcal{L}_{\text{labeled}}$ represents the loss function based on labeled data, $\mathcal{L}_{\text{reg}}$ represents the loss function based on graph structure information, $\|\omega\|_2^2$ is the L2 regularizer, and λ and γ are hyperparameters that respectively adjust the relative importance of $\mathcal{L}_{\text{reg}}$ and $\|\omega\|_2^2$. Data loss $\mathcal{L}_{\text{labeled}}$ considers the distance between the predicted values and the ground truth, which is the supervised loss. As a regularization penalty term, $\mathcal{L}_{\text{reg}}$ only considers the loss induced by the weight coefficients Θ.

As per Section 3.3, it is evident that LPA has the ability to replace GCN in training edge weights. Therefore, the loss of LPA (Equation (15)) can replace the regularization penalty term $\mathcal{L}_{\text{reg}}$ in Equation (17) for training, thereby updating W (Equation (13)), which is the proposed learnable edge weight. The data loss in Equation (17) is composed of the loss of traditional GCN (Equation (16)). Ultimately, the loss function can be written as:

$$\mathcal{L} = \frac{1}{m} \sum_{\text{labeled nodes}} J(\hat{y}^{gcn}, y) + \lambda \left(\frac{1}{m} \sum_{\text{labeled nodes}} J\left(\hat{y}^{lpa}, y\right) \right) + \gamma \|\omega\|_2^2, \tag{18}$$

where m is the number of labeled nodes in the graph. The first term corresponds to the segment of GCN that learns the edge weights W and the trainable matrix Θ. Meanwhile, the second term corresponds to the label propagation segment, which can be seen as imposing constraints on the edge weights W. The second term can be viewed as a regularization on W to assist GCN in learning edge weights.

The above process is implemented using Algorithm 1. First, we initialize a trainable edge weight matrix $W^{(0)}$ as a matrix of ones (line 2). Then, we utilize GCN to perform feature convolution, setting the output dimension of the last GCN layer to 2 for binary classification, obtaining the GCN's label predictions $\hat{y}^{gcn}$ (lines 4–8). The computation of LPA is similar to GCN, resulting in the LPA's label predictions $\hat{y}^{lpa}$ (lines 9–12). Subsequently, the cross-entropy losses between $\hat{y}^{gcn}$ and ground truth y are calculated, as well as $\hat{y}^{gcn}$ and y, to obtain the total loss $\mathcal{L}$ (line 13). Finally, the GCN training weights Θ and the learnable edge weight W are updated using gradient descent.

Algorithm 1 Learnable edge weight.

Input: Graph $G(V, E)$; adjacency matrix A; label matrix Y; degree matrix D;
number of GCN layers L; number of LPA iterations K;
square matrix W, with the size equal to the number of edges in E, where all elements are ones.
Output: Predicted abnormal nodes out.

1: **for** $i = 1$ **to** number of edges in E **do**
2: Set $W^{(0)}[i]$ to 1
3: **end for**
4: **for** $l = 1$ **to** L **do**
5: $H^{(1)} \leftarrow X$
6: $H^{(l)} \leftarrow$ GCN convolution with Equation (4);
7: **end for**
8: $\hat{y}^{gcn} \leftarrow$ Set output dimension of GCN to 2;
9: **for** $k = 1$ **to** K **do**
10: $Y^{(1)} \leftarrow Y$
11: $\hat{y}^{lpa} \leftarrow$ LPA propagation with Equations (1) and (2);
12: **end for**
13: $\mathcal{L} \leftarrow$Calculated loss with Equation (18);
14: Backpropagation;
15: Update Θ and W
16: **return** out

4.2. Co-Embedding of Edge and Node Features

This section will focus on constructing edge embeddings and aggregating them with node embeddings. The amount of information contained in H directly determines the feature influence of a node on its neighboring nodes (as described in Equation (9)). We start by considering increasing the information contained in H. By adding the embeddings of edge direction and properties and generating edge features, we introduce a method to aggregate node features and edge features into a co-embedding feature.

As discussed in Section 3, each edge consists of two elements: edge direction and attribution. Reiterating the definition provided, matrix $E \in \{0,1\}^{1\times c}$ stores the direction of the edges, and matrix $Z \in \{1,\ldots,n\}^{1\times c}$ stores the attribution of the edges.

Once the edge direction and edge attributes are defined as matrices, they can be aggregated using an additive approach for propagation [37]. Although this simple aggregation calculation is computationally efficient, it fails to capture dependencies between different relationships and interactions between node and edge features [38].

To ensure uniform dimensions between node feature matrices and edge feature matrices, while also reflecting the relationship between node and edge features, incorporating edge features into node features is needed. This involves performing aggregation operations on both node features and edge features.

First, we map the high-dimensional discrete features E and Z to continuous low-dimensional vector spaces, obtaining dense vectors, which enables the model to better handle information from E and Z. The embedding layer maps each node with values (both e and z) to a d-dimensional dense vector representation, denoted as E_{emb} and Z_{emb}. The calculation process of the embedding layer can be expressed as:

$$\mathcal{E}[E] = E_{emb}, \tag{19}$$

$$\mathcal{E}[Z] = Z_{emb}, \tag{20}$$

where $\mathcal{E} \in \mathbb{R}^{v\times d}$ is the weight matrix of the embedding layer. For the edge direction matrix and edge attribute matrix, v is 2 and n, respectively, representing the dimensionality of the input space for the embedding layer. Now, we have obtained matrices $E_{emb} \in \mathbb{R}^{c\times d}$ and $Z_{emb} \in \mathbb{R}^{c\times d}$ after the embedding operation, and perform matrix addition to obtain the edge features:

$$F = E_{emb} + Z_{emb}, \tag{21}$$

where $F \in \mathbb{R}^{c\times d}$.

Secondly, we expand the node feature matrix X to $\hat{X}$. We index the node feature vector based on the elements stored in the first row $\mathcal{A}_1$ (source node) of the edge index matrix $\mathcal{A}$. The elements in $\mathcal{A}$ correspond to the row numbers in X. Since the same element value may appear multiple times in $\mathcal{A}$ (because the out-degree of a specific node may be greater than one, i.e., it may act as a source node multiple times), there will be duplicate rows in the extended node feature matrix $\hat{X}$. Each row in the extended node feature matrix $\hat{X}$ is a copy of the corresponding row in the node feature matrix X. For example, if element 2 appears twice in $\mathcal{A}$, then the row corresponding to X_2 will also appear twice in $\hat{X}$. Ultimately, $X \in \mathbb{R}^{I\times P}$ is expanded to $\hat{X} \in \mathbb{R}^{c\times P}$, where c is the number of edges in the graph and P is the dimensionality of the node features.

Thirdly, concatenate F with $\hat{X}$ to obtain the aggregated feature matrix $\hat{F}$, where c is the number of edges in the graph, and $P + d$ is the sum of the dimensions of the edge embedding and the extended node feature.

Finally, $\hat{F}$ is scattered to transform edge features into node features. $\hat{F}$ is set as the input tensor, and scatter indices are specified at target node positions from the second row of $\mathcal{A}$. We specify the output tensor size as the number of nodes (the dimension of adjacency matrix A, denoted by I), and aggregate the data using the sum method. Ultimately, we obtain the co-embedding matrix concat$(H_{node}, H_{edge}) \in \mathbb{R}^{I\times(P+d)}$ of node features and edge features. Figure 6 illustrates the process of generating co-embeddings by embedding edge direction and edge attributes, combining them with extended node features to generate aggregated features, and then scattering to obtain edge-node co-embeddings.

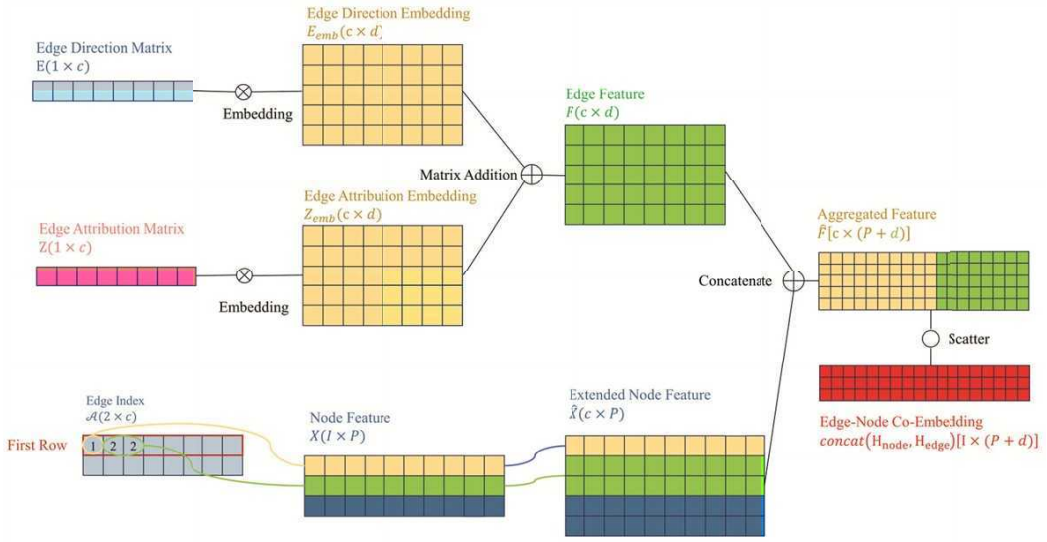


Figure 6. Second improvement: Introducing information from edges through edge-node co-embedding.

4.3. Separate Trainable GCN Weights

In this section, we design separate trainable GCN weights for node features and co-embedding features. The term "separate trainable GCN weights" refers to splitting the original trainable weight Θ in GCN into two weights, Θ_1 and Θ_2. These weights are used to transform node embeddings and co-embeddings in their respective feature spaces before combining them through weighted summation.

In theory, our method extracts high-dimensional information from the neighborhood of graph nodes by dimensionality reduction into dense vector embeddings. These embeddings can be provided to downstream machine learning systems to accomplish the GAD task. In graph analysis tasks, edge embeddings have been proven to be highly useful as input features for GAD [39].

Our discussion still starts with the traditional GCN. For the sake of simplicity in exposition, we rewrite Equation (4) in a more concise form:

$$H'_{node} = H_{node}\Theta. \tag{22}$$

In the above equation, the activation function σ and normalized adjacency matrix $\tilde{S}$ are omitted, and only one layer of GCN is considered, which does not affect the construction of our aggregation function. The equation can be seen as the transformation of node embeddings H to obtain new embeddings H' through trainable weight Θ, where $H^{(0)}_{node} = X$.

To cleverly combine node embedding H_{node} and co-embedding concat(H_{node}, H_{edge}), we designed different Θ for H_{node} and concat(H_{node}, H_{edge}), projecting them into different feature spaces. This helps enhance the model's expressive ability, which can be written as follows:

$$H' = H_{node}\Theta_1 + \phi \cdot \text{concat}(H_{node}, H_{edge})\Theta_2, \tag{23}$$

here, W is the learnable edge weight discussed in Section 4.1, which is incorporated into the edge embedding network, ϕ is a hyperparameter that adjusts the relative importance.

By substituting the co-embeddings into Equation (23), Θ_1 and Θ_2 can be trained separately. Figure 7 illustrates our third improvement, which is designing separate GCN trainable weights to enhance the model's expressive power, and Algorithm 2 outlines the process of our co-embedding and separate training algorithm. First, we perform embedding operations separately on the directional and attribute information of the edges (lines 1–2). Then, we add the obtained embedding matrices to obtain the edge feature matrix (line 3). We redistribute the node feature matrix to obtain the extended node feature matrix (line 4). Next, we aggregate the edge feature matrix and the node feature matrix (line 5), and obtain the edge-node co-embedding through scatter operations (line 6). Finally, we project node

features and edge-node co-embedding into different spaces using different GCN trainable weights (line 7).

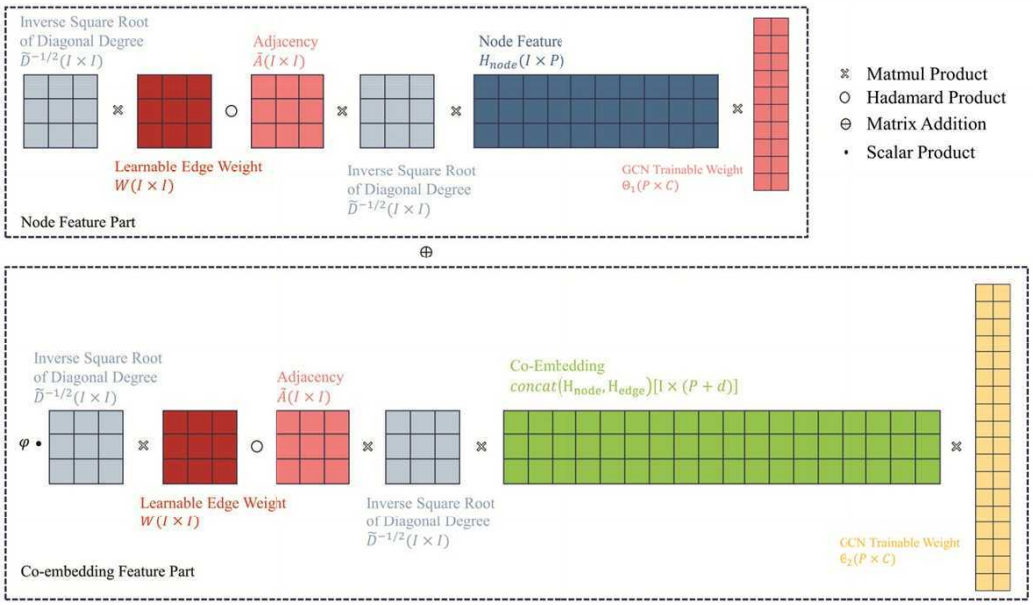


Figure 7. Third improvement: Designing separate Graph Convolutional Network (GCN) trainable weights to enhance the model's expressive power (only one layer of GCN is displayed).

Algorithm 2 Node-edge co-embedding and separate training.

Input: Graph G; edge direction E; edge attribution Z; edge index $\mathcal{A}$; node feature X.
Output: Predicted Abnormal Nodes out.

1: $E_{emb} \leftarrow$ Embed edge direction with Equation (19);
2: $Z_{emb} \leftarrow$ Embed edge attribution with Equation (20);
3: $F \leftarrow E_{emb} + Z_{emb}$;
4: $\hat{X} \leftarrow$ Redistribute the rows of X according to $\mathcal{A}_1$;
5: $\hat{F} \leftarrow concat(F, \hat{X})$;
6: $concat(H_{node}, H_{edge}) \leftarrow scatter(\hat{F})$;
7: Convolute with Equation (23) using Algorithm 1
8: **return** out

5. Experiments

5.1. Datasets

We selected the publicly available dataset, DGraph [6], a real-world GAD graph in the finance domain, as our test sample. This dataset comprises a directed, unweighted graph that represents a social network among users. In this graph, each node represents a user, and the node features, derived from basic personal profiles, consist of a 17-dimensional vector. Each dimension corresponds to a different element of the individual's profile, primarily including the following aspects: (1) basic personal information, such as age, gender, and location; (2) past borrowing behavior, including agreed-upon repayment due dates and actual repayment dates; (3) emergency contact-related information, such as the existence of emergency contacts. An edge from one user to another signifies a different type of connection, with a total of 12 distinct types of connections. It is noteworthy that DGraph is highly sparse, containing approximately 3 million nodes and 4 million edges. The number of nodes in DGraph is 17.1 times that of Elliptic [40], making it the largest publicly available dataset in the GAD domain to date.

As a financial dataset, DGraph originates from real-world borrowing and lending behaviors. The determination of whether a node is classified as normal or anomalous depends on its borrowing and lending activities, summarized briefly as follows. All labeled

nodes in DGraph have engaged in borrowing or lending at least once. Nodes that are registered but have not engaged in borrowing or lending are classified as unlabeled nodes. Among the labeled nodes, those classified as anomalous are users who have failed to repay their loans for a considerable period after the loan's due date, disregarding repeated reminders from the platform. According to the aforementioned criteria, 15,509 users are classified as anomalous. Among these nodes, normal users account for 22.07% (nodes labeled as 0), while abnormal users account for 0.26%. The ratio of positive to negative samples is approximately 85:1, indicating a significant class imbalance in the dataset. The nodes are randomly split into training/validation/test sets with a ratio of 70:15:15. Table 3 lists the key parameters of DGraph.

Table 3. Key parameters of DGraph.

Parameters	Value
# Nodes	3,700,550
# Node features	17-dimensional anonymous features
# Edges	4,300,999
Edge information	edge direction, edge attribution
Normal-to-anomaly ratio	85:1
Proportion of missing feature values	56.37%
-for normal nodes	39%
-for anomaly nodes	67%
Average node degree	1.22

Table 4 summarizes the comparison of parameters between DGraph and mainstream anomaly detection datasets. Compared to other datasets, DGraph is closer to real-world applications because it has a minimum percentage of anomalous nodes and a diluted network, which is a common phenomenon in real GAD task. At the same time, the label data proportion in DGraph is also very low. Compared to DGraph, YelpChi is constructed using a Yelp Review Filter [41], making it not the real ground truth. The ratio of edges to nodes (in other words, the degree of the nodes) in the Amazon dataset is approximately 368:1, creating a graph that is too dense to be common in the GAD field. Elliptic is the closest to our scenario, but it treats edges as homogeneous, meaning that edges do not have attributes. In real GAD scenarios, however, the connections between users are diverse. Excessive features also reduce the usability of Elliptic. This is the reason why DGraph was chosen as the dataset for this study.

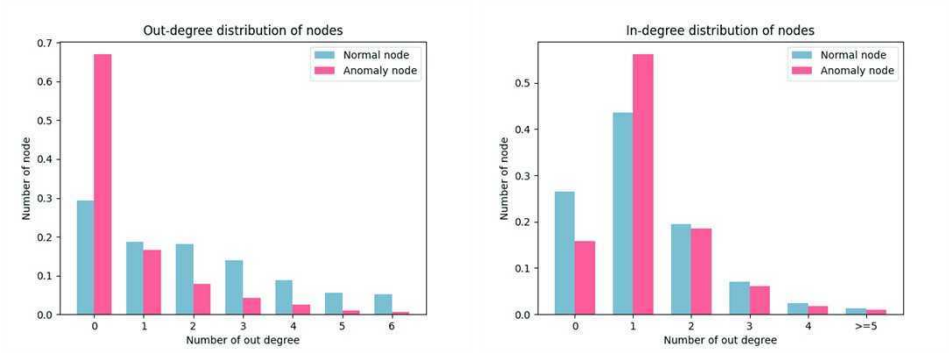
Table 4. Comparison of parameters between DGraph and mainstream Graph Anomaly Detection (GAD) datasets.

Dateset	# Nodes	# Edges	# Labeled Nodes	# Anomalous Nodes	# Features
YelpChi [2]	45954	3846979	45954	14.50%	32
Amazon [2]	11944	4398392	11944	9.50%	25
Elliptic [40]	203769	234355	46564	9.80%	166
DGraph [6]	3700550	4300999	1225601	1.17%	17

It is important to note that, for privacy protection purposes, the features of the nodes in DGraph are anonymously provided and not ordered by importance. This anonymity of features prevents this paper from visualizing the dataset in the same manner as other datasets, where the first two features are typically chosen for the x-axis and y-axis in plots. The anonymous provision of features also limits our ability to further explain which features contribute more significantly to node anomalies. (The DGraph provider anonymizes users by removing personal identifiers and randomizing transaction details, making it impossible to trace back to user IDs. Furthermore, since user features are not unique, users cannot be identified from the dataset, ensuring DGraph's anonymity.)

Figure 8 lists the distribution of node out-degrees and in-degrees. There is a significant difference in the distributions of out-degrees and in-degrees for the two types of nodes.

According to the definition of DGraph, each node represents a user. The out-degree represents the completeness of information for a user, where the higher out-degree indicates higher information completeness. The in-degree represents the trust relationship of a user, where a higher in-degree indicates higher trust from other users. The out-degree of a node is determined by the user, while the in-degree is not. The significant difference between node out-degrees and in-degrees provides a basis for our edge direction embedding. The edge direction treats the out-degree and in-degree of nodes differently, accurately distinguishing between the two types of nodes.



(**a**) (**b**)

Figure 8. Distribution of node out-degrees and in-degrees. (**a**) Out-degree distribution of nodes. (**b**) In-degree distribution of nodes.

Background nodes are an important attribute of the DGraph dataset. [6] These nodes refer to entities that lack borrowing and lending activities, are not the target of anomaly detection, but play a crucial role in maintaining the connectivity of DGraph. Throughout our experiments, we adhere to the following assumption: retain all background nodes but only perform anomaly detection on labeled nodes (normal or anomalous).

5.2. Baselines

We have selected eight methods as our baseline approaches, comprising one baseline method (MLP), two generic graph methods (Node2Vec and LPA), and five GCN-based methods (GCN, GAT, GATv2, GraphSAGE, and UniMP). We adopt AUC as the evaluation metric, which is extremely useful in the scenario of data imbalance where abnormal data sets are scarce while normal data sets are extremely abundant. The higher the AUC value, the better the model's performance. For example, in DGraph, abnormal users only account for 1.16% of the dataset. Therefore, even if the model predicts all samples as normal, the accuracy would still be as high as 99%. However, it is evident that the model's predictive performance is poor. Moreover, in GAD scenarios, the losses incurred by predicting fraudulent users as normal would be far greater than predicting normal users as fraudulent. Hence, for our GAD task, AUC is more crucial.

5.3. Experimental Setup

The experimental environment for all experiments remained consistent. The GPU utilized was an NVIDIA V100 (32 GB NVLink memory, single GPU with 5120 CUDA Cores), the CPU consisted of 12 Intel Xeon processors (Platinum 8163 (Skylake) @2.50 GHz), and the system had 92 GB of RAM. For each method, five runs were executed, with each run comprising 500 epochs, and the best-performing value was selected. Unless otherwise specified, all experiments are conducted on undirected graphs.

In implementing all algorithms, we utilized PyTorch Geometric. During the implementation process, we employed one trick: we replaced the storage of the matrix containing the source and target nodes of edges with a sparse adjacency matrix. This significantly accelerated computation speed, except for GAT, GATv2, and co-embedding, as they require explicit edge propagation.

All the hyper-parameters are listed below:

- Hidden layer size: 40;
- GCN layers: 2;
- LPA iteration: 5;
- Edge direction and attribution embedding size: 10;
- L2 weight: 5×10^{-5};
- Dropout rate: 0.3;
- Learning rate: 0.05.

5.4. Comparison with Baselines

Table 5 presents the results of the comparison between our unified model and the baseline algorithms in terms of the AUC metric. The hyperparameters used here are: LPA weight $\lambda = 1$ and co-embedding weight $\phi = 1$. In the table, "WL" refers to the first improvement module, which is the edge weight learnable module in our model; "CE" refers to the second improvement module, which is the edge-node co-embedding module.

Table 5. Comparison between our unified model and the baseline algorithms (with $\lambda = 1$, $\phi = 1$). T.T. means average training time per epoch.

	Methods	**AUC**		**T.T.**
		Valid	**Test**	
Baseline	MLP	71.74%	72.31%	0.412 s
Generic Graph Methods	LPA [8]	50.89%	50.95%	N/A
	Node2Vec [7]	62.61%	62.92%	N/A
GCN-based Methods	GCN [9]	70.78%	72.32%	0.608 s
	GAT [10]	72.33%	73.31%	0.798 s
	GATv2 [11] (ICLR 2022)	75.26%	76.24%	0.901 s
	GraphSAGE [12]	75.48%	76.21%	0.671 s
	UniMP [13] (IJCAI 2021)	77.17%	78.38%	0.641 s
Our model	GCN-WL	73.01%	73.82%	0.625 s
	GCN-WL+CE	78.52%	79.45%	0.998 s

From Table 5, we observe that the MLP baseline, which does not utilize any graph information, achieved good results. Among the generic graph methods, Node2Vec, which only utilizes graph structure, and LPA, which only utilizes node labels, did not achieve high AUC scores. This suggests that in the GAD task on DGraph, the primary information lies within the node features. Among the GCN-based models, classical GCN and GAT performed similarly, while GraphSAGE showed significant improvement in performance. This is because GraphSAGE designs different trainable features for neighboring nodes. GATv2 improves upon GAT and exhibits noticeable performance gains. UniMP, incorporating attention mechanisms into GCN models, achieved very significant improvements.

Our edge-weight-learnable method, which is an improvement of GCN utilizing LPA to incorporate node labels, achieved higher AUC compared to traditional GCN. This demonstrates that both node features and node labels contain rich information, and LPA can act as a regularization penalty to aid training.

In terms of model training time, the general graph methods LPA and Node2Vec do not involve explicit "training" and therefore do not have training time. The MLP model, being the smallest in scale and simplest in architecture, naturally has the shortest training time. Among the methods employing the sparse matrix trick, our edge weight learnable method has the second shortest training time, only slightly longer than the classical GCN method. This is because our edge weight learnable method does not increase the model's dimensionality; it only introduces node labels (which are also used in traditional GCN-based methods) and achieves a good trade-off between time and performance. Methods that do not use the sparse matrix trick—such as GAT, GATv2, and our co-embedding method—require significantly longer training times. This is because the edge information needs to be explicitly propagated along the edges and cannot be represented using sparse

matrices to reduce computation. Additionally, extracting edge features and the growth in model size both require more time. It is worth noting that both GraphSAGE and GAT have corresponding mini-batch methods, which can reduce the size of each batch to improve speed. However, for the sake of fairness, we did not use mini-batch training in this study.

The method incorporating edge-node co-embedding and improving the convolution operator achieved significant progress in performance. This indicates that the direction and attributes of edges are helpful in enhancing the ability of GAD to detect anomalies. Additionally, designing different trainable weights for node features and co-embedding features contributes to improving algorithm expressiveness.

5.5. Infuence of LPA Regularization

To study the effectiveness of LPA regularization, we varied the LPA weight λ from 0 to 5 in the weight-learnable algorithm, without adding co-embedding. We observed the performance changes, as shown in Table 6.

Table 6. The variation of Area Under the Curve (AUC) values due to different Label Propagation Algorithm (LPA) weights λ (without incorporating co-embedding).

λ	**0 (Classical GCN)**	**0.01**	**0.1**	**0.5**	**1**	**5**
AUC	72.32%	72.53%	72.99%	73.26%	73.82%	73.24%

From the table, it can be observed that when λ is non-zero (i.e., after adding learnable weight), the performance is better than before adding learnable weight. This demonstrates the effectiveness of our LPA regularization strategy. However, when λ is too large, the term $\mathcal{L}_{\text{reg}}$ in Equation (17) becomes greater than the term $\mathcal{L}_{\text{labeled}}$. In the computation of the loss, the loss function based on labeled data $\mathcal{L}_{\text{labeled}}$ is more important, as it determines the direction of gradient descent, while the regularization penalty $\mathcal{L}_{\text{reg}}$ serves as an auxiliary factor.

5.6. Impact of Edge and Node Co-Embedding

To investigate the influence of co-embedding, we fixed λ to 1 and varied the weight ϕ of the co-embedding term from 0 to 2, observing the performance changes. Table 7 reflects this variation. It can be seen from the table that the inclusion of edge directionality and attribute information leads to significant performance improvements, despite the fact that there are only two types of directional information and 12 types of edge attribute information. This demonstrates the effectiveness of our co-embedding method. Additionally, designing different trainable weights for node features and co-embedding features, projecting different features into different feature spaces, greatly enhances the model's expressive ability. However, overly large ϕ values can lead to performance degradation. We believe this is because most of the information in the graph is contained within the node features, and the fundamental purpose of the GCN algorithm is to find neighboring nodes based on the similarity of node features, with edge information serving as an auxiliary.

Including reverse edges can increase the density of edges and provide more directional information, which can greatly improve performance when edge information is required for co-embedding. To demonstrate this, we fixed λ to 1 and tested the performance under the same conditions on a directed graph (i.e., without adding reverse edges). Table 7 reflects this variation. It can be observed that as the co-embedding weight ϕ increases, the performance gain from reverse edges decreases. We believe that when the co-embedding weight is low, the information contained in the graph's topological structure is more prominent, while undirected graphs contain more topological structures. However, as the co-embedding weight increases, the directional and attribute information carried by the edges becomes more important.

Table 7. The variation of Area Under the Curve (AUC) values due to different co-embedding weights ϕ on directed and undirected graphs (with Label Propagation Algorithm (LPA) weights $\lambda = 1$).

ϕ		0 (Just WL, No CE)	0.01	0.1	0.5	1	2
AUC	**Directed**	66.51%	72.90%	75.31%	76.77%	77.37%	76.69%
	Undirected	73.82%	74.54%	77.26%	79.30%	79.45%	73.23%

To demonstrate the effectiveness of the learnable edge weight strategy, we fixed ϕ to 1 and removed the learnable edge weight on an undirected graph; the AUC decreased from 79.45% to 79.03%. This demonstrates that although the second term concat(H_{node}, H_{edge}) in Equation (23) is composed of node features H_{node} and edge features H_{edge} co-embedded, it is still a feature matrix aggregated based on node dimensions. Therefore, using the loss term from the LPA algorithm instead of the regularization penalty in GCN, as derived in Equation (18), still improves the algorithm's performance.

6. Conclusions

This research contributes to advancing the field of graph anomaly detection, offering a robust framework for future studies and applications.

Firstly, by establishing the relationship between label propagation and feature convolution, we have shown that the LPA can function as a regularization penalty term for GCN, thereby facilitating training and enabling the learning of edge weights. Secondly, our node and edge feature co-embedding method fully utilizes the directional and attribute information inherent in the edges, increasing the spatial dimensionality of the model. Finally, we employ separate trainable GCN weights to handle node features and co-embedding features, enabling the projection of different features into separate feature spaces, greatly enhancing the model's generalization capability.

Experimental results on the DGraph dataset have demonstrated superior AUC performance compared to baseline models, underscoring the feasibility and effectiveness of our proposed approach in tackling GAD tasks.

The method proposed in this paper currently incurs relatively high time and memory costs, requiring significant computational resources. This limitation hinders the application of the proposed method in larger-scale scenarios. Future work includes exploring techniques such as downsampling to reduce time and memory overhead while maintaining algorithm performance, making it more responsive to the rapid growth of data brought about by the Industry 4.0 era.

Author Contributions: Conceptualization, J.Y. and X.T.; methodology, X.T. and Z.Z.; validation, X.T.; formal analysis, X.T.; investigation, X.T.; resources, J.Y. and J.X.; data curation, X.T. and Z.Z.; writing—original draft preparation, X.T.; writing—review and editing, J.Y., Z.Z., and J.X.; visualization, X.T. and C.L.; project administration, J.Y.; funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Institute of Health Insurance Research grant number YB2021B12.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The DGraph dataset is a publicly available dataset.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LPA	Label Propagation Algorithm
GCN	Graph Convolutional Network
MLP	Multilayer Perceptron
GAT	Graph Attention Network
GNN	Graph Neural Network
GAD	Graph Anomaly Detection
ROC	Receiver Operating Characteristic Curve
AUC	Area Under ROC Curve

References

1. Dou, Y.; Shu, K.; Xia, C.; Yu, P.S.; Sun, L. User Preference-aware Fake News Detection. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Online, 11–15 July 2021. [CrossRef]
2. Dou, Y.; Liu, Z.; Sun, L.; Deng, Y.; Peng, H.; Yu, P.S. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Online, 19–23 October 2020; pp. 315–324. [CrossRef]
3. Akoglu, L.; Tong, H.; Koutra, D. Graph based anomaly detection and description: A survey. *Data Min. Knowl. Discov.* **2015**, *29*, 626–688. [CrossRef]
4. Ma, X.; Wu, J.; Xue, S.; Yang, J.; Zhou, C.; Sheng, Q.Z.; Xiong, H.; Akoglu, L. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 12012–12038. [CrossRef]
5. Ren, J.; Xia, F.; Lee, I.; Noori Hoshyar, A.; Aggarwal, C. Graph learning for anomaly analytics: Algorithms, applications, and challenges. *ACM Trans. Intell. Syst. Technol.* **2023**, *14*, 28. [CrossRef]
6. Huang, X.; Yang, Y.; Wang, Y.; Wang, C.; Zhang, Z.; Xu, J.; Chen, L.; Vazirgiannis, M. DGraph: A Large-Scale Financial Dataset for Graph Anomaly Detection. In Proceedings of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, 28 November–9 December 2022; pp. 22765–22777. [CrossRef]
7. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864. [CrossRef]
8. Zhu, X.; Ghahramani, Z. Learning from Labeled and Unlabeled Data with Label Propagation. 2002. Available online: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8a6a114d699824b678325766be195b0e7b564705 (accessed on 2 March 2024).
9. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017. [CrossRef]
10. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018. [CrossRef]
11. Brody, S.; Alon, U.; Yahav, E. How Attentive are Graph Attention Networks? In Proceedings of the International Conference on Learning Representations, Virtual, Austria, 3–7 May 2021. [CrossRef]
12. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.
13. Shi, Y.; Huang, Z.; Feng, S.; Zhong, H.; Wang, W.; Sun, Y. Masked label prediction: Unified message passing model for semi-supervised classification. In Proceedings of the 30th International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 19–27 August 2021; pp. 1548–1554. [CrossRef]
14. Zhu, X.; Goldberg, A.B. *Introduction to Semi-Supervised Learning*; Springer Nature: Berlin/Heidelberg, Germany, 2022.
15. Chong, Y.; Ding, Y.; Yan, Q.; Pan, S. Graph-based semi-supervised learning: A review. *Neurocomputing* **2020**, *408*, 216–230. [CrossRef]
16. Van Engelen, J.E.; Hoos, H.H. A survey on semi-supervised learning. *Mach. Learn.* **2020**, *109*, 373–440. [CrossRef]
17. Zhou, Z.H.; Zhou, Z.H. Semi-supervised learning. In *Machine Learning*; Springer: Singapore, 2021; pp. 315–341. [CrossRef]
18. Lebichot, B.; Braun, F.; Caelen, O.; Saerens, M. A graph-based, semi-supervised, credit card fraud detection system. In *Complex Networks & Their Applications V, Proceedings of the International Workshop on Complex Networks and Their Applications, Milan, Italy, 30 November–2 December 2016*; Springer: Cham, Switzerland, 2016; pp. 721–733. [CrossRef]
19. Huang, Q.; He, H.; Singh, A.; Lim, S.N.; Benson, A. Combining Label Propagation and Simple Models out-performs Graph Neural Networks. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020. [CrossRef]
20. Li, Z.; Chen, X.; Song, J.; Gao, J. Adaptive label propagation for group anomaly detection in large-scale networks. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 12053–12067. [CrossRef]

21. Wang, H.; Li, Z.; Huang, J.; Hui, P.; Liu, W.; Hu, T.; Chen, G. Collaboration based multi-label propagation for fraud detection. In Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 2477–2483. [CrossRef]
22. Wang, J.; Guo, Y.; Wen, X.; Wang, Z.; Li, Z.; Tang, M. Improving graph-based label propagation algorithm with group partition for fraud detection. *Appl. Intell.* **2020**, *50*, 3291–3300. [CrossRef]
23. Zhang, J.; Yang, F.; Lin, K.; Lai, Y. Hierarchical Multi-Modal Fusion on Dynamic Heterogeneous Graph for Health Insurance Fraud Detection. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; pp. 1–6. [CrossRef]
24. Liu, F.; Ma, X.; Wu, J.; Yang, J.; Xue, S.; Beheshti, A.; Zhou, C.; Peng, H.; Sheng, Q.Z.; Aggarwal, C.C. DAGAD: Data augmentation for graph anomaly detection. In Proceedings of the IEEE International Conference on Data Mining (ICDM), Orlando, FL, USA, 28 November–1 December 2022; pp. 259–268. [CrossRef]
25. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous graph attention network. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2022–2032. [CrossRef]
26. Pourhabibi, T.; Ong, K.L.; Kam, B.H.; Boo, Y.L. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decis. Support Syst.* **2020**, *133*, 113303. [CrossRef]
27. Ding, K.; Li, J.; Bhanushali, R.; Liu, H. Deep anomaly detection on attributed networks. In Proceedings of the SIAM International Conference on Data Mining. SIAM, Calgary, AB, Canada, 2–4 May 2019; pp. 594–602. [CrossRef]
28. Liu, Z.; Chen, C.; Yang, X.; Zhou, J.; Li, X.; Song, L. Heterogeneous graph neural networks for malicious account detection. In Proceedings of the 27th ACM International Conference on Information & Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 2077–2085. [CrossRef]
29. Wu, B.; Chao, K.M.; Li, Y. DualFraud: Dual-Target Fraud Detection and Explanation in Supply Chain Finance Across Heterogeneous Graphs. In *Database Systems for Advanced Applications, Proceedings of the International Conference on Database Systems for Advanced Applications, Tianjin, China, 17–20 April 2023*; Springer: Cham, Switzerland, 2023; pp. 370–379. [CrossRef]
30. Wang, H.; Leskovec, J. Combining graph convolutional neural networks and label propagation. *ACM Trans. Inf. Syst. (TOIS)* **2021**, *40*, 73. [CrossRef]
31. Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W.L.; Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 974–983. [CrossRef]
32. Zhang, Y.; Yuan, M.; Zhao, C.; Chen, M.; Liu, X. Integrating label propagation with graph convolutional networks for recommendation. *Neural Comput. Appl.* **2022**, *34*, 8211–8225. [CrossRef]
33. Li, Z.; Wang, B.; Huang, J.; Jin, Y.; Xu, Z.; Zhang, J.; Gao, J. A graph-powered large-scale fraud detection system. *Int. J. Mach. Learn. Cybern.* **2024**, *15*, 115–128. [CrossRef]
34. Wang, H.; Zhang, F.; Zhang, M.; Leskovec, J.; Zhao, M.; Li, W.; Wang, Z. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 968–977. [CrossRef]
35. Koh, P.W.; Liang, P. Understanding black-box predictions via influence functions. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1885–1894.
36. Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.I.; Jegelka, S. Representation learning on graphs with jumping knowledge networks. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5453–5462. [CrossRef]
37. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Greece, 3–7 June 2018; Proceedings 15; Springer: Berlin/Heidelberg, Germany, 2018; pp. 593–607. [CrossRef]
38. Jiang, X.; Zhu, R.; Li, S.; Ji, P. Co-embedding of nodes and edges with graph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *45*, 7075–7086. [CrossRef] [PubMed]
39. Wang, D.; Cui, P.; Zhu, W. Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1225–1234. [CrossRef]
40. Weber, M.; Domeniconi, G.; Chen, J.; Weidele, D.K.I.; Bellei, C.; Robinson, T.; Leiserson, C.E. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv* **2019**, arXiv:1908.02591. [CrossRef].
41. Mukherjee, A.; Venkataraman, V.; Liu, B.; Glance, N. What yelp fake review filter might be doing? In Proceedings of the International AAAI Conference on Web and Social Media, Cambridge, MA USA, 8–11 July 2013; Volume 7, pp. 409–418. [CrossRef].

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

remote sensing

MDPI

Article

Elimination of Irregular Boundaries and Seams for UAV Image Stitching with a Diffusion Model

Jun Chen [1,2,3,*,†], Yongxi Luo [4,†], Jie Wang [4,†], Honghua Tang [4], Yixian Tang [3] and Jianhui Li [1]

[1] Computer Network Information Center, Chinese Academy of Sciences, Beijing 100083, China; lijh@cnic.cn
[2] School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[3] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; tangyx@aircas.ac.cn
[4] School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China; 2112130009@e.gzhu.edu.cn (Y.L.); wangjie@gzhu.edu.cn (J.W.); 2112230031@e.gzhu.edu.cn (H.T.)
* Correspondence: chenjun@aircas.ac.cn; Tel.: +86-15810430728
† These authors contributed equally to this work.

Abstract: Unmanned aerial vehicle (UAV) image stitching refers to the process of combining multiple UAV images into a single large-format, wide-field image, and the stitched image often contains large irregular boundaries and multiple stitching seams. Usually, irregular boundaries are addressed using grid-constrained methods, while seams are optimized through the design of energy functions and penalty terms applied to the pixels at the seams. The above-mentioned two solutions can only address one of the two issues individually and are often limited to pairwise stitching of images. To the best of our knowledge, there is no unified approach that can handle both seams and irregular boundaries in the context of multi-image stitching for UAV images. Considering that addressing irregular boundaries involves completing missing information for regularization and that mitigating seams involves generating images near the stitching seams, both of these challenges can be viewed as instances of a mask-based image completion problem. This paper proposes a UAV image stitching method based on a diffusion model. This method uniformly designs masks for irregular boundaries and stitching seams, and the unconditional score function of the diffusion model is then utilized to reverse the process. Additional manifold gradient constraints are applied to restore masked images, eliminating both irregular boundaries and stitching seams and resulting in higher perceptual quality. The restoration maintains high consistency in texture and semantics. This method not only simultaneously addresses irregular boundaries and stitching seams but also is unaffected by factors such as the number of stitched images, the shape of irregular boundaries, and the distribution of stitching seams, demonstrating its robustness.

Keywords: UAV image; diffusion model; image stitching; irregular boundaries; stitching seams

1. Introduction

To obtain images with a broader field of view, image stitching is a necessary pre-processing step in unmanned aerial vehicle (UAV) remote sensing applications [1]. The common issues involved in stitched images include irregular boundaries and stitching seams due to the inability to achieve perfect alignment of multiple images. For UAV image stitching, accumulations of stitching seams and irregular boundaries often occur because more than two images are required to be stitched through multiple stitching processes, and these two phenonmena can be clearly seen in Figure 1.

Existing image stitching methods often optimize global or local geometric distortions to enhance alignment between different images, while irregular boundaries and stitching seams still exist [2–5]. For the UAV image stitching task, Guo et al. [6] combined shape-preserving with global alignment, achieving higher alignment accuracy when compared

with many advanced methods. Cui et al. [7] proposed a strategy to fully utilize features extracted by transformers for point matching, which can reduce positional errors when stitching images containing inconspicuous features, such as forests, bare land, and rivers, thus achieving promising performance. Lin et al. [8] introduced an image stitching method based on Vector Shape Preserving (VSP) deformation, which can achieve high-precision alignment. Although the above-mentioned methods have achieved much better image stitching performance, they still cannot ensure smooth transitions between the overlapping image regions, and they also suffer irregular boundaries. Recently, methods have been proposed to address irregular boundaries such as rectangular cropping of stitched images, resulting in significant loss of field of view. Some pioneer researchers employed the two-stage grid deformation method based on grid search and grid optimization [9–11]. Although this type of method can preserve linear structures, they often introduce serious distortions in nonlinear structures.

To tackle the irregular boundary problem, an intuitive method is the use of image completion, which requires the design of masks to guide the completion of missing regions. This idea has not been adopted for this task up to now. For stitching seams, there are two main types of methods. One type of method involves designing various energy functions and introducing penalty terms to achieve a natural transition of the overlap region's boundaries [12–17]. The other type of method defines a seam loss, aiming to find the optimal image alignment transformation by minimizing this loss [18–21]. These methods have strict requirements on the geometric features of the images, and a lack of sufficient geometric features may seriously degrade their performance and even lead to failure in some cases.

To the best of our knowledge, there is no unified method that can simultaneously address irregular boundaries and stitching seams. If one wants to eliminate both stitching seams and irregular boundaries, an intuitive and straightforward method is to handle the two problems in parallel or in series. This cannot guarantee the optimal performance because these two problems are solved independently without joint optimization. After considering that irregular boundaries can be solved by image completion and that seams can be eliminated by generating images near the stitching seams, these two problems can be unified into one problem, i.e., the mask-based image completion problem. Accordingly, this paper proposes a unified method that treats both irregular boundaries and stitching seams as image completion problems. The proposed method is expected to prevent the introduction of cumulative errors, achieve smooth transitions in overlapping regions, and generate completely rectangular images.

The goal of image completion is to fill in missing areas of an image. These repaired regions need to coordinate consistently with other parts of the image and maintain semantic consistency; thus, the image completion task often requires a powerful generative model. Inspired by non-equilibrium thermodynamics in deep unsupervised learning [22], a generative model called the diffusion model [23,24] is introduced, and Dhariwal et al. [25] have demonstrated its superior image generation capabilities compared with many state-of-the-art GAN (Generative Adversarial Network)-based methods. Most existing image completion methods focus on training for specific mask distributions, limiting their generalization ability to different tasks such as image stitching for UAV images. Some pioneering studies [26–28] have shown the impressive performance of the diffusion model in mask-based image restoration tasks.

This study provides the first attempt to adopt the diffusion model into the image stitching task to address both irregular boundaries and stitching seams. There exist two main steps in the proposed method. Firstly, we generate a unified mask for irregular boundaries and stitching seams when stitching multiple images. Secondly, during the inverse process of the trained diffusion model, we introduce the constraint of manifold gradients. This method utilizes the unconditional score function and applies additional constraints to achieve high-quality image restoration with good generalization capabilities [28]. Additionally, when considering that the diffusion model can only handle fixed-size input–output

situations and that the size of stitched images is often uncertain, we further introduce the local implicit image function (LIIF) method proposed by Chen et al. [29] to address these issues. By doing so, the proposed method can represent images continuously to obtain results for arbitrary resolutions, and we can then adjust the size of the stitched images to match the diffusion model in an arbitrary way. Compared with other state-of-the-art methods,the proposed method treats stitching seams and irregular boundaries as one unified problem. This allows us to handle irregular boundaries and seams of arbitrary distribution, and the final restoration quality is less affected by low-level semantic features such as geometric structures in the image restoration region, indicating the more versatile and robust performance of the proposed method when compared with many state-of-the-art methods.

Experimental results show that the proposed method can effectively eliminate stitching seams and irregular boundaries, improving the visual effects of the stitched images and demonstrating promising generalization. Our contributions are summarized as follows: Firstly, a specially designed mask for irregular boundaries and stitching seams is proposed for the diffusion model to improve the quality of stitched UAV images. Secondly, by designing suitable masks, we treat the two problems of stitching seams and irregular boundaries as one unified problem, avoiding the use of two different methods to separately address these two problems, thereby reducing potential uncertainties and cumulative errors.



Figure 1. Irregular boundaries and stitching seams are present in the stitched images of the unmanned aerial vehicle. In the (**a**,**b**) results, the areas enclosed by the red and yellow boxes, respectively, represent portions of stitching seams and irregular boundaries.

2. Related Work

2.1. Image Rectangling and Seam Cutting

The pioneering method for obtaining rectangular stitched images involves optimizing linear grid deformation [9], while its energy function can only preserve linear structures. Considering the potential existence of nonlinear structures in image stitching, Li et al. [10] later refined the energy term from preserving straight lines to preserving geodesic lines. It is well known that geodesic lines are difficult to directly detect from stitched images; thus, its application in image stitching is limited. Zhang et al. [11] integrated rectification and image stitching into a unified optimization process, aiming to reduce distortion in the stitched rectangular images. They relaxed the constraints on rectangular shapes and adopted segmented rectangular boundaries. Nie et al. [30] proposed a grid-based deep learning method for rectification, establishing probably the first deep learning approach and achieving state-of-the-art performance. However, for UAV image stitching scenarios, the above-mentioned methods still cannot fully rectify the stitched images and fail to solve the unnatural transitions caused by imperfect alignment as depicted in Figure 2b.

Typically, seam cutting is regarded as a post-processing step in image stitching. Traditional methods can be categorized into two types: One involves introducing penalty terms by defining a generalized energy function to handle seams. These penalty terms include

Euclidean color difference [12], gradient difference [15,16], motion and exposure-aware differences [17], saliency differences, and so on. By minimizing these energy functions, seam fusion is performed at the seams using graph-cut optimization. The second type defines seam losses and aims to find the best image alignment warping by minimizing these losses [18–21]. Nie et al. [31] synthesized stitched images at seams through unsupervised learning of seam-driven composite masks, achieving state-of-the-art performance with perfect transitions at seams. It seems that the stitched images still exhibit many irregular boundaries as depicted in Figure 2c.

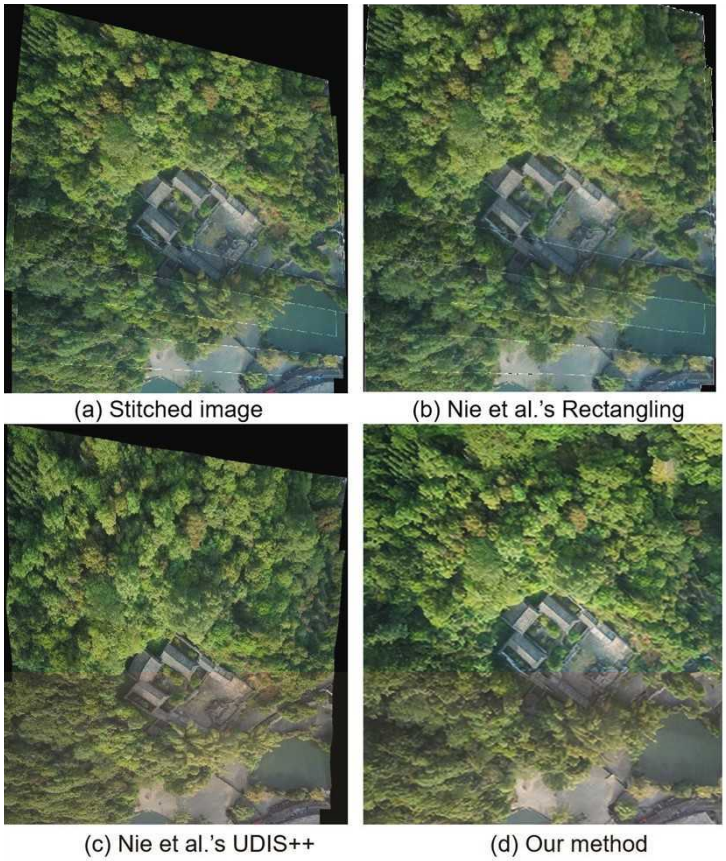


Figure 2. Comparison of different solutions: (**a**) Raw stitched image; (**b**) Nie et al.'s retangling [30]: most areas are rectangularized, but this method cannot completely eliminate irregular boundaries; (**c**) Nie et al.'s UDIS++ [31]: this method can achieve perfect transitions at seams but exhibits large irregular boundaries; (**d**) our method: the proposed method provides perfect transitions at seams and completely eliminates irregular boundaries.

2.2. Denoising Diffusion Probabilistic Models

Inspired by non-equilibrium thermodynamics, a generative model known as the diffusion model is proposed [22]. Building upon this, a denoising diffusion probability model has been demonstrated to be effective for high-quality image generation [23]. In this paper, we introduce diffusion models [23] as a generative method. Similar to other generative models, the denoising diffusion probability model (DDPM) learns the distribution of images given a training dataset. The inference process includes sampling a random noise vector x_T and progressively denoising it until a high-quality output image x_0 is achieved. During the training process, the DDPM method has a diffusion process, transforming the image x_0 into white Gaussian noise having a mean of zero and a standard deviation of one, i.e., $x_T \sim \mathcal{N}(0,1)$, over T time steps. Each step in the forward direction is given by,

$$q(x_1|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \;\; q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{1}$$

where the sample x_t is obtained by adding i.i.d. Gaussian noise with variance β_t at time step t and scaling the previous sample x_{t-1} using $\sqrt{1-\beta_t}$ based on the variance table.

The DDPM is trained to reverse the process presented in Equation (1). The inverse process is modeled by a neural network, which predicts the parameters $\mu_\theta(x_t, t)$ and variance $\sum_\theta(x_t, t)$ of a Gaussian distribution, given by

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \;\; p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sum_\theta(x_t, t)). \tag{2}$$

The learning objective of the model is derived through the variational lower bound on the log-likelihood between the reverse process and the diffusion process. This involves the use of variational inference to establish an objective function that the model aims to maximize during the learning process, which can be described as

$$\begin{aligned} \mathbb{E} = [-log\ p_\theta] &\leq \mathbb{E}_q[-log\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}] \\ &= \mathbb{E}_q[-log\ (x_T) - \sum_{t\geq 1} log\frac{p_{\theta(x_{t-1}|x_t)}}{q(x_t|x_{t-1})}] = L. \end{aligned} \tag{3}$$

Equation (3) has been reformulated by Ho et al. [23], which can be written as

$$\begin{aligned} &\mathbb{E}_q[\underbrace{D_{KL}(q(x_T|x_0)||p(x_T))}_{L_T} \\ &+ \sum_{t>1}\underbrace{D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))}_{L_{t-1}}\underbrace{-log\ p_\theta(x_0|x_1)}_{L_0}]. \end{aligned} \tag{4}$$

As pointed out by Ho et al. [23], the optimal approach for the parameterized model is to predict the cumulative noise added to the current image x_t, and for the predicted mean $\mu_\theta(x_t, t)$, it can be given by

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{a}_t}}\epsilon_\theta(x_t, t)\right). \tag{5}$$

With L_{t-1} in Equation (4), a simplified training objective function was derived by Ho et al. [23], which can be given by

$$L_{simple}(\theta) = E_{t,x_0,\epsilon}[\|\epsilon - \epsilon_\theta(x_t, t)\|^2, \tag{6}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is the white Gaussian noise added to the image through random sampling. ϵ_θ represents the parameters of the model undergoing training, which is responsible for predicting the noise level of the image sample x_t at time step t. Therefore, we can efficiently train the model using data and generate image data that conform to the distribution of interest through the process of reverse sampling.

3. Methods

For a set of UAV images $I_1 \ldots I_n$, the target of stitching is to obtain a broader field of view. As mentioned above, irregular boundaries and stitching seams can adversely affect the visual perception of the result. This section presents the proposed method that can stitch a set of images, addressing irregular boundaries and stitching seams simultaneously. The overall diagram of the proposed method is illustrated in Figure 3. The proposed method has the following three main parts:

(1) Compute masks for irregular boundaries and stitching seams that occur during image stitching, determining the areas that need to be repaired;
(2) Adjust the input and output image sizes in an adaptive way to match the input and output dimensions of the diffusion model;

(3) Employ the diffusion model to perform inverse diffusion on the stitched image with masks, repairing the masked regions.

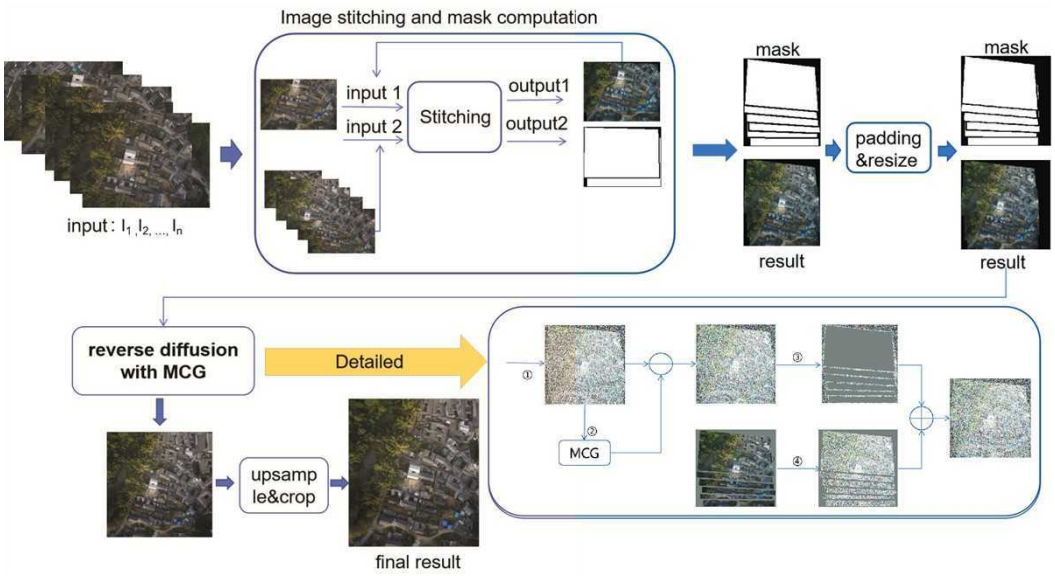


Figure 3. The overall diagram of the proposed method. The framework consists of two stages: stitching and restoration. In the stitching stage, the boundary and seam masks for each stitched image are also computed. In the restoration stage, the inverse process of the pre-trained diffusion model is used to restore the raw stitched image with the help of the masks computed in the stitching stage.

3.1. UAV Image Stitching and Mask Computation

To handle the irregular boundaries and stitching seams in the stitched images, this paper designs an efficient method for computing masks. Using this method allows us to obtain masks for stitching seams and irregular boundaries simultaneously for the UAV image stitching task.

The process of computing the mask corresponding to stitching seams involves transforming each target image through its corresponding cumulative matrix. By calculating the lines between the transformed rectangular vertices, the position of the stitching seam mask can be determined. The process of calculating the mask for irregular boundaries involves determining the minimum rectangle of the overlaid image on the source image based on the positions of the transformed rectangular vertices. The irregular boundary is then the complement of the overlaid image within this minimum rectangle. Through these steps, a unified mask computation scheme for irregular boundaries and seams is obtained to facilitate the repairment task. The following two parts introduce the two important steps of the proposed method in detail: UAV image alignment and calculation of masks for stitched images.

3.1.1. UAV Image Alignment

In order to obtain stable feature points, this paper utilizes the Scale-Invariant Feature Transform (SIFT) method [32] to extract features from pairs of images with overlapping regions, resulting in feature description vectors D and keypoint position vectors P. After that, the paper calculates the feature points for the matching image pairs I_j and I_{j-1}. To achieve sufficient accuracy with less time consumption, the Hierarchical Navigable Small World (HNSW) method [33] is employed to build and search the graph of feature vectors. Based on the relationship between feature vectors and keypoints, RANSAC is utilized to calculate a convergent homography matrix H_j for the transformation from image I_j to I_{j-1}. The overall process is shown in Figure 4.

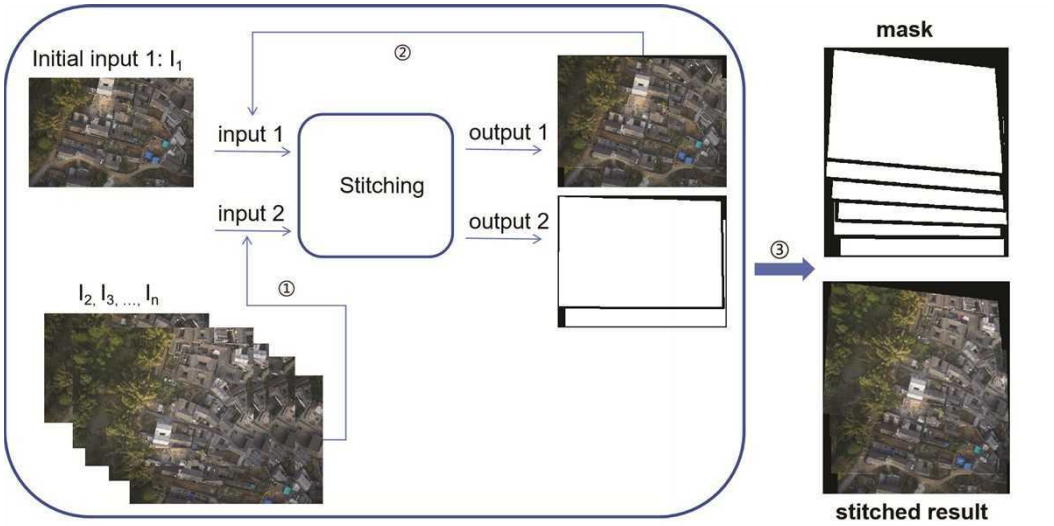


Figure 4. Stitching process. The first UAV image is used to initialize input 1, and the remaining UAV images I_2 to I_n are used in order as input 2. After stitching the two images, i.e., input 1 and input 2, there are two ouputs: One is output 1 indicating the stitched image, and the other output is the masks. Ouput 1 is repeatedly used as input 1 until all UAV images are used in the image stitching stage. Accordingly, the final output consists of the stitched image and the masks for irregular boundaries and seams.

3.1.2. Computation of Masks for Stitched Images

To transform I_j onto I_{j-1} and simultaneously calculate the corresponding positions for irregular boundaries and seams to generate masks, the cumulative matrix H_j^* needs to be computed from I_j to I_{j-1} , which is

$$H_j^* = H_j H_{j-1} \dots H_2. \tag{7}$$

Simultaneously, to obtain the smallest possible irregular boundary while maintaining the integrity of the range of the stitched image, we define a translation matrix H_{trans}, which can be given by

$$H_{trans} = \begin{bmatrix} 1 & 0 & -minx \\ 0 & 1 & -miny \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

where $minx$ and $miny$ are the minimum values between the vertices after transforming the target image and the source image vertex coordinates. The translation matrix is then superimposed on the cumulative matrix to obtain the final transformation from I_j and I_{j-1}.

The position of the seams is calculated by determining the location of the vertices after transforming the image through the cumulative matrix, and a mask is generated accordingly. To ensure consistency between the designed masks and the stitched images, each time the stitching occurs, the target image is placed on top of the source image. Accordingly, when each target image undergoes transformation, a mask of the same size as the target image is generated. This mask undergoes the same transformation as the target image to determine the range that needs to be placed on top of the image. The specific steps are summarized in Algorithm 1.

Algorithm 1 Stitching and Mask Calculation

Input: UAV images $I_1, I_2, \ldots I_n$, and its corresponding blank mask $mask_1, mask_2, \ldots mask_n$
Output: Stitched image and masks of seam and irregular boundary

Extracting feature points using SIFT for each image $P_j, D_j \leftarrow SIFT(I_j)$
Using HNSW and RANSAC for feature point matching and mismatch elimination to obtain the homography matrix H_j warping I_j to I_{j-1}
for $j = 2$ *to* N **do**
    Compute the cumulative matrix $H_j^* \leftarrow H_j H_{j-1} \ldots H_2$
    Compute the the positions of vertices using the cumulative matrix $V_j^* \leftarrow V_j H_j^*$
    Compute the minimum coordinate values $minx, miny \leftarrow x, y \in V_j^* \cup V_{j-1}^* \cup \cdots \cup V_1$;
    Compute the translation matrix H_{trans} based on $minx, miny$;
    Compute the warp of I_j and its mask $I_j^*, mask_j^* \leftarrow H_{trans} H_j^* (I_j, mask_j)$
    Compute the warp of I_{j-1} and its mask $I_{j-1}^*, mask_{j-1}^* \leftarrow H_{trans}(I_{j-1}, mask_{j-1})$
    Compute the stitched result $I_j \leftarrow (I_{j-1}^* \odot mask_j^*) + I_j^*$
    Compute the mask of seam $mask_{seam} \leftarrow Line(V_j^*)$
    Compute the overall mask $mask_j \leftarrow mask_{j-1}^* + mask_j^* + mask_{seam}$
end for

3.2. Irregular Boundaries and Stitching Seam Repairment with a Diffusion Model

For a forward diffusion process $x_t \in \mathbb{R}^n, t \in [0, n]$, we set $x_0 \sim p_0(x) = p_{data}$ where p_{data} represents the data distribution of the image of interest and $x_n \sim p_n(x)$ with $p_n(x)$ approximates a completely Gaussian distribution without data information. The process can be defined by the following stochastic differential equation (SDE) [34]:

$$dx = \bar{f}(x, t)dt + \bar{g}(t)dw, \tag{9}$$

where $\bar{f}$ is the linear drift function, $\bar{g}(t)$ is a scalar diffusion coefficient, and w denotes the standard Wiener process. The forward SDE is coupled with the following reverse SDE:

$$dx = [\bar{f}(x, t) - \bar{g}(t)^2 \nabla_x log\ p_t(x)]dt + \bar{g}(t)d\bar{w}, \tag{10}$$

where d_t denotes the infinitesimal negative time step and $\bar{w}$ defines the standard Wiener process running backward in time. The reverse SDE defines the generation process through the score function $\nabla_x log\ p_t(x)$. When the parameters of the score function's parametric model $s_{\theta(x_t,t)}$ are estimated, it can be substituted into the score function in Equation (10), enabling the solution of the reverse SDE. This can be expressed as follows:

$$x_{i-1} = f(x_i, s_\theta) + g(x_i)z,\ z \sim \mathcal{N}(0, I). \tag{11}$$

As reported by Ho et al. [23], the optimal approach is to predict the cumulative noise added to the current image, as described in Equation (5). However, unlike unconditional generation, our goal is to recover unknown pixels $x \in \mathbb{R}^n$ from known observational values $y \in \mathbb{R}^m$:

$$y = Hx + \epsilon, H \in \mathbb{R}^{m \times n}, \tag{12}$$

where H is the mask matrix having m rows and n columns and ϵ is the noise in the observed values.

The inverse problem that we aim to solve is to recover the image x at the masked positions of the stitched image with masks y. According to Bayesian theorem, we should replace the score function in Equation (10) with $\nabla_x log\ p_t(x|y)$. However, this would lead to the need for retraining the score function because the conditions change, making it unable to generalize for masks with different distributions in stitched images. Recent studies on conditional diffusion models [28,34,35] have utilized unconditional score functions, but they often only rely on a projection-based measurement constraint to impose the conditions.

The Manifold Gradient Constraint (MCG) [28] has demonstrated much better performance in image recovery, which can be expressed as

$$x'_{i-1} = f(x_i, s_\theta) - \alpha \frac{\partial}{\partial x_i} \|W(y - Hx_i)\|_2^2 + g(x_i)z, \;\; z \sim \mathcal{N}(0, I), \tag{13}$$

where α and W depend on the noise covariance. The specific form of the Manifold Gradient Constraint is expressed as follows:

$$\frac{\partial}{\partial x_i} \|W(y - Hx_i)\|_2^2 = -2J_{Q_i}^T H^T W^T W(y - H\hat{x}_0), \tag{14}$$

where Q_i denotes the local projection mapping of x_i onto the manifold and J_f denotes the Jacobian matrix of the vector f. We pre-train the diffusion model using the UAV image dataset. For any stitched images, applying the Manifold Gradient Constraint (MCG) during its inverse diffusion allows obtaining high-quality restoration images. The process is illustrated in Figure 5.

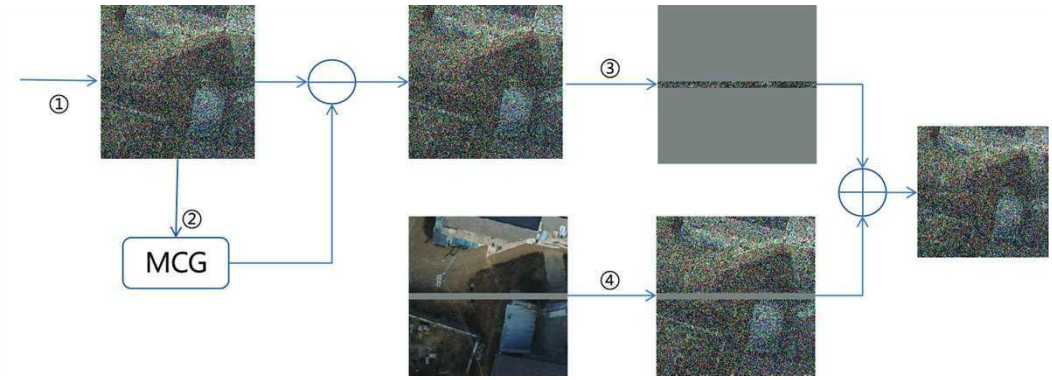


Figure 5. The inverse process of the diffusion model for restoration: ① The initial sampling of inverse diffusion; ② performing inverse iteration using the manifold gradient constraint; ③ taking the orthogonal complement with the mask; ④ sampling from $p(y_i|y)$ and adding to the orthogonal complement.

3.3. Image Dimension Adaptation

Due to varying spatial correspondences between images, different image stitching may result in generating images with different sizes. The diffusion model is trained using images having a specific size, making a single diffusion model ineffective for handling images having multiple different sizes. Training models separately for all possible sizes of stitched results is extremely difficult if not impossible. This requires the adaptation of images to match the demanded size of the model before using the diffusion model.

Directly resizing images using interpolation methods is one solution, while such methods may result in some information loss. In contrast, deep learning-based methods can achieve better reconstruction quality. However, most of such methods are trained only for fixed magnification mapping, making it unable to handle cases where images need arbitrary resolution scaling. To address this issue, Chen et al. [29] proposed the Local Implicit Image Function (LIIF). This method trains an EDSR encoder and a shared function f_θ using a neural network. The EDSR encoder maps the image to a two-dimensional feature $M_i \in R^{H \times W \times C}$, as shown in Figure 6.

The shared function can be queried to obtain the pixel value for any continuous coordinate in the image, which enables arbitrary resolution adjustment of the image, producing high-quality results while minimizing potential information loss associated with interpolation methods. The shared function can be given by

$$I(x_q) = f_\theta(z^*, x_q - v^*), \tag{15}$$

where z^* and v^*, respectively, represent the neighboring encoding and coordinates in the feature M_i. Each latent code z in M_i represents a local segment of the continuous image. It is responsible for predicting the signal for a set of coordinates closest to itself.

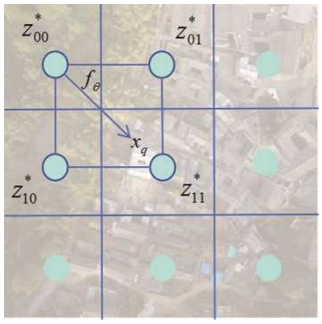


Figure 6. LIIF representation with local ensemble.

The pixel-based representation of LIIF is plotted in Figure 7. In this figure, x_{hr} represents the central coordinates of the pixel to be queried, and s_{pred} is the obtained RGB value from the query. Due to the different aspect ratios between stitched images and the diffusion model, directly resizing stitched images may cause stretching and deformation. To avoid this problem, before adjusting the image size, padding is applied to adjust the ratio of the length and width of the stitched images to match the trained diffusion model, which is

$$\frac{H_{stitched}}{W_{stitched}} = \frac{H_{diffusion}}{H_{diffusion}}, \ \lambda = \frac{H_{diffusion}}{H_{stitched}}. \tag{16}$$

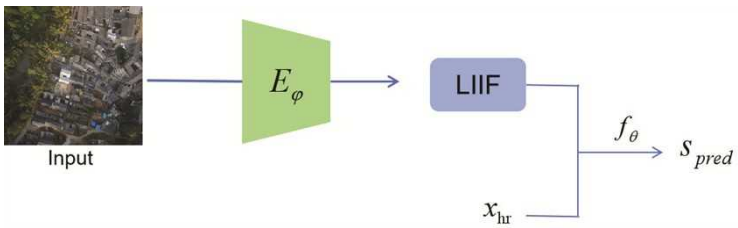


Figure 7. Pixel-based representation of LIIF.

The padded image is then used to calculate the central coordinates of each pixel with a scaling factor λ. LIIF is employed for querying to achieve adjustment to match the size of the diffusion model. For the results after restoration through the diffusion model, the size is similarly restored using LIIF, and the padded areas are cropped to obtain the restored image.

4. Experiment

4.1. Data Preparation

We conducted on-site aerial photography using a UAV, capturing a total of 1008 images at a resolution of 6000 × 4000 pixels. From these, 48 images were set aside as a validation set denoted as dataset 3. Each set of 6 images within this validation set was designated as a stitching object, resulting in a total of 8 groups. Due to the significant difference in data dimensions compared to the diffusion model, we downsampled the images to a resolution of 480 × 320 pixels. The remaining 960 images were cropped into squares, downsampled to 256 × 256 pixels, and subjected to random flips or exchanges, resulting in 1920 images to form the training dataset denoted as dataset 1 for the diffusion model. The original high-resolution dataset denoted as dataset 2 was utilized as the training dataset for the Learning-based Inpainting with Fourier Features (LIIF) model.

4.2. Model Training Details

4.2.1. Diffusion Model Training

The network model used in this paper is a U-Net model with temporal injection, as shown in Figure 8. The backbone network is a residual network with attention layers. To enable the model to learn sufficient details for image restoration, we set the channels of the residual block to 256. Within the same-scale module of the U-Net, the number of residual blocks was set to 2, and the number of heads in the attention module was set to 64. This paper compares the restoration results of two models trained on ImageNet 256 × 256 and our dataset 1 of UAV images, respectively. The goal is to investigate the generalization and fitting of the models in the restoration task. We conducted training on an NVIDIA RTX 3090 GPU (Santa Clara, CA, USA) with a batch size of 1. As shown in Figure 9, the model trained on the ImageNet dataset for 500k epochs occasionally produced counterintuitive images when the mask proportion was high. In contrast, the model trained solely on our dataset 1 for 180k epochs demonstrated excellent inference capabilities, consistently generating high-quality images.

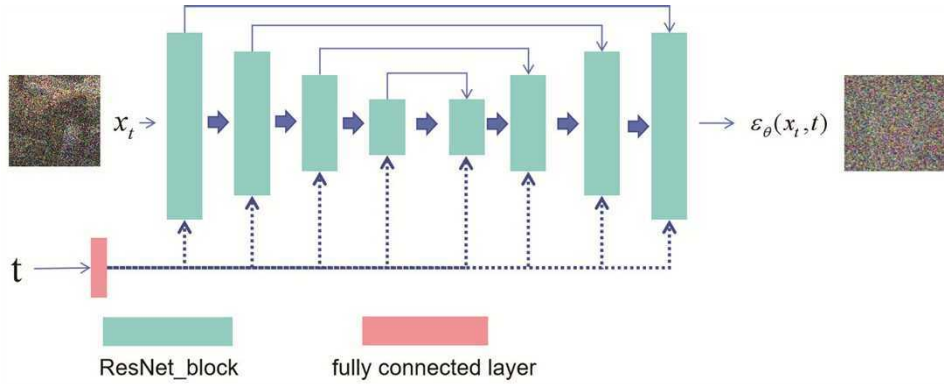


Figure 8. U-Net model with time injection. Each layer of the U-Net consists of a ResNet with different sizes.

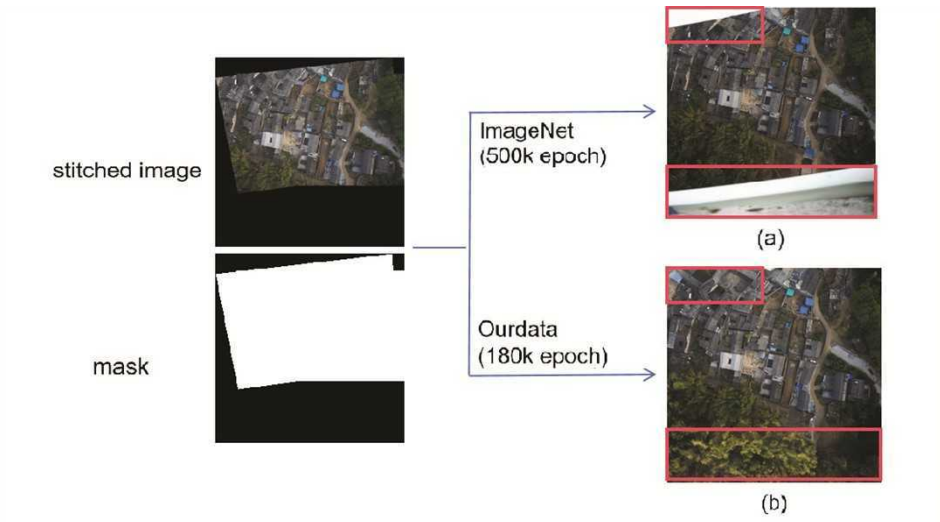


Figure 9. Comparison of the model trained on ImageNet and that trained on dataset 1 in terms of perceptual quality: (**a**) model trained on ImageNet with two patching errors (red boxes); (**b**) model trained on dataset 1.

4.2.2. LIIF Training

This part compares the performance of a pre-trained model [29] with a model fine-tuned with our dataset. Although using this pre-trained model for image size adjustment directly has shown significant improvement when compared with direct interpolation

methods, some features such as the direction of roof tiles are still over smoothed. We fine-tuned the pre-trained model with our dataset (dataset 2) and compared the final results with the pre-trained model. One can observe that, at the adjustment of $\times 23$ in image size for downsampling and upsampling, using the pre-trained model directly leads to the loss of many texture details. After fine-tuning the pre-trained model with our dataset, the model exhibits richer texture details, as shown in Figure 10. If only at a smaller scale adjustment, such as $2.3\times$, the difference in image details is difficult to observe when compared with the ground truth, as shown in Figure 11.



Figure 10. Comparison of the details of LIIF after data augmentation under large-scale adjustment. (**a**) Ground truth data; (**b**) details of the image at 23.4× downsampling and upsampling using the pre-trained model; (**c**) details of the image at 23.4× downsampling and upsampling using the fine-tuned model.

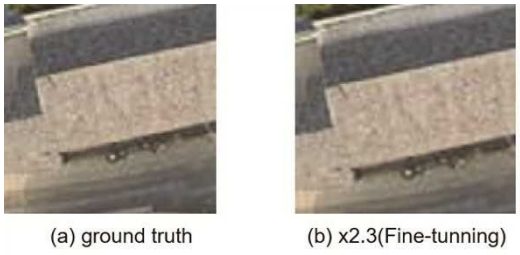


Figure 11. Comparison of the details under small-scale adjustments after model enhancement. (**a**) Ground truth data, (**b**) details of downsampling and upsampling by ×2.3 with the enhanced model.

4.3. UAV Image Stitching Results and Analysis

4.3.1. Overall Results Comparison

We evaluated the performance of many stitching and restoration methods using the UAV images taken from dataset 3. The proposed method was compared with AANAP [36], UDIS++ [31], and Deep Rectangling [30]. The comparison results are shown in Figure 12. From this figure, it is obvious that the proposed method performs better than the competing methods in terms of both image completeness and transition smoothness.

4.3.2. Seam Repairment Details

We compared the repairment details at the image seams in our results, as shown in Figure 13. It can be observed that, although high-quality repairment results were achieved, some small textures at the seams were altered.

Furthermore, we compared the results of repairing seams with different widths of masks for the same degree of misalignment as shown in Figure 14. We can observe that when using the red mask, which cannot completely cover the misaligned area, the repairment result is relatively poor. On the other hand, when using the yellow mask, which can entirely cover the entire misaligned area, the repairment effect is promising but comes with the trade-off of accepting some changes in fine textures within the masked region when compared with the ground truth. Through experiments conducted on the training dataset as the preliminary experiments, we found that setting the mask width to be between 1/60 and 1/80 of the total pixels can achieve a balanced performance. For

practical applications, adjustments can be made based on the actual displacement at the seams, which is out of the scope of this paper.

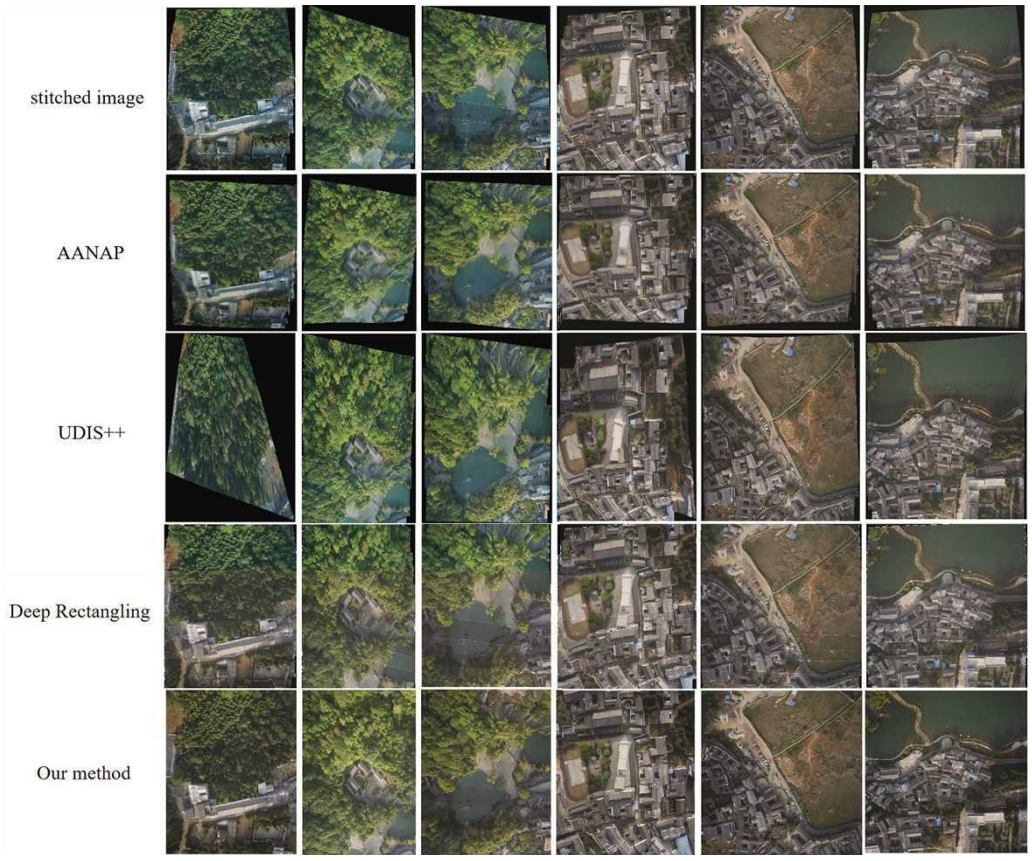


Figure 12. Comparison of the final results from different methods.



Figure 13. Comparison of seam repairment details with and without the proposed seam repairment method. (**a**) The first example, (**b**) the second example. The images with and without the proposed method for each example are placed on the left and right parts, repsectively.

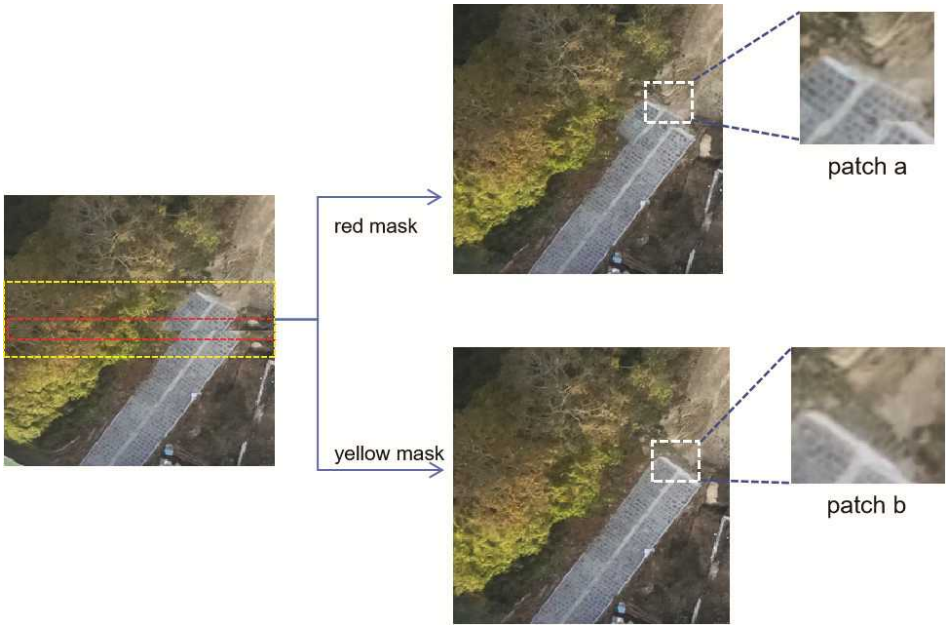


Figure 14. Comparison of the repair effects of different width masks on the same seam. (patch a) Solving the seam with the red mask, the transition is natural, but misalignment still exists. (patch b) Solving the seam with the yellow mask, misalignment and seam repair are good, but it may introduce unrealistic changes.

For the repairment of irregular boundaries, we captured additional images from a perspective roughly similar to the stitched result. We identified corresponding content for some irregular boundaries in these images and used them as ground truth for comparison with our repair results. The repair outcomes exhibit a consistent alignment with the ground truth in terms of both texture extension and semantics, as depicted in Figure 15.

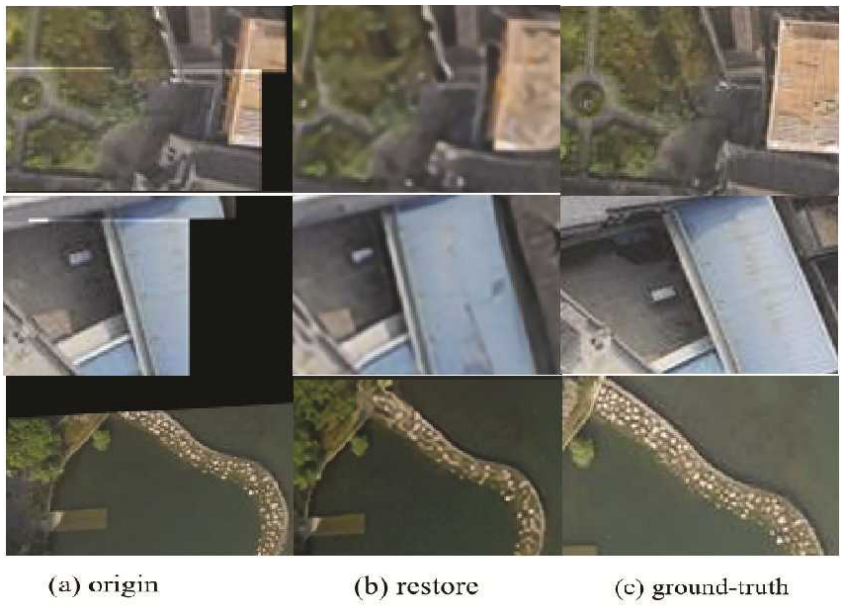


Figure 15. Comparison between repaired values and ground truth for irregular boundaries.

4.4. Quantitative Evaluation

Due to the difficulty in obtaining globally accurate reference values for the repaired images, we manually selected image patches as the ground truth for comparison with the results before and after restoration. This allows us to assess the overall quality of

the images. In this study, we used the SSIM and PSNR metrics to compare the similarity between the images before and after restoration and the reference values. We manually selected image regions from the stitched images as reference values and calculated the average similarity scores before and after restoration using the above-mentioned metrics. The results are presented in Table 1. The similarity scores after restoration showed a significant improvement, attributed to the high level of agreement between the repaired generated images and the reference values in terms of texture and semantics.

Table 1. Measurement of similarity between images with and without the proposed restoration method and the reference image.

Method	SSIM↑	PSNR↑
Origin	0.361	12.724
Restore	**0.434**	**17.011**

Due to the limited comparison conducted in certain regions of the images and the absence of the global ground-truth reference, we further compared the quality of the images in an intuitive way. Our results repaired the texture at seams, making transitions smooth, and eliminated large irregular boundaries. However, due to the upsampling and downsampling during the restoration process, resolution could be decreased, which adversely affects the evaluation of our restored results. We further employed a non-intrusive perceptual image quality assessment tool, e.g., PaQ-2-PiQ [37], to solve the problem of the absence of the global ground-truth reference. It should be mentioned that PaQ-2-PiQ utilizes a large subjective quality database for training and can infer globally to locally and locally to globally, and it focuses more on the overall subjective perception of images rather than just sharpness. It also provides more consistent evaluation results with human perception, especially for "more blurred but more visually pleasing" images. When using PaQ-2-PiQ to evaluate different methods, the proposed method was compared with UDIS++, AANAP, and Deep Rectangling. The PaQ-2-PiQ scores are summarized in Table 2. The quantitative results double confirm that the proposed method outperforms many existing SOTA methods in terms of perceptual quality, which is consistent with the visual results presented in Figure 12. For methods that require multiple stitching operations, such as AANAP and UDIS++, numerous cumulative errors might be inevitably introduced during repeating the stitching process several times, resulting in their scores being even lower than the original stitched results.

Table 2. Comparison of PaQ-2-PiQ before and after restoration.

Method	PaQ-2-PiQ↑
Stitched image	0.741
AANAP	0.684
UDIS++	0.734
Deep Rectangling	0.743
Ours	**0.766**

4.5. Ablation Studies

To demonstrate the effectiveness of using the combined mask including the boundary and seam masks, we compared the three restoration results: using only the irregular boundary mask, only the seam mask, and the two masks combined.

Boundary Mask. If masks are only computed for dealing with irregular boundaries, the proposed method can only be expected to reduce the irregular boundaries, while the seams cannot be solved. In this case, there might be unnatural transitions at the junctions between irregular boundaries and image edges, as indicated by the arrows in Figure 16a–c.

Seam mask. If masks are computed only for seams, the proposed method can only be expected to reduce the seams, while it is unable to achieve promising results with a

rectangular field of view, and nonlinear distortion still occurs at the edges of irregular boundaries as shown in Figure 16d–f.

When using the combination of the boundary and seam masks, it is expected that the proposed method can reduce both the irregualr boundaries and seams. This is the case that can be seen from Figure 16g–i. From Figure 16, we can conclude that if we separately compute masks for irregular boundaries and seams, unnatural transitions and nonlinear distortions often occur at some edges of the image. Moreover, if we repair the remaining parts continuously, the masks calculated earlier cannot accurately correspond to the repaired results.

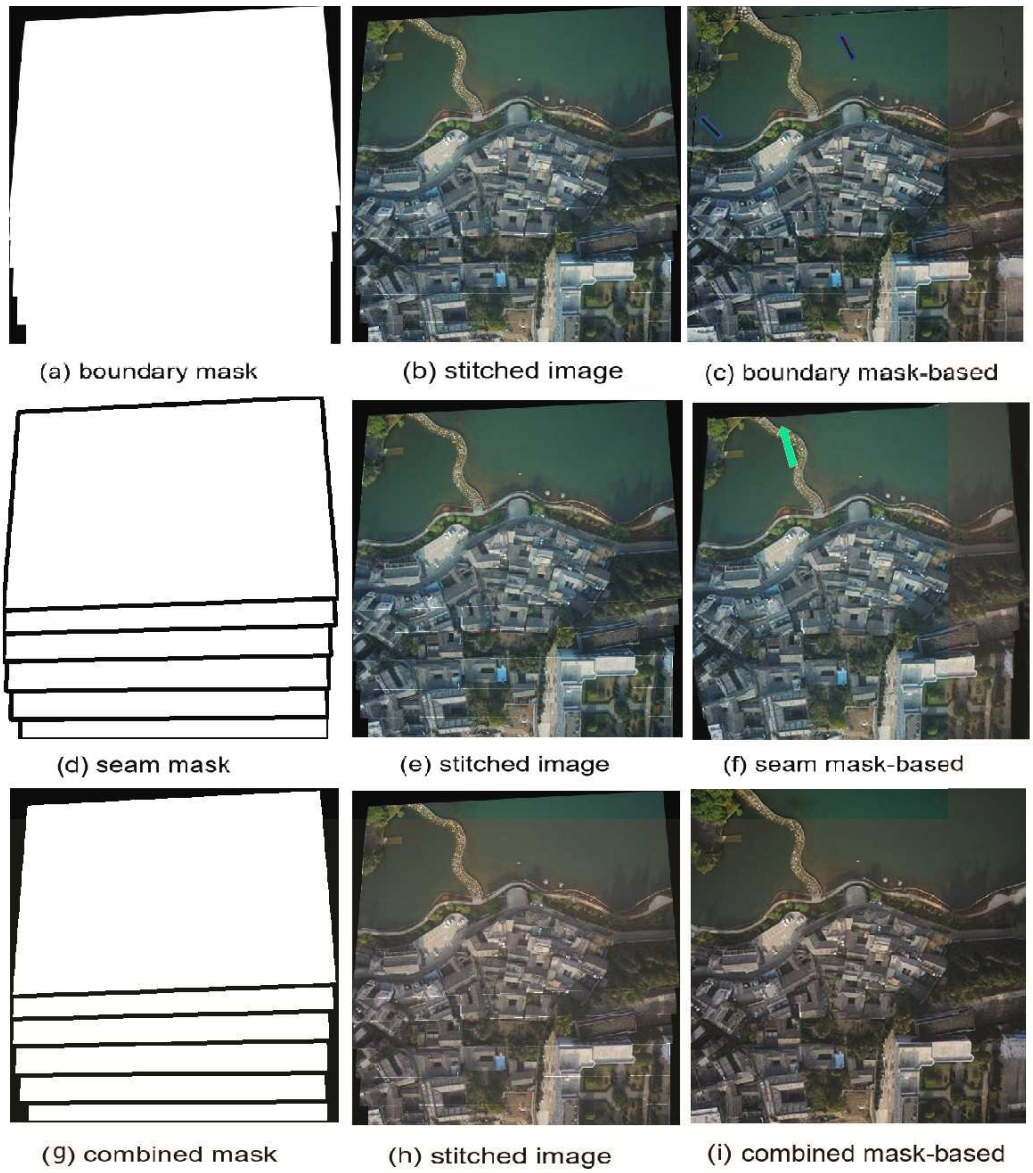


Figure 16. The ablation study using different masks. (**a**–**c**) show the importance of using boundary masks; (**d**–**f**) show the importance of using seam masks; (**g**–**i**) show the importance of using both boundary and seam masks.

5. Conclusions

We addressed the irregular boundaries and stitching seams for the UAV image stitching task by devising a method to generate masks during image stitching, and a processing framework based on the diffusion model was developed, in which the unconditional score function of the diffusion model is ultilized during the inverse process. We also applied

additional manifold gradient constraints to repair masked images, aiming to eliminate irregular boundaries and stitching seams, resulting in improved perceptual quality.

Unlike those often-used methods based on grids or energy minimization functions, our proposed method is data-driven, considering the overall distribution of pixels. It does not strictly require specific geometric structures or texture features near irregular boundaries and stitching seams. Therefore, even in challenging scenarios with complex shapes of irregular boundaries, multiple seams, and insufficient geometric and texture features, our proposed method can still achieve high-quality results. Moreover, due to the powerful learning and inference capabilities of the diffusion model for data distribution, training on a small dataset of UAV images can obtain a high-quality model for the restoration of stitched images, facilitating its potential applications.

Note that our proposed method may introduce some perceptually acceptable but "unrealistic" or artificial noise locally into the mask. In scenarios where strict authenticity requirements for data are essential, such as in ultra-high-resolution map drawing, introducing small unrealistic/artificial noise may be intolerable. Yet, for scenarios with a higher tolerance for realism, such as VR/AR applications, our proposed method may provide an alternative way to improve the quality of the stitched images. Evaluating and quantifying the introduced unrealistic noise as well as determining its acceptability still lack quantitative analysis tools. Future research can focus on how to quantitatively evaluate these introduced unrealistic noise components and solve this problem in an efficient way.

Author Contributions: Conceptualization, J.C., J.W., J.L. and Y.L.; methodology, J.C. and Y.L.; software, Y.L., H.T. and J.C.; validation, Y.L.; formal analysis, J.C. and Y.L.; investigation, Y.L. and J.C.; resources, J.C.; data curation, J.C.; writing—original draft preparation, Y.L. and C.J.; writing—review and editing, J.C. and Y.L.; funding acquisition, Y.T. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The National Natural Science Foundation of China (Grant No. 42327801).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions. Our trained model can be accessed from the following page: https://huggingface.co/xhehe/uav_diffusion/tree/main, accessed on 24 March 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LIIF	Local Implicit Image Function
DDPM	Denoising Diffusion Probabilistic Models
HNSW	Hierarchical Navigable Small World
SIFT	Scale-Invariant Feature Transform
UAV	Unmanned Aerial Vehicle
GAN	Generative Adversarial Network

References

1. Gómez-Reyes, J.K.; Benítez-Rangel, J.P.; Morales-Hernández, L.A.; Resendiz-Ochoa, E.; Camarillo-Gomez, K.A. Image mosaicing applied on UAVs survey. *Appl. Sci.* **2022**, *12*, 2729. [CrossRef]
2. Chen, Y.S.; Chuang, Y.Y. Natural image stitching with the global similarity prior. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 186–201.
3. Lee, K.Y.; Sim, J.Y. Warping residual based image stitching for large parallax. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8198–8206.
4. Lin, W.Y.; Liu, S.; Matsushita, Y.; Ng, T.T.; Cheong, L.F. Smoothly varying affine stitching. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 345–352.
5. Zaragoza, J.; Chin, T.J.; Brown, M.S.; Suter, D. As-projective-as-possible image stitching with moving DLT. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2339–2346.

6. Guo, D.; Chen, J.; Luo, L.; Gong, W.; Wei, L. UAV image stitching using shape-preserving warp combined with global alignment. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 8016005. [CrossRef]
7. Cui, Z.; Tang, R.; Wei, J. UAV image stitching with Transformer and small grid reformation. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 5001305. [CrossRef]
8. He, L.; Li, X.; He, X.; Li, J.; Song, S.; Plaza, A. VSP-Based Warping for Stitching Many UAV Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5624717. [CrossRef]
9. He, K.; Chang, H.; Sun, J. Rectangling panoramic images via warping. *ACM Trans. Graph.* **2013**, *32*, 1–10. [CrossRef]
10. Li, D.; He, K.; Sun, J.; Zhou, K. A geodesic-preserving method for image warping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 213–221.
11. Zhang, Y.; Lai, Y.K.; Zhang, F.L. Content-preserving image stitching with piecewise rectangular boundary constraints. *IEEE Trans. Vis. Comput. Graph.* **2020**, *27*, 3198–3212. [CrossRef] [PubMed]
12. Kwatra, V.; Schödl, A.; Essa, I.; Turk, G.; Bobick, A. Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph.* **2003**, *22*, 277–286. [CrossRef]
13. Li, N.; Liao, T.; Wang, C. Perception-based seam cutting for image stitching. *Signal Image Video Process.* **2018**, *12*, 967–974. [CrossRef]
14. Chen, X.; Yu, M.; Song, Y. Optimized seam-driven image stitching method based on scene depth information. *Electronics* **2022**, *11*, 1876. [CrossRef]
15. Agarwala, A.; Dontcheva, M.; Agrawala, M.; Drucker, S.; Colburn, A.; Curless, B.; Salesin, D.; Cohen, M. Interactive digital photomontage. In *ACM SIGGRAPH 2004 Papers*; Association for Computing Machinery: New York, NY, USA, 2004; pp. 294–302.
16. Dai, Q.; Fang, F.; Li, J.; Zhang, G.; Zhou, A. Edge-guided composition network for image stitching. *Pattern Recognit.* **2021**, *118*, 108019. [CrossRef]
17. Eden, A.; Uyttendaele, M.; Szeliski, R. Seamless image stitching of scenes with large motions and exposure differences. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 7–22 June 2006; Volume 2, pp. 2498–2505.
18. Lin, K.; Jiang, N.; Cheong, L.F.; Do, M.; Lu, J. Seagull: Seam-guided local alignment for parallax-tolerant image stitching. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 370–385.
19. Gao, J.; Li, Y.; Chin, T.J.; Brown, M.S. Seam-driven image stitching. In Proceedings of the Eurographics, Girona, Spain, 6–10 May 2013; pp. 45–48.
20. Zhang, F.; Liu, F. Parallax-tolerant image stitching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3262–3269.
21. Li, J.; Zhou, Y. Automatic color image stitching using quaternion rank-1 alignment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 8–24 June 2022; pp. 19720–19729.
22. Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2256–2265.
23. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
24. Nichol, A.Q.; Dhariwal, P. Improved denoising diffusion probabilistic models. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 8162–8171.
25. Dhariwal, P.; Nichol, A. Diffusion models beat gans on image synthesis. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 8780–8794.
26. Zhu, Y.; Zhang, K.; Liang, J.; Cao, J.; Wen, B.; Timofte, R.; Van Gool, L. Denoising Diffusion Models for Plug-and-Play Image Restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 1219–1229.
27. Kawar, B.; Elad, M.; Ermon, S.; Song, J. Denoising diffusion restoration models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 23593–23606.
28. Chung, H.; Sim, B.; Ryu, D.; Ye, J.C. Improving diffusion models for inverse problems using manifold constraints. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 25683–25696.
29. Chen, Y.; Liu, S.; Wang, X. Learning continuous image representation with local implicit image function. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8628–8638.
30. Nie, L.; Lin, C.; Liao, K.; Liu, S.; Zhao, Y. Deep rectangling for image stitching: A learning baseline. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 8–24 June 2022; pp. 5740–5748.
31. Nie, L.; Lin, C.; Liao, K.; Liu, S.; Zhao, Y. Parallax-tolerant unsupervised deep image stitching. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 7399–7408.
32. Ng, P.C.; Henikoff, S. SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Res.* **2003**, *31*, 3812–3814. [CrossRef] [PubMed]
33. Malkov, Y.A.; Yashunin, D.A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 824–836. [CrossRef] [PubMed]
34. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv* **2020**, arXiv:2011.13456.

35. Kadkhodaie, Z.; Simoncelli, E. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 13242–13254.
36. Lin, C.C.; Pankanti, S.U.; Natesan Ramamurthy, K.; Aravkin, A.Y. Adaptive as-natural-as-possible image stitching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1155–1163.
37. Ying, Z.; Niu, H.; Gupta, P.; Mahajan, D.; Ghadiyaram, D.; Bovik, A. From patches to pictures (PaQ-2-PiQ): Mapping the perceptual space of picture quality. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3575–3585.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

mathematics

MDPI

Article

Automatic Design of Energy-Efficient Dispatching Rules for Multi-Objective Dynamic Flexible Job Shop Scheduling Based on Dual Feature Weight Sets

Binzi Xu [1,*], Kai Xu [1], Baolin Fei [1], Dengchao Huang [1], Liang Tao [1] and Yan Wang [2]

[1] School of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China; 2230342129@stu.ahpu.edu.cn (K.X.); 2210320117@stu.ahpu.edu.cn (B.F.); huangdengchao@ahpu.edu.cn (D.H.); taoliang@ahpu.edu.cn (L.T.)

[2] School of IoT and Engineering, Jiangnan University, Wuxi 214122, China; wangyan88@jiangnan.edu.cn

* Correspondence: xubinzi@ahpu.edu.cn

Abstract: Considering the requirements of the actual production scheduling process, the utilization of the genetic programming hyper-heuristic (GPHH) approach to automatically design dispatching rules (DRs) has recently emerged as a popular optimization approach. However, the decision objects and decision environments for routing and sequencing decisions are different in the dynamic flexible job shop scheduling problem (DFJSSP), leading to different required feature information. Traditional algorithms that allow these two types of scheduling decisions to share one common feature set are not conducive to the further optimization of the evolved DRs, but instead introduce redundant and unnecessary search attempts for algorithm optimization. To address this, some related studies have focused on customizing the feature sets for both routing and sequencing decisions through feature selection when solving single-objective problems. While being effective in reducing the search space, the selected feature sets also diminish the diversity of the obtained DRs, ultimately impacting the optimization performance. Consequently, this paper proposes an improved GPHH with dual feature weight sets for the multi-objective energy-efficient DFJSSP, which includes two novel feature weight measures and one novel hybrid population adjustment strategy. Instead of selecting suitable features, the proposed algorithm assigns appropriate weights to the features based on their multi-objective contribution, which could provide directional guidance to the GPHH while ensuring the search space. Experimental results demonstrate that, compared to existing studies, the proposed algorithm can significantly enhance the optimization performance and interpretability of energy-efficient DRs.

Keywords: dynamic flexible job shop; genetic programming; dispatching rule; dual feature weight sets; energy-efficient

MSC: 90-08

1. Introduction

With the vigorous development of manufacturing technology and artificial intelligence technology, intelligent manufacturing has become one of the mainstream directions of the manufacturing industry [1,2]. So far, artificial intelligence technology has been effectively applied in many directions and fields of the manufacturing industry. In terms of the operation research and management aspect, it mainly focuses on using intelligent optimization algorithms to solve the job shop scheduling problem (JSSP) [3].

JSSP entails the efficient allocation of production resources for a decomposable processing task under certain constraints to optimize performance metrics, such as total processing time, flow time, and tardiness [4]. The flexible job shop scheduling problem (FJSSP) allows for more flexible use of machine resources compared to the JSSP, where each operation of a job can be processed on multiple candidate machines, better reflecting real-world

situations [5–7]. Therefore, it is essential to first assign jobs to suitable machines (i.e., the routing decisions) in the FJSSP and then sequence the jobs in the waiting queue of each machine for processing (i.e., the sequencing decisions).

The JSSP and the FJSSP, as typical NP-hard problems, have garnered widespread attention and thorough research from experts and scholars due to their extensive application and complexity in the manufacturing domain. For small-scale problems, researchers tend to utilize exact methods, such as mathematical programming [8] and branch-and-bound [9]. These methods can solve the problem effectively while ensuring the accuracy and optimality of the obtained solutions, but they also suffer from the drawbacks of being computationally intensive and time-consuming. On the other hand, to tackle large-scale problems and tighter time constraints, meta-heuristic algorithms have been viewed as prominent choices. Meta-heuristic algorithms, such as the genetic algorithm [10], the Tabu Search algorithm [11], particle swarm optimization [12], teaching–learning-based optimization [13], and simulated annealing [14], etc., can find approximate optimal solutions in a relatively short time, offering acceptable solutions for decision-making in practical production scenarios.

However, the actual manufacturing environment is usually dynamic, which means that unexpected random events (such as the arrival of new jobs, machine failures, reworking of jobs, etc.) often occur in the dynamic flexible job shop scheduling problem (DFJSSP) [15]. Traditional meta-heuristic methods typically use rescheduling mechanisms to tackle these dynamic events, but these approaches present challenges in terms of increased computational complexity and stability issues [16,17].

As another promising approach, dispatching rules (DRs) demonstrate their effectiveness as a heuristic strategy for solving the DFJSSP due to their scalability, reusability, and rapid response to dynamic events [18–20]. In essence, DRs are priority functions that consist of features containing job shop-related information and mathematical operators. At each routing and sequencing decision point, DRs compute priority values for each scheduling object (such as waiting operations and candidate machines) and choose the most prioritized one for the next processing action. In the past few decades, numerous types of features (e.g., processing time of each operation, idle time of each machine, etc.) and manually designed DRs (e.g., first in, first out, etc.) have been proposed to address various job shop scenarios and optimization objectives [21].

However, it is exceedingly challenging and perhaps even infeasible to depend solely on workers' expertise to clarify potential correlations among different features and to design effective DRs. This is due to the intricate interactive interconnection of various production components in the job shop, as well as the distinct decision environment and information requirements associated with different scheduling decisions in the DFJSSP. In this case, the application of a genetic programming hyper-heuristic (GPHH) for the automated design and generation of appropriate DRs has become a widely used method for effectively addressing the DFJSSP. Numerous studies have demonstrated that using GPHH could automatically evolve DRs that outperform manually designed ones [22,23].

GPHH has the ability to autonomously generate efficient DRs without extensive domain-specific knowledge. This is achieved by continuously adapting the combination of features and mathematical operators in the DRs during evolutionary iterations with predetermined algorithmic parameters (e.g., maximum depth of the tree, function set, feature set, etc.) and genetic operators (e.g., crossover, mutation, etc.). In comparison to most meta-heuristic algorithms for solving the JSSP, GPHH has the advantages of flexible encoding representation, powerful search capability, and easier application to real-world environments. Hence, DRs produced by GPHH are naturally suitable for solving the large-scale DFJSSP. In solving the JSSP, GPHH first uses a set of features (corresponding to leaf nodes) representing the states of jobs, machines, and job shops, as well as a set of mathematical functions (corresponding to non-leaf nodes) to generate DRs automatically and iteratively through off-line training. When scheduling online, the generated DRs are

directly used to make scheduling decisions without repeated training by GPHH, thus realizing real-time scheduling [24].

Previous works have shown that the maximum depth of DRs, the function set, and the feature set were the three main factors that could determine the diversity of DRs and the search space of GPHH [25]. If the maximum depth of the DRs is d, and the DRs are generated using the "full method" (i.e., full growth method), the size of the GPHH search space is $|\mathcal{F}|^{2^d-1} \times |\mathcal{T}|^{2^d}$, where $\mathcal{F}$ and $\mathcal{T}$ denote the function set and feature set, respectively. However, the number of effective DRs is much smaller than the above theoretical value, which indicates that finding solutions through GPHH is still a difficult task.

Therefore, reducing the number of features is a viable approach to narrow the search space of GPHH and enhance its searching efficiency [26,27]. The feature set of GPHH can encompass numerous features describing various aspects and forms of information pertaining to the job shop scheduling process, including system-related, machine-related, and job-related features. However, not all of these features contribute positively to scheduling decisions. For instance, the due date of a job is usually considered to be an irrelevant feature for optimizing the mean flow time [26]. In addition, there are correlations between many features, which can cause some of the same information to be double-counted. Therefore, by integrating feature selection techniques with GPHH, it is possible to eliminate irrelevant and redundant features, allowing the algorithm to purposefully explore regions containing more promising DRs and then improve algorithm efficiency.

As far as we know, the research on feature selection methods for GPHH and its variations is still limited and primarily focuses on single-objective problems. The initial studies emphasized measuring the importance of features by calculating the frequency of their occurrence in the best DRs [28]. Mei et al. [29] were the first to propose a metric to measure the feature importance based on the contribution of features to the fitness values of individuals. Compared to feature frequency, this metric could avoid the negative influence brought by redundant features and offer better accuracy. However, these methods were mainly explored in the context of the DFJSSP, considering only sequencing rules. Zhang et al. [30] were pioneers in applying feature selection methods to the DFJSSP, involving two feature sets for both routing and sequencing decisions, respectively. Additionally, in their subsequent work [31], they further developed a continuous two-stage GPHH framework to fully utilize excellent individuals during the feature selection process. However, although these methods could reduce the dimensionality of the feature set and enhance the interpretability of generated DRs through feature selection, they did not lead to an improvement in the optimization performance of GPHH.

Although feature selection has been successfully applied to single-objective problems, no feasible methods have been found for the multi-objective DFJSSP. The main challenge is that calculating the feature importance in multi-objective problems takes into account not only the performance indexes themselves, but also the correlation and non-domination relationships among these indexes. As a result, the existing feature importance measures based on fitness values for single-objective problems are no longer applicable, especially considering that the objectives in multi-objective problems are usually interrelated, and a feature may bring positive effects in one objective but may lead to negative effects in another. Therefore, it becomes very difficult to design an effective feature importance measure for multi-objective problems. In addition, the result of feature selection is heavily dependent on manually set parameters. For instance, the condition for features to be selected in the previous studies [29–31] is that the feature weight value is greater than or equal to half of the total weight value. Such a parameter setting is too rigid, lacking not only fault tolerance and flexibility, but also sufficient adaptability for different scheduling problem scenarios.

Aiming at the limitations of the aforementioned feature selection methods, this paper proposes an improved GPHH with dual feature weight sets to automatically design energy-efficient DRs. In this proposed GPHH, two novel feature weight measures are proposed for calculating the feature weights in the multi-objective DFJSSP and forming the dual feature

weight sets. Additionally, a novel hybrid population adjustment strategy is also presented to use the obtained dual feature weight sets to guide the algorithm. The main difference between the proposed feature weight measures and the existing feature selection methods is that the proposed feature weight measures do not select the features but rather assign proper weights to the features according to their contribution/importance to multi-objective problems, which serve as the probabilities of the features being selected in subsequent iterations. This way, the influence of manually set parameters on the feature set and the optimization performance of obtained DRs can be avoided, and the directional guidance for the searching process of GPHH can be provided while ensuring the diversity of generated DRs and the search space of GPHH. In this case, more excellent DRs can be obtained. The main contributions and innovations of this paper are as follows:

1. An improved GPHH algorithm based on dual feature weight sets is designed to guide the exploration of GPHH through measured feature weights, thereby improving the searching efficiency of the algorithm and automatically generating more promising and understandable DRs.
2. In order to measure feature weights (i.e., feature importance) more accurately in the multi-objective DFJSSP, two feature weight measures are proposed: one based on the fitness values of DRs and another based on the diversity of the Pareto front. Based on these two feature weight measures, the feature set for GPHH can be separated as dual feature weight sets for routing and sequencing decisions, respectively.
3. In order to use the obtained dual feature weight sets more effectively, a novel hybrid population adjustment strategy is also given in this paper. This strategy can adjust and refine the current population based on the feature weights so that the irrelevant and redundant features can be eliminated.
4. By considering total energy consumption and mean tardiness as two optimization objectives [32,33], the effectiveness of the proposed GPHH is demonstrated on an energy-efficient DFJSSP by comparing them with the existing related algorithms. Additionally, the specific behaviors and associated impacts of the dual feature weight sets in the scheduling process are also comprehensively analyzed.

2. Background

This section gives a brief description of the energy-efficient DFJSSP considered in this paper and the basic idea of GPHH.

2.1. Mathematical Description of the Energy-Efficient DFJSSP

The energy-efficient DFJSSP presents as a typical combinatorial optimization problem [33]. Its mathematical description is as follows. There are m machines $\mathcal{M} = \{M_1, \cdots, M_m\}$ utilized for processing jobs. Each machine has a designated standby power MP_k in this job shop. New jobs $\mathcal{J} = \{J_1, \cdots, J_n\}$ randomly arrive over time, and their related information is allocated after their arrival, including arrival time (at_i), due date (dd_i), and the specified operation sequence $\{P_{i1}, \cdots, P_{in_i}\}$ (where n_i represents the number of operations for processing job J_i). Each operation P_{ij} has a designated candidate machine set $\mathcal{M}_{ij}$, and this operation can be processed by any machine $M_k \in \mathcal{M}_{ij}$. In this case, the corresponding processing time is PT_{ijk}, and the corresponding energy consumption is EC_{ijk}. Considering the real production situations, the DFJSSP must adhere to the following constraints:

1. Before jobs arrive at the shop floor, the job-related information is unknown and therefore not taken into consideration during the current machining process.
2. Jobs can only be processed upon reaching the job shop, and all operations must be processed in the order given. Each operation can be performed on only one machine selected from its candidate machine set.
3. Machines can process only one job/operation at a time, and the process cannot be interrupted. Additionally, they are in a standby state when not processing, and they consume energy with standby power.

In this paper, we simultaneously optimize two production objectives in the DFJSSP: Mean Tardiness (MT) and Total Energy Consumption (TEC) [33,34], whose mathematical expressions are shown in Equations (1) and (2). Here, MT is used to ensure that considered jobs can be finished in time, while TEC responds to the energy consumption of the whole job shop, which mainly consists of two parts: energy consumption for machining and energy consumption for being on standby.

$$\min \quad MT = \frac{1}{n}\sum_{i=1}^{n}\max\{c_i - dd_i, 0\}, \tag{1}$$

$$\min \quad TEC = \sum_{k=1}^{m}\left(c_{\max} - \sum_{i=1}^{n}\sum_{j=1}^{n_i} PT_{ijk}\right)MP_k + \sum_{i=1}^{n}\sum_{j=1}^{n_i}\sum_{k=1}^{m} EC_{ijk}, \tag{2}$$

where c_i is the completion time of the job J_i, and $c_{\max} = \max_{i=\{1,\ldots,n\}} c_i$ is the completion time of the last job.

2.2. Solving the DFJSSP Based on GPHH

As a well-known optimization approach, GPHH has been widely used in all kinds of optimization problems, especially the combinatorial optimization problem represented by the JSSP [7,35]. As a hyper-heuristic method, GPHH does not directly find a specific solution to the problem but iteratively generates a set of heuristic rules to guide the solution generation. In the JSSP, GPHH generates DRs, which are used to make scheduling decisions to generate a specific scheduling scheme for solving the JSSP, rather than the scheduling scheme itself like meta-heuristic algorithms do [36,37]. For solving the energy-efficient DFJSSP mentioned in Section 2.1, the DRs generated by GPHH in this paper contain two rules: a routing rule and a sequencing rule, which are used to handle the routing decisions and sequencing decisions, respectively.

The DRs generated by GPHH are usually represented as a tree-based structure, as shown in Figure 1. Such a tree-based structure can help GPHH perform genetic operators, such as replication, crossover, and mutation, to generate new DRs. Figure 1 illustrates an example of the DR with two rules. The leaf nodes at the bottom of the tree are constants or features that represent job shop-related information, such as PT standing for the processing time of the current operation, WIQ standing for the total processing time of all the operations in the waiting queue of a machine, etc. The non-leaf nodes are connected by function operators such as {+, −, ×, / }, etc.

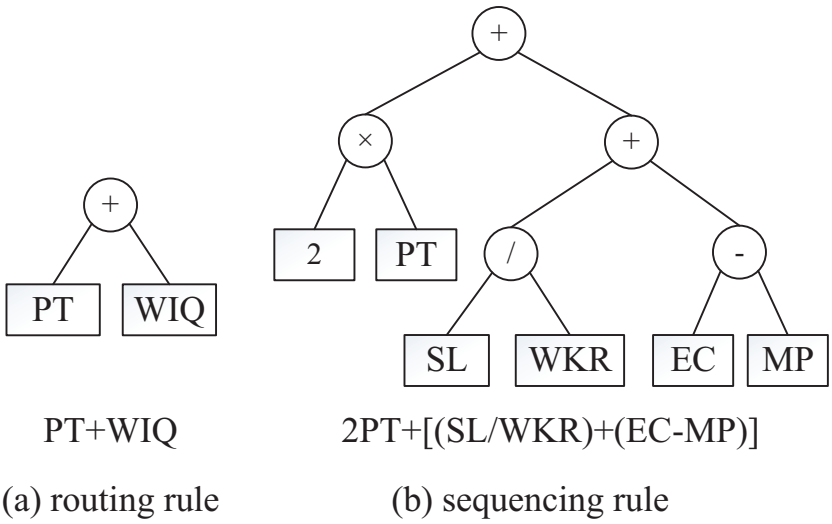


Figure 1. An example of the DR with routing and sequencing rules.

In this paper, only one dynamic event is considered, i.e., random job arrivals, and their related information is unknown until the jobs arrive at the shop floor. A typical scheduling

decision-making process for how to use generated DRs to solve the DFJSSP in real time is given in Figure 2. At each decision point, priority values are computed for each scheduling object (i.e., candidate machines for routing decisions or waiting operations in the machines' queues for sequencing decisions) using DRs, and the scheduling object with the minimum priority value is chosen as the decision output. As shown in Figure 2, when a new job arrives, its related information becomes known, and it enters the waiting area. The routing decision, guided by the routing rule, assigns a machine from the corresponding candidate machine set for the current operation of the job. When the machine is machining, the job will be placed into the queue of this chosen machine. The sequencing decision uses the sequencing rule to designate the next job/operation being processed from the machine's queue immediately after it completes one operation. For instance, in the figure, after the operation P_{12}, which means the second operation of job J_1, is processed on machine M_3 based on routing and sequencing decisions, its next operation P_{13} will be assigned to the queue of machine M_2, waiting for machining. This process continues until all operations are completed, leading to the output of the finished job. Routing and sequencing decisions work together to create a scheduling scheme that adheres to constraints and optimizes production objectives.

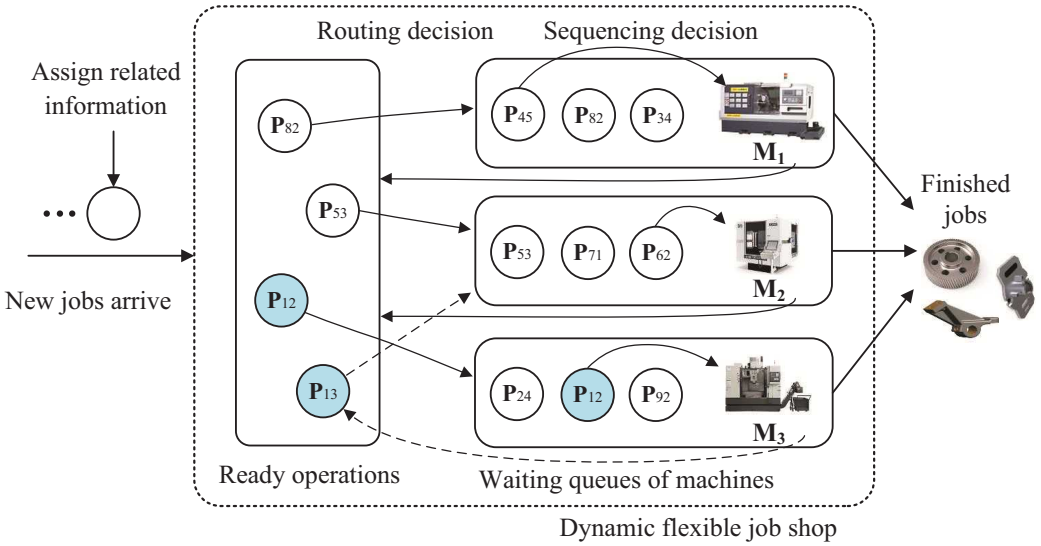


Figure 2. A typical scheduling decision-making process based on DRs.

2.3. Difference between Feature Selection and Feature Weights

It has been proven that features in GPHH are not of equal importance [26]. In fact, including irrelevant features in the feature set could reduce the optimization performance of GPHH. Existing feature selection methods for GPHH focus more on using fewer features to obtain more compact and understandable DRs than on improving their effectiveness. For instance, in previous work [31], more interpretable DRs were evolved by applying feature selection methods in GPHH to only select useful features. However, since the results of feature selection are overly dependent on human-set thresholds, it is easy to mistakenly remove both irrelevant and redundant features and features that are useful to the DFJSSP. In this case, GPHH fails to get enough valid information when generating DRs, which ultimately affects the performance of the algorithm.

In addition, different objectives need different features when solving the multi-objective DFJSSP. It is more difficult to measure the contribution and importance of features in such a situation, which further increases the difficulty of feature selection because existing feature selection methods focus mainly on single-objective problems [31].

Therefore, compared to the feature selection methods, this paper proposes two novel feature weight measures for the multi-objective DFJSSP. These measures assign appropriate weight values to features based on their importance and contribution to solving and

optimizing multi-objective DFJSSP. In doing so, they can effectively avoid the heavy dependence on human-set thresholds in the selection methods. Additionally, feature weights can provide guidance for the evolutionary process of GPHH while ensuring the algorithm's search space. This characteristic could improve the search efficiency of the algorithm so that higher-quality and more interpretable DRs can be generated.

In summary, the improved GPHH with dual feature weight sets proposed in this paper retains the advantages of the feature selection methods (i.e., improving the search efficiency of GPHH and the interpretability of the generated DRs) while avoiding the negative impact of feature selection on the algorithm performance (i.e., guaranteeing the search space of GPHH).

3. GPHH Based on Dual Feature Weight Sets

3.1. Framework of the Proposed GPHH

This paper considers the diversity of decision information needs for routing and sequencing decisions and introduces the concept of dual feature weight sets into GPHH, which essentially generates two different feature weight sets for the routing and sequencing DRs. For this goal, two novel feature weight measures and a hybrid population adjustment strategy are presented to more effectively utilize the useful information from outstanding individuals during the iterative process, thus producing superior DRs.

Figure 3 depicts the overall framework of our proposed GPHH in detail. Compared with traditional GPHH, Figure 3 highlights the proposed feature weight measure module and feature weight utilization module (including the hybrid population adjustment strategy and the mutation operator in Stage 2) in red.

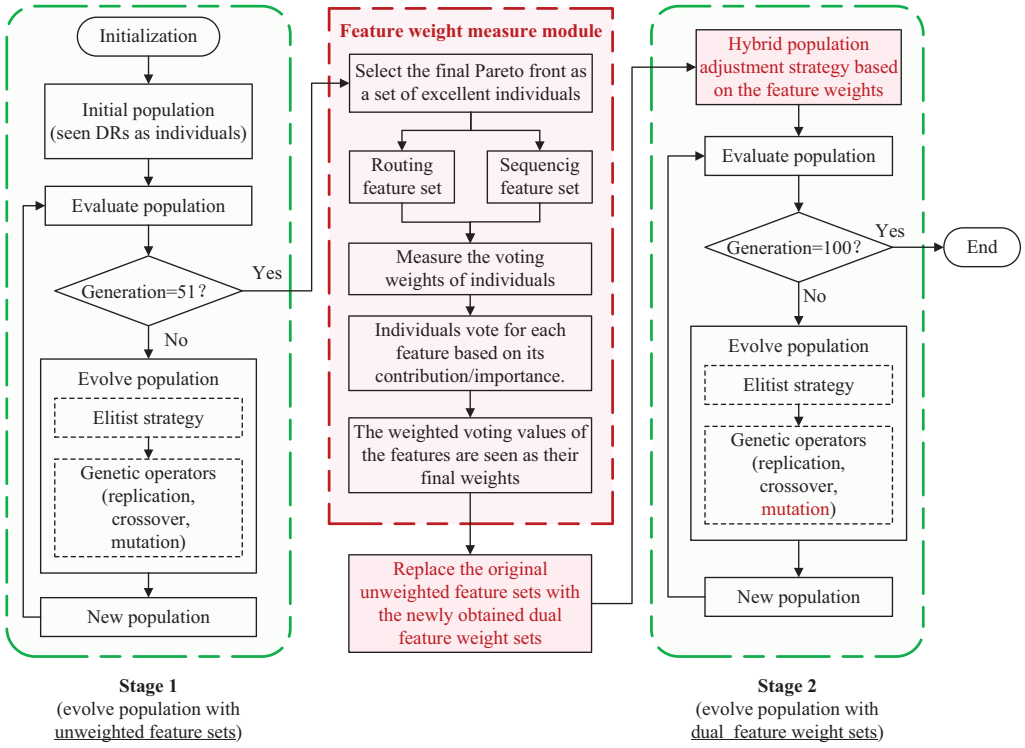


Figure 3. The overall framework of the proposed GPHH with dual feature weight sets.

There are two stages shown in Figure 3. In Stage 1, the algorithm utilizes an unweighted feature set for both routing and sequencing decisions in the initial 50 generations, essentially following the steps of the traditional GPHH. The goal of this stage is to generate a set of high-quality individuals for precise feature weight assessment.

After this, the feature weight measure module calculates the contributions and significance of each feature for two different scheduling decisions, hence forming the rout-

ing/sequencing feature weight set (i.e., the dual feature weight sets). Notably, the Pareto front generated from Stage 1 is directly chosen as a set of excellent and diverse individuals in the feature weight measure modules due to the nature of the multi-objective optimization problem. Here, the feature weight measure module only performs one time during the entire iteration of the improved GPHH for the sake of efficiency, as shown in Figure 3.

The resulting new dual feature weight sets are then applied in Stage 2 of our algorithm for 50 additional generations. Specifically, these sets aid in refining the population from Stage 1 by eliminating irrelevant features, which is called population adjustment in Figure 3. Additionally, the dual feature weight sets can also guide the search and evolution direction of the improved algorithm. Features with higher weights, indicating their importance in optimization performance, are more likely to be selected by genetic operators, enhancing the performance of individuals in the subsequent 50 generations of evolution (i.e., Stage 2). Conversely, features with lower weights, having minimal impact on performance, are less likely to be chosen, also leading to the improved optimization performance of individuals. This kind of mechanism can enhance the effectiveness and interpretability of the proposed GPHH without altering the algorithm's search space.

3.2. Feature Weight Measures for the Multi-Objective DFJSSP

As shown in Figure 3, accurately measuring the weights of features is crucial for guiding GPHH to generate outstanding DRs. This is because the accuracy of measured feature weights will directly affect the subsequent population adjustment strategy and mutation operator, which in turn affects the search direction of the whole algorithm. For this reason, this paper proposes two different feature weight measures suitable for multi-objective optimization problems: fitness-based and diversity-based feature weight measures.

3.2.1. Fitness-Based Feature Weight Measure

Mei et al. [26] presented a feature selection method for GPHH to tackle the single-objective JSSP and pointed out that the importance of each feature was determined by both individuals' fitness values and the contribution of the features themselves. Following this idea, our study focuses on the non-dominated relationships among individuals in the multi-objective DFJSSP and presents a fitness-based weight measure suitable for solving multi-objective problems.

Algorithm 1 gives the pseudo-code of the proposed fitness-based feature weight measure, which essentially is a weighted voting process. First, the Pareto front $\mathcal{R}$ obtained from the 50th generation (viewed as a group of outstanding individuals) and the unweighted feature sets used in the first 50 generations are utilized as inputs. Then, each feature is evaluated one by one for its contribution to the fitness of the individuals in $\mathcal{R}$. If a feature contributes to the fitness value of that individual, the individual will vote for it. Eventually, each feature accumulates all the obtained voting weights as its own weight value, which is used to measure the importance of this feature. Note that the dual feature weight sets mentioned in this paper indicate that the feature weight sets for routing and sequencing decisions are different.

Equation (3) is used to determine the voting weight $w_{fit}(r)$ of an individual r, where there are m objectives, and $fit_i(r)$ represents the fitness value of the DR r for the ith objective. The goal of this study is to minimize the objective values; hence, the lower the fitness value of an individual, the higher its voting weight.

$$w_{fit}(r) = \sum_{i=1}^{m} \frac{\max\{fit_i(r) \mid r \in \mathcal{R}\} - fit_i(r)}{\max\{fit_i(r) \mid r \in \mathcal{R}\} - \min\{fit_i(r) \mid r \in \mathcal{R}\}} \tag{3}$$

Equation (4) defines the contribution of a feature f to an individual r. Here, when a feature is set to 1, it is analogous to removing this feature from the individual. For instance, (PT + WIQ|PT = 1) = 1 + WIQ. The contribution of a feature to an individual is determined by the result in the numerator of Equation (4). If the final $Con(f, r) > 0$, it indicates that

this feature has a positive contribution to the individual, and the individual will vote for this feature.

$$Con(f,r) = \sum_{i=1}^{m} \frac{fit_i(r \mid f=1) - fit_i(r)}{\max\{fit_i(r) \mid r \in \mathcal{R}\} - \min\{fit_i(r) \mid r \in \mathcal{R}\}} \tag{4}$$

Algorithm 1 Fitness-Based Feature Weight Measure

Require: the Pareto front $\mathcal{R}$ obtained from the 50th generation and the unweighted feature sets F_w
Ensure: the dual feature weight sets F_w^*
for all feature $f \in F_w$ **do**
  $vote(f) \leftarrow 0$
  for all individual/DR $r \in \mathcal{R}$ **do**
    calculate the voting weight $w_{fit}(r)$ of this individual r using Equation (3)
    calculate the contribution $Con(f,r)$ of this feature f to this individual r using Equation (4)
    if $Con(f,r) > 0$ **then**
      $vote(f) \leftarrow vote(f) + w_{fit}(r)$
    end if
  end for
end for ▷ `Each feature has been assigned a weight value.`
$F_w^* \leftarrow$ normalized feature weights for all $f \in F_w$
return F_w^*

3.2.2. Sparsity-Based Feature Weight Measure

Considering that the goal of solving multi-objective problems is to obtain a more widely distributed and diverse Pareto front, in addition to the above proposed fitness-based feature weight measure, this paper also proposes a feature weight measure based on the sparsity of the Pareto front to evaluate the importance/contribution of features. This method pays more attention to the contribution of both features and individuals to the entire Pareto front, whose pseudo-code is given in Algorithm 2.

Algorithm 2 Sparsity-Based Feature Weight Measure

Require: the Pareto front $\mathcal{R}$ obtained from the 50th generation and the unweighted feature sets F_w
Ensure: the dual feature weight sets F_w^*
for all feature $f \in F_w$ **do**
  $vote(f) \leftarrow 0$
  for all individual/DR $r \in \mathcal{R}$ **do**
    calculate the voting weight $w_{spa}(r)$ of this individual r using Equation (5)
    set feature f to 1 in individual r, denoted as $r \mid f=1$, and calculate $fit_i(r \mid f=1)$
    if there exists an individual/DR in $\mathcal{R}$ dominating $r \mid f=1$, **then**
      $Con(f,r) = 1$
    else
      $Con(f,r) = 0$
    end if
    if $Con(f,r) > 0$ **then**
      $vote(f) \leftarrow vote(f) + w_{spar}(r)$
    end if
  end for
end for ▷ `Each feature has been assigned a weight value.`
$F_w^* \leftarrow$ normalized feature weights for all $f \in F_w$
return F_w^*

Equation (5) describes the sparsity of an individual as its voting weight, where $fit_i(r+1) - fit_i(r-1)$ represents the difference in fitness values between two individuals adjacent to the individual r. The advantage of this feature weight measure is that it can more evenly assign voting weights to all individuals in the Pareto front, helping to enhance the optimization performance of multi-objective problems considering the non-dominance relationship. Hence, small voting weights are given to individuals that are clustered too close in a very small region, thus ensuring that the final individuals obtained in the Pareto front are uniformly distributed.

$$w_{spa}(r) = \sum_{i=1}^{m} \frac{\mid fit_i(r+1) - fit_i(r-1) \mid}{\max\{fit_i(r) \mid r \in \mathcal{R}\} - \min\{fit_i(r) \mid r \in \mathcal{R}\}} \tag{5}$$

When calculating the contribution $Con(f, r)$ of a feature f to an individual r, the first step is to compute $fit_i(r \mid f = 1)$. If $r \mid f = 1$ is dominated by any existing individual in the Pareto front, the feature f is considered to have a positive contribution to the individual r. The condition for domination is defined as: for $\forall i \in \{1, 2, \cdots, m\}$, $fit_i(r) \leq fit_i(r \mid f = 1)$, and there exists at least one $i \in \{1, 2, \cdots, m\}$, such that $fit_i(r) < fit_i(r \mid f = 1)$.

3.3. Hybrid Population Adjustment Strategy Based on Feature Weights

After completing the measure of feature weights, less important features will be assigned lower weight values. If a feature is given a weight of zero, it can be identified as an irrelevant feature to the problem. Although GPHH itself has the inherent ability to select suitable features during the iteration, it is still challenging to effectively eliminate these irrelevant features solely through genetic operators such as crossover and mutation. Allowing these irrelevant features to persist in individuals will lead to the contamination of the whole population, resulting in more individuals with redundant, duplicated, and unnecessary features. This hinders the further evolution of the population and the optimization performance of generated DRs.

To this end, the existing studies focusing on feature selection methods propose the following three population adjustment strategies [31], which remove unselected features from the current population and improve the interpretability of the resulting DRs based on the selected feature set.

1. Simply replace each unselected feature with a constant of 1. This strategy is proposed based on the above calculation of the feature contribution $Con(f, r)$, which can maximally retain the structure and performance of the current good individuals.
2. Replace unselected features with other selected features. However, such random substitution may change the behavior of good individuals in some aspects, affecting their final optimization performance.
3. Directly use the resulting selected feature set to randomly initialize the population. This way, the effective information of good individuals in the current population will be lost.

Related studies using feature selection methods to solve single-objective problems [31] have shown that the first strategy can better improve the performance of the algorithm compared with the third strategy. In addition, Section 3.2 of this paper also adopts the basic idea of calculating the feature contribution $Con(f, r)$ in the first strategy as the basis of the proposed feature weight measures. Hence, this method not only helps to eliminate redundant information and retain the individuals' structure, but also may improve the performance of these individuals.

Therefore, on the basis of the above existing work, this paper proposes a novel hybrid population adjustment strategy based on the obtained dual feature weight sets. In this hybrid strategy, the first and third strategies are both applied to adjust the current population so that the irrelevant features (whose weights are zero) can be removed and the search efficiency can be enhanced.

Algorithm 3 gives the pseudo-code of this hybrid population adjustment strategy. Specifically, Algorithm 3 only removes irrelevant features from the top 20% of individuals

using the first strategy to retain the valid information of good individuals. For the remaining 80% of non-excellent individuals, they are directly deleted and regenerated based on the obtained dual feature weight sets using the third strategy to improve the diversity of individuals in the population while deleting irrelevant features.

Algorithm 3 Hybrid Population Adjustment Strategy Based on Feature Weights

Require: the population $\mathcal{P}$ generated from the 50th generation and the obtained dual feature weight sets F_w^*
Ensure: the new population $\mathcal{P}^*$
sort individuals by the non-dominated rank and crowding distance
for all individual/DR $r \in \mathcal{P}$ **do**
  if r is the top 20% of all the individuals in $\mathcal{P}$ **then**
    for all feature $f \in F_w^*$ **do**
      if $weight(f) < 0.0001$ **then**
        replace feature f with a constant of 1
        ▷ `This is the first strategy.`
      end if
    end for
  else
    use F_w^* to randomly generate a new individual and replace the old one
    ▷ `This is the third strategy.`
  end if
end for
return $\mathcal{P}^*$

4. Experimental Design

4.1. Simulation Model Design

The configuration of the energy-efficient DFJSSP simulation model adopted in this paper is shown in Table 1, which has been extensively employed in previous research [30,33,38].

Table 1. Configuration of the energy-efficient DFJSSP simulation model.

Parameter	Setting
Number of machines m	10
Number of arrived jobs n	2000
Number of warm-up jobs	500
Number of operations per job	discrete $U(1, 10)$
Available machines per operation	discrete $U(1, 10)$
Job arrival process	Poisson process
Utilization level u	0.85, 0.95
Due date factor α	2, 4, 6
Mean processing time $\overline{PT}_{ij}$	discrete $U(1, 99)$
Mean energy consumption $\overline{EC}_{ij}$	discrete $U(1, 99)$
Standby power MP_k	{10, 12.5, 4.5, 3.6, 7.0, 1.5, 8.5, 2.2, 22.9, 6.4}

According to existing research [35,39], the machine utilization level u and the due date factor α are the primary factors defining the workload of different job shop environments. Therefore, this paper considers two machine utilization levels {0.85, 0.95} and three due date factors {2, 4, 6}, resulting in a total of 2 × 3 = 6 different production scenarios. During the training phase, the GPHH reruns independently 30 times under each scenario using different random seeds. To evaluate the effectiveness of the generated DRs, the results of each run undergo 100 independent tests. The average of these 100 test results is then taken as the final fitness value of the obtained DRs, thereby validating their real generality and applicability.

4.2. Parameter Settings of GPHH

Table 2 presents the initial feature set of GPHH in this paper, consisting of 28 features related to the JSSP. These features are commonly utilized in existing research [33] and encompass various aspects such as information related to jobs, machines, and the job shop itself.

Table 2. Feature set of the GPHH.

No.	Feature	Description
1	PT	Processing time PT_{ijk} of operation P_{ij} on M_k
2	SL	Slack of the job J_i
3	OWT	Waiting time of operation P_{ij} since ready
4	NPT	Processing time of the next operation P_{ij+1}
5	WKR	Work remaining for the job J_i
6	TIS	Time of the job J_i in job shop
7	WIQ	Workload in the queue of the machine M_k
8	MWT	Waiting time of the machine M_k since ready
9	DD	Due date of the job J_i
10	MRT	Ready time of the machine M_k
11	ORT	Ready time of the operation P_{ij}
12	WINQ	Workload in the queue of the next machine
13	AT	Arrival time of the job J_i
14	NRT	Ready time of the next machine
15	EC	Energy consumption EC_{ijk} of operation P_{ij} on M_k
16	NEC	Energy consumption of the next operation P_{ij+1}
17	ECR	Energy consumption remaining for the job J_i
18	EIQ	Total energy consumption of all operations in the queue of the next machine M_k
19	EINQ	Total energy consumption of all operations in the queue of machine M_k
20	MP	Standby power of machine M_k
21	NOR	Number of remaining operations of the job J_i
22	NOS	Number of optional machines for the operation P_{ij}
23	NIQ	Number of operations in the queue of machine M_k
24	NINQ	Number of operations in the queue of next machine
25	NOPS	Number of operations of the job J_i
26	RPT	Relative processing time = $\frac{PT_{ijk}}{\min\{PT_{ijk} \mid k=1,\ldots,\lvert\mathcal{M}_{ij}\rvert\}}$
27	REC	Relative energy consumption = $\frac{EC_{ijk}}{\min\{EC_{ijk} \mid k=1,\ldots,\lvert\mathcal{M}_{ij}\rvert\}}$
28	RMP	Relative standby power = $\frac{MP_k}{\min\{MP_k \mid k=1,\ldots,m\}}$

The function set includes {+, −, ×, /, max, min} [27,33], where the division returns one if divided by zero. Additionally, max/min are functions that take two inputs and return their maximum/minimum values, respectively.

Table 3 shows the parameter setting of the GPHH in this section [27,33]. The feature weights are measured at the 51st generation, and this process is executed only once in the whole algorithm.

Table 3. Parameter settings of GPHH.

Parameter	Setting
Initialization	Ramped half-and-half
Population size	600
Maximum depth of DRs	8
Crossover rates	80%
Mutation rates	15%
Reproduction rates	5%
Parent selection	Tournament selection with size 7
Elitism	10 best individuals
Number of generations	101
Feature weight measure	At the 51st generation
Population adjustment strategy	See Algorithm 3

4.3. Comparison Design

In order to effectively analyze the performance of our proposed GPHH with dual feature sets, which considers feature weights during the iterations, three existing algorithms are compared in this section. In this way, a total of five improved GPHH algorithms are compared in this section as follows.

1. GPLWT [40] (i.e., GPHH-LWT) uses the Least Waiting Time (LWT) as its routing rule, and the sequencing rule is generated by GPHH. In other words, this algorithm only automatically evolves sequencing rules via GPHH while its routing rule is fixed. In this section, it is considered as the baseline for solving the energy-efficient DFJSSP.
2. GPDR [33] (i.e., GPHH-Delayed-Routing) is one of the current state-of-the-art algorithms, which adopts both multi-tree representation [39] to generate routing and sequencing rules simultaneously and a delayed routing strategy to ensure the timeliness of the feature information at the decision-making point. In this paper, GPDR is used as the basic algorithm, in which the proposed feature weight measures and the hybrid population adjustment strategy are introduced.
3. GPFS [31] (i.e., GPHH-Feature-Selection) is the GPHH algorithm that incorporates a feature selection method. In their work, GPFS is only used to solve single-objective problems. It is applied to multi-objective problems based on the linear weighting method in this section so that its performance can be analyzed.
4. GPFW(fit) (i.e., GPHH-Feature-Weight based on fitness) is the GPDR with dual feature sets that adopts the feature weight measure based on fitness values of individuals (see Section 3.2.1) and the hybrid population adjustment strategy proposed in Section 3.3.
5. GPFW(spa) (i.e., GPHH-Feature-Weight based on sparsity) is the GPDR with dual feature sets that adopts the feature weight measure based on sparsity of the Pareto front (see Section 3.2.2) and the hybrid population adjustment strategy proposed in Section 3.3.

4.4. Performance Measures for Comparison

When assessing the quality of the Pareto front obtained by the algorithms, this paper employs two commonly used performance metrics for multi-objective optimization problems: Hyper-Volume (*HV*) and Inverted Generational Distance (*IGD*) [41,42]. Their calculation formulas are as follows:

$$HV = \bigcup_{i=1}^{n_{\mathrm{PF}}} v_i, \tag{6}$$

$$IGD = \frac{1}{n_{\mathrm{APF}}} \left(\sum_{i=1}^{n_{\mathrm{APF}}} d_i \right), \tag{7}$$

where n_{PF} is the number of individuals in the Pareto front, v_i is the hyper-volume formed by individuals in the Pareto front and a reference point, n_{APF} is the number of individuals in the true Pareto front, and d_i is the minimum Euclidean distance from individuals in the true Pareto front to the Pareto front obtained by algorithms.

An outstanding algorithm is expected to have a higher *HV* value and a lower *IGD* value, indicating that the Pareto front generated by this algorithm is closer to the true Pareto front and more evenly distributed. Normalization is essential when computing *HV* and *IGD*. As the true Pareto front is unknown in our energy-efficient DFJSSP, the results from all algorithms in this paper are amalgamated, and the resulting Pareto front is utilized as an approximate true Pareto front. *HV* also necessitates a reference point, set as $1 + 1/(n_{\mathrm{APF}} - 1)$ [43], where n_{APF} represents the number of individuals in the approximate true Pareto front.

5. Experimental Results Analysis

5.1. Overall Performance Analysis of the Algorithms

Tables 4 and 5 show the mean and standard deviation of the *HV* and *IGD* values obtained by the different algorithms after 30 independent runs for optimizing both *MT* and *TEC* in six different scenarios. Additionally, Wilcoxon rank-sum tests were conducted (at a 5% confidence level) to further analyze the significance of their optimization performance [44]. Here, GPFW(fit) and GPFW(spa) are compared with three existing algorithms (i.e., GPLWT, GPDR, and GPFS). If the *p*-value of the Wilcoxon rank-sum test is less than 0.05 for all of them, the *significant differences* are marked in bold in the tables.

Table 4. Mean (standard deviation) of *HV* values for different algorithms in six scenarios.

Algorithms	<0.85–2>	<0.85–4>	<0.85–6>	<0.95–2>	<0.95–4>	<0.95–6>
GPLWT	0.324 (0.001)	0.549 (0.000)	0.602 (0.000)	0.392 (0.001)	0.578 (0.001)	0.668 (0.001)
GPDR	0.977 (0.010)	1.000 (0.007)	1.024 (0.005)	0.970 (0.009)	1.005 (0.005)	1.027 (0.006)
GPFS	0.923 (0.101)	0.996 (0.018)	1.024 (0.004)	0.957 (0.033)	0.994 (0.024)	1.017 (0.019)
GPFW(fit)	**0.985 (0.008)**	1.002 (0.005)	1.024 (0.005)	**0.982 (0.010)**	1.011 (0.006)	1.029 (0.005)
GPFW(spa)	**0.984 (0.009)**	1.002 (0.006)	1.024 (0.005)	**0.983 (0.012)**	**1.012 (0.006)**	1.029 (0.005)

Under the Wilcoxon rank-sum test with the significance level of 0.05, the significantly better results are marked in bold.

Table 5. Mean (standard deviation) of *IGD* values for different algorithms in six scenarios.

Algorithms	<0.85–2>	<0.85–4>	<0.85–6>	<0.95–2>	<0.95–4>	<0.95–6>
GPLWT	0.538 (0.001)	0.420 (0.000)	0.398 (0.000)	0.457 (0.001)	0.372 (0.001)	0.325 (0.001)
GPDR	0.014 (0.005)	0.025 (0.006)	0.047 (0.010)	0.025 (0.006)	0.044 (0.013)	0.032 (0.007)
GPFS	0.044 (0.074)	0.025 (0.012)	0.045 (0.010)	0.034 (0.021)	0.045 (0.020)	0.036 (0.012)
GPFW(fit)	**0.009 (0.006)**	0.024 (0.005)	0.046 (0.008)	0.019 (0.006)	**0.037 (0.014)**	0.031 (0.007)
GPFW(spa)	**0.009 (0.005)**	0.023 (0.006)	0.047 (0.008)	**0.018 (0.005)**	0.040 (0.010)	0.030 (0.005)

Under the Wilcoxon rank-sum test with the significance level of 0.05, the significantly better results are marked in bold.

Overall, one can see that GPLWT is the worst among the five algorithms. Of the three existing algorithms, GPDR performs the best. As for the two algorithms proposed in this paper, GPFW(fit) and GPFW(spa) both show better optimization performance when the problem is difficult (see scenarios <0.85–2> and <0.95–2> in Tables 4 and 5).

From these tables, it can be observed that the performance of all algorithms is significantly better than GPLWT, indicating that using routing rules generated by GPHH is more effective than using a fixed LWT rule. This is because GPHH can generate appropriate routing rules based on the current job shop state for routing decisions, thereby creating more rational waiting queues for the machines and enhancing the following sequencing decisions.

Among the three existing GPHH algorithms, GPDR and GPFS are both the current state-of-the-art algorithms that consider the influence of GPHH's feature sets. It is pointed out that GPFS can filter out insignificant features in single-objective problems, making the generated DRs more interpretable and similar performance-wise to traditional GPHH by utilizing fewer features [31]. In this section, GPFS is applied to multi-objective problems using the linear weighting method. The results shown in Tables 4 and 5 indicate that the performance of GPFS is inferior to GPDR, with a notably higher standard deviation in most scenarios, especially in situations with a tighter due date factor such as <0.85–2> and <0.95–2>. This suggests that a simple transfer of the GPFS to multi-objective problems through the linear weighting method is not appropriate. The selected features cannot satisfy the requirements of multi-objective optimization well, thus impacting the algorithm's optimization performance and stability.

When comparing the two algorithms proposed in this paper with three existing algorithms, especially GPDR, it can be seen that the proposed algorithms are better than the other algorithms overall, especially in the scenarios of <0.85–2> and <0.95–2>, which show significant advantages. This phenomenon suggests that the proposed fitness-based and sparsity-based feature weight measures can more accurately calculate the importance/contribution of each feature to solving the multi-objective optimization problem, which further improves the algorithm's search efficiency and optimization performance. Tables 4 and 5 also show that the proposed algorithm has limited performance improvement relative to the basic algorithm (i.e., GPDR), especially in the case of easier problems. This is due to the fact that the GPHH itself has a certain function of feature selection, because the performance of the DRs that incorporate irrelevant and non-essential features is not good, and thus it will be eliminated by the GPHH in the process of iterative evolution. Therefore, even with the introduction of the dual feature weight sets, the improved GPFWs (i.e., GPFW(fit) and GPFW(spa)) will not have a significantly larger performance improvement.

The comparison between GPFW(fit) and GPFW(spa) reveals that their results are very close to each other, with no statistically significant differences. In theory, the sparsity-based feature weight measure may be slightly better than the fitness-based method, as the former takes into account the non-dominated relationships between individuals. This suggests that the Pareto front obtained after 50 iterations of GPHH is good enough to be used to accurately measure the feature weights.

5.2. Comparison of Training and Testing Time

Training and testing time are two important measures of the efficiency of an algorithm. Hence, the average training time for 30 independent runs of all five algorithms is given in Table 6. One can see that GPLWT has the shortest training time because it uses a fixed routing rule (i.e., LWT) and is less computationally intensive. Although GPFS, GPFW(fit), and GPFW(spa) all need extra steps to select features or measure feature weights, there is no significant difference between these algorithms and GPDR. This indicates that the introduction of the feature selection module or feature weight measure module does not bring too much additional computation cost and time to the algorithms, and the overall training time consumed is still acceptable.

Table 6. Mean training time of 30 independent runs for all five algorithms (s).

Algorithms	**<0.85–2>**	**<0.85–4>**	**<0.85–6>**	**<0.95–2>**	**<0.95–4>**	**<0.95–6>**
GPLWT	5124	4988	5032	4587	4602	4495
GPDR	11,745	11,568	11,814	10,656	11,948	12,530
GPFS	12,670	11,923	11,086	9657	9502	10,183
GPFW(fit)	13,029	12,383	13,441	10,122	10,590	10,097
GPFW(spa)	10,741	10,375	13,697	11,638	13,791	12,864

Table 7 gives the average testing time of the DRs obtained by different algorithms after testing. It can be observed that these DRs take only about 20 ms (or less) to complete all the scheduling decisions in one single simulation. In this case, the decision time spent by these DRs will be even less for each decision point, thus meeting the requirement of real-time decision-making.

Table 7. Mean testing time for DRs obtained by different algorithms (ms).

Algorithms	<0.85–2>	<0.85–4>	<0.85–6>	<0.95–2>	<0.95–4>	<0.95–6>
GPLWT	3.90	3.00	3.00	4.40	4.04	3.43
GPDR	14.36	15.58	16.17	14.34	16.93	16.50
GPFS	11.31	15.08	13.26	12.99	13.74	13.95
GPFW(fit)	21.00	20.95	23.02	19.45	22.10	25.04
GPFW(spa)	15.03	15.90	14.45	14.88	16.73	17.42

6. Behavior Analysis of the Proposed Algorithms

6.1. Analysis of the Number of Unique Features

The number of unique features refers to the quantity of different types of features included in the DRs [31,45]. In this study, the quantity of unique features existing in the final population is analyzed, which should positively contribute to the performance of the proposed algorithms. The specific calculation involves examining the number of different types of features present in the final population. A lower number of unique features indicates that the DRs formed by these features are more concise, possess better interpretability, and are consequently more comprehensible to humans.

Figure 4 illustrates the mean and standard deviation of the number of unique features in the DRs generated by different algorithms over six scenarios. Since GPLWT only evolves sequencing rules, it is excluded from the analysis in this section. In this context, GPFS demonstrates a relatively lower number of unique features due to its aim of generating high-quality DRs using a smaller feature set. The number of unique features of GPFS is mainly determined by the threshold value set during the feature selection, where features are selected if their voting weight is greater than half of the total voting weight [31]. Consequently, features with weights below half of the total voting weight are removed, leading to a significant reduction in the number of feature types. However, this reduction also results in a decrease in the optimization performance of the algorithm.

In contrast, our proposed feature weight measures do not directly remove features like GPFS does. Only irrelevant features with a weight of zero are completely removed in the hybrid population adjustment strategy. Additionally, the obtained dual feature weight sets can guide the evolutionary direction of GPHH, as less important features may not appear in the final population. This also contributes to a reduction in the number of unique features. Therefore, the results in Figure 4 suggest that the proposed GPFWs with dual feature weight sets can effectively decrease the number of unique features in generated DRs and even improve the performance of GPHH. This reduction contributes to the evolution of more readable and interpretable energy-efficient DRs. Furthermore, it underscores the rationale and effectiveness of introducing feature weights into the GPHH, since the algorithm does benefit a lot, even in complex multi-objective environments.

Additionally, it can be observed from Figure 4 that the number of unique features for sequencing rules is optimized more easily. One possible reason is that sequencing decisions require relatively less decision information. Sequencing decisions are typically more concerned with job/operation-related information, since the scheduling objects of a sequencing decision are the job/operations in the machines' queues. In contrast, routing rules need to determine which candidate machine is suitable for processing the current job/operation, taking into account the availability of the machine, the status of the processing queue, and other factors in addition to the characteristics of the job/operation itself. This makes the routing rules require more complex decision information (i.e., a

greater number of unique features) to assist in decision-making. Hence, the number of unique features for routing rules is relatively higher, making more complex routing decisions.

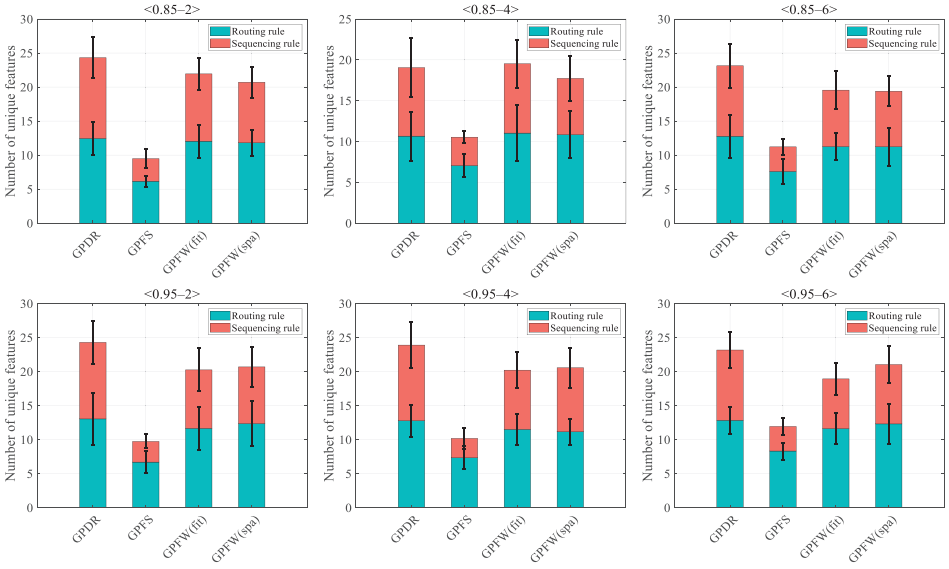


Figure 4. The numbers of unique features of both routing and sequencing rules obtained by different algorithms.

6.2. Analysis of the Rule Sizes of DRs

Rule size is another key indicator that describes the readability and interpretability of DRs. Typically, the smaller the rule size, the lower the depth of the rule, the simpler the structure of the rule, and the easier it is understand.

Figure 5 gives the rule sizes of the DRs obtained by each algorithm. Overall, the rule sizes of DRS generated by these algorithms are not significantly different. Through the introduction of feature selection methods or feature weight measures, the rule sizes of DRs obtained from GPFS and the proposed GPFWs are not significantly reduced, and they only show advantages in a few scenarios, e.g., <0.95–6>.

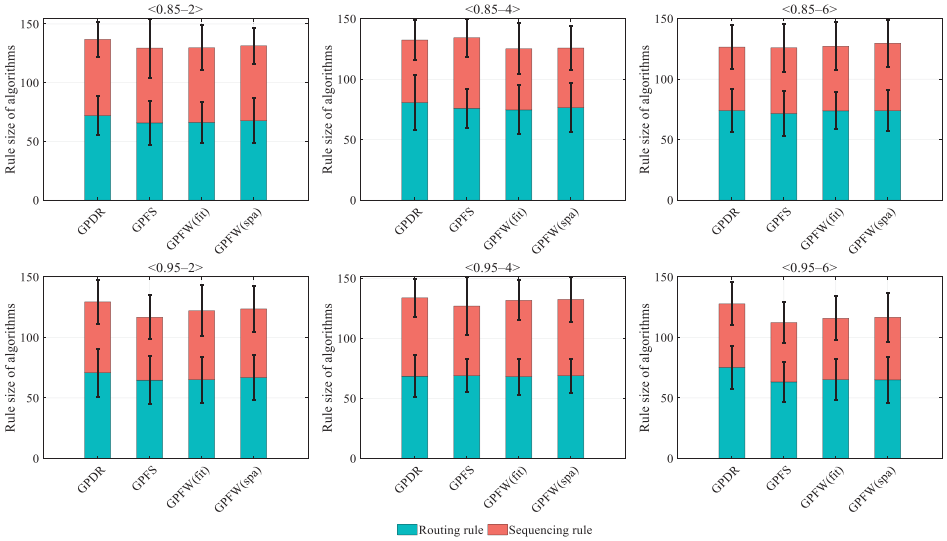


Figure 5. The rule sizes of both routing and sequencing rules obtained by different algorithms.

This result is also expected because what really reduces the rule size is the initialization method (i.e., the third strategy) within the population adjustment strategy. The top 20% of outstanding individuals, although inheriting the valid information from the first

50 generations, also have relatively large rule sizes and are more likely to be selected by genetic operators in the next 50 generations, thus leading to a rise in the rule size of other individuals in the population.

6.3. Conjoint Analysis of Feature Weights and Feature Frequencies

In order to gain insight into the importance/contribution of different features in the evolutionary process of GPHH, this section takes the examples of GPFW(fit) and GPFW(spa) to conduct a conjoint analysis of feature weights and feature frequencies. Figures 6 and 7 present the feature weights and frequencies of routing/sequencing rules, respectively, in six scenarios using these two algorithms, and the statistics are the average of the results of 30 independent runs.

1. Gen50 counts the frequency of each feature in the individuals from the Pareto front that GPFW(fit) and GPFW(spa) have iterated to the 50th generation. Because the first 50 generations are the same for these two algorithms, the same results are obtained.
2. Gen51(fit) and Gen51(spa) denote the feature weights obtained by the fitness-based and sparsity-based feature weight measures, respectively, at generation 51.
3. Gen100(fit) and Gen100(spa) denote the frequency of each feature in the individuals from the final obtained Pareto front after 100 generations of GPFW(fit) and GPFW(spa), respectively.

Figures 6 and 7 unambiguously demonstrate that the results of Gen50, Gen51(fit), Gen51(spa), Gen100(fit), and Gen100(spa) are generally congruent, indicating the correlation between feature weights and feature frequencies. The feature frequencies in Gen50 align well with the feature weights obtained in Gen51(fit) and Gen51(spa), suggesting that feature frequencies can also, to some extent, measure the importance of features. On the other hand, this indirectly implies that GPHH itself has the feature selection ability. The Pareto front achieved after 50 generations is excellent enough to serve as a set of outstanding individuals for feature weight measures.

The minor difference between the results of Gen51(fit) and Gen51(spa) suggests a strong correlation between them, offering diverse viewpoints on the importance of features in solving multi-objective problems. This conclusion is consistent with the findings in Tables 4 and 5, suggesting that, when the results of feature weight measures are similar, the final optimization performance also tends to be similar.

When contrasted with the feature frequencies in Gen50, the results in Gen100(fit) and Gen100(spa) show that features with higher weights have higher frequencies, while lower-weighted features have lower frequencies. This implies that the obtained dual feature weight sets indeed guide the algorithm's search direction in the latter 50 generations. The proposed hybrid population adjustment strategy based on feature weights aids the algorithm in mitigating the negative effects of irrelevant features.

Another observation from Figures 6 and 7 is that the results of most features' weights and frequencies are quite similar in all scenarios. Specifically, features such as PT, MP, and their respective related features RPT, RMP, and WIQ, MWT, NIQ, etc., demonstrate a high level of importance in the generation of routing rules. This aligns with the actual situation, as PT, MP, RPT, and RMP play critical roles in optimizing both MT and TEC. Combining them with features like WIQ, MWT, and NIQ enables a more effective and rational allocation of machine resources, leading to increased speed in job processing. This also ensures that jobs are scheduled on machines with higher MP, thereby reducing their energy consumption from being idle.

From the perspective of sequencing rules, PT, SL, and WKR are three important features. Previous studies [26,27] have demonstrated their significant role in optimizing MT. Xu et al.'s research [33] also highlights the importance of these features in optimizing energy consumption related objectives. An interesting observation is that WKR (i.e., the remaining work of the job) and NOR (i.e., the remaining number of jobs) are functionally correlated, both related to the remaining work of jobs. The results reveal a competitive relationship between these two features: when the weight/frequency of WKR is high,

the weight/frequency of NOR is low, and vice versa, though they can also both be high simultaneously. This implies inherent correlations among different features, emphasizing the necessity of introducing the feature weight measures in this study.

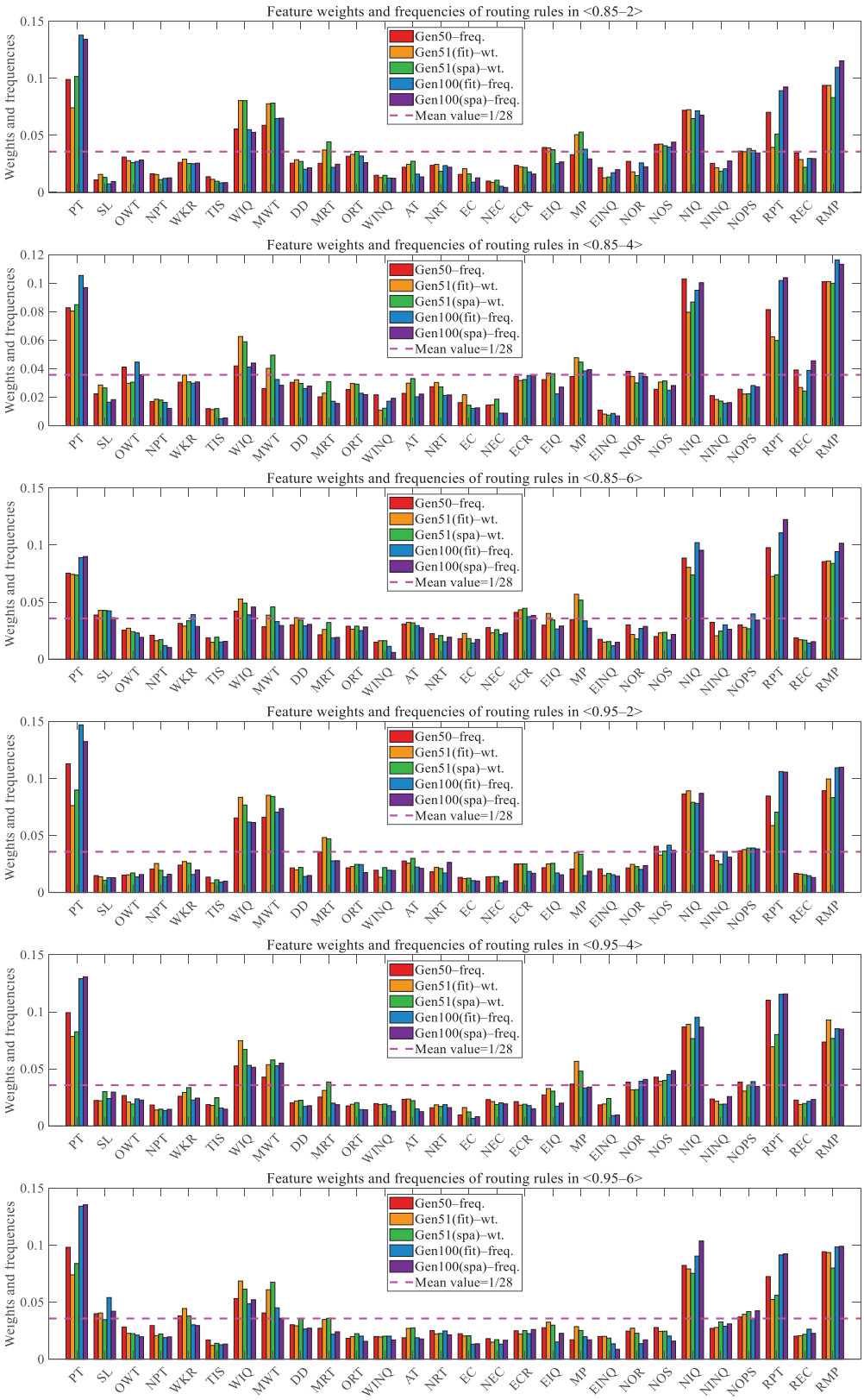


Figure 6. The feature weights and frequencies of routing rules in six scenarios.

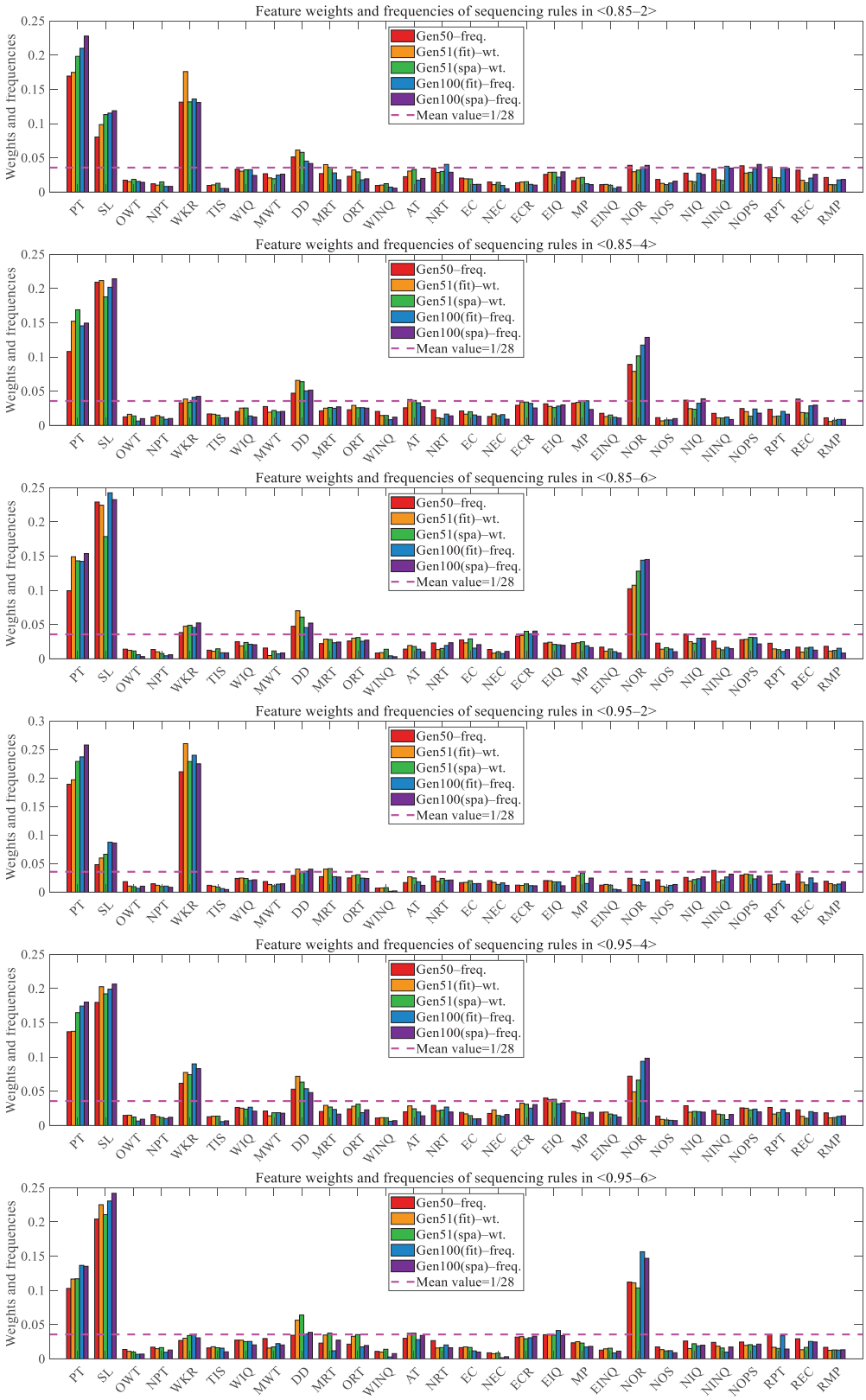


Figure 7. The feature weights and frequencies of sequencing rules in six scenarios.

7. Comprehensive Discussion of the Experimental Results

As numerous experimental results have been presented in Sections 5 and 6, this section offers a comprehensive discussion of the results.

First of all, the experimental results in Sections 5.1 and 5.2 point out that our improved GPHH algorithms (i.e., GPFW(fit) and GPFW(spa)) outperform existing methods (i.e., GPLWT, GPDR, and GPFS) with slightly longer training and testing times. This indicates that using feature weights to guide the evolutionary process of GPHH does benefit the optimization performance when compared to GPDR and surpasses the feature selection method proposed in existing work [31], which focuses more on single-objective problems. Additionally, if we take the experimental results for the number of unique features (in Section 6.1) and the rule size (in Section 6.2) into consideration, one can see that our algorithms not only perform better but also produce more interpretable DRs. This is because the improved GPHH generates DRs with fewer unique features and smaller rule sizes after eliminating irrelevant and redundant features, making the DRs easier to understand.

Considering that the occurrence frequency of each feature in a rule can reflect to some extent the importance or contribution of the feature to the optimization objective, a conjoint analysis of feature weights and feature frequencies is also given in Section 6.3. From the experimental results, one can see that feature frequencies and computed feature weights mostly align. This is because features useful for the optimization objective have larger contributions in individuals/DRs and also occur more frequently. However, feature frequencies can be influenced by irrelevant and redundant features in individuals, limiting the rationality of the results.

In summary, all the above experiments demonstrate that the proposed feature weight measures can more effectively and rationally calculate the significance or contribution of features in the multi-objective DFJSSP compared to existing methods. Hence, they can effectively improve the optimization performance of GPHH and the interpretation of generated DRs by removing irrelevant and redundant features and increasing the occurrence frequencies of high-weight features. Further experiments also point out that the proposed feature weights, in contrast to feature frequencies, better capture the importance and contribution of each feature to the optimization objectives. This is because the feature weights take into account the validity of both individuals and features for the Pareto front.

8. Conclusions and Future Work

This paper addresses the negative impact of irrelevant features on scheduling decisions made by GPHH when generating scheduling rules, as well as the different demands of routing and sequencing scheduling decisions on different feature information. An improved GPHH with dual feature weight sets is proposed and applied to the energy-efficient multi-objective DFJSSP so that high-quality and understandable DRs can be automatically developed.

Specifically, this paper investigates the importance/contribution of different features in solving multi-objective problems and evolving high-quality routing and sequencing rules with GPHH. Then, an improved GPHH with dual feature weight sets is proposed. During the iteration of the population, the importance/contribution of each feature is evaluated based on the fitness of individuals or the sparsity of the Pareto front. In this way, different weights are assigned to the features, which form two different feature weight sets for both routing and sequencing decisions. After this, a hybrid population adjustment strategy is also proposed to genetically update the population based on these newly obtained dual feature weight sets. Experimental results indicate that the proposed improved GPHH with dual feature weight sets surpasses existing related GPHH algorithms and their feature selection methods. The DRs generated by our proposed GPFWs can achieve better optimization performance and interpretability when dealing with multi-objective problems.

In future work, it will be worthwhile to explore the impact of mutation probability on the optimization performance of this proposed GPFW, extend the proposed feature weight measures to many-objective problems, and further improve the population adjustment

strategy. These directions could provide valuable exploration for enhancing the algorithm's performance and broader applications.

Author Contributions: Conceptualization, B.X. and B.F.; methodology, K.X. and B.F.; software, B.F.; validation, B.X., K.X., and B.F.; data curation, D.H.; writing—original draft preparation, K.X. and B.F.; writing—review and editing, D.H. and L.T.; supervision, B.X. and Y.W.; funding acquisition, B.X., L.T., and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation (NNSF) of China (Grant No. 62203011), the Young and Middle-aged Teachers Training Action Project of Anhui Province (Grant No. JNFX2023017), the Science and Technology Project of Wuhu (Grant No. 2023jc05), the Scientific Research Foundation for Introduced Talent Scholars, Anhui Polytechnic University (Grant No. 2020YQQ041), the Pre-research Project of the National Natural Science Foundation (NNSF), Anhui Polytechnic University (Grant Nos. Xjky2022042 and Xjky2022048), the Open Fund of Anhui Automotive Display Integrated System Engineering Research Center (Grant No. VDIS2023B02), the Open Fund Key Project of Anhui Provincial Key Laboratory of Detection Technology and Energy Saving Device (Grant No. JCKJ2022A09).

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank the anonymous reviewers and the editor for their positive comments.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GPHH	Genetic Programming Hyper-Heuristic
DR	Dispatching Rule
DFJSSP	Dynamic Flexible Job Shop Scheduling Problem
MT	Mean Tardiness
TEC	Total Energy Consumption

References

1. Ranade, A.; Gómez, J.; De Juan, A.; Chicaiza, W.D.; Ahern, M.; Escaño, J.M.; Hryshchenko, A.; Casey, O.; Cloonan, A.; O'Sullivan, D.; et al. Implementing Industry 4.0: An In-Depth Case Study Integrating Digitalisation and Modelling for Decision Support System Applications. *Energies* **2024**, *17*, 1818. [CrossRef]
2. Andrei, M.; Thollander, P.; Sannö, A. Knowledge demands for energy management in manufacturing industry-A systematic literature review. *Renew. Sustain. Energy Rev.* **2022**, *159*, 112168. [CrossRef]
3. Zhou, X.; Rao, W.; Liu, Y.; Sun, S. A Decentralized Optimization Algorithm for Multi-Agent Job Shop Scheduling with Private Information. *Mathematics* **2024**, *12*, 971. [CrossRef]
4. Jiang, B.; Ma, Y.; Chen, L.; Huang, B.; Huang, Y.; Guan, L. A Review on Intelligent Scheduling and Optimization for Flexible Job Shop. *Int. J. Control Autom. Syst.* **2023**, *21*, 3127–3150. [CrossRef]
5. Li, X.; Guo, X.; Tang, H.; Wu, R.; Wang, L.; Pang, S.; Liu, Z.; Xu, W.; Li, X. Survey of integrated flexible job shop scheduling problems. *Comput. Ind. Eng.* **2022**, *174*, 108786. [CrossRef]
6. Destouet, C.; Tlahig, H.; Bettayeb, B.; Mazari, B. Flexible job shop scheduling problem under Industry 5.0: A survey on human reintegration, environmental consideration and resilience improvement. *J. Manuf. Syst.* **2023**, *67*, 155–173. [CrossRef]
7. Zhang, F.; Mei, Y.; Nguyen, S.; Zhang, M. Survey on Genetic Programming and Machine Learning Techniques for Heuristic Design in Job Shop Scheduling. *IEEE Trans. Evol. Comput.* **2024**, *28*, 147–167. [CrossRef]
8. Lunardi, W.T.; Birgin, E.G.; Laborie, P.; Ronconi, D.P.; Voos, H. Mixed integer linear programming and constraint programming models for the online printing shop scheduling problem. *Comput. Oper. Res.* **2020**, *123*, 105020. [CrossRef]
9. Gmys, J.; Mezmaz, M.; Melab, N.; Tuyttens, D. A computationally efficient Branch-and-Bound algorithm for the permutation flow-shop scheduling problem. *Eur. J. Oper. Res.* **2020**, *284*, 814–833. [CrossRef]
10. Lin, S.W.; Ying, K.C. Minimising makespan in job-shops with deterministic machine availability constraints. *Int. J. Prod. Res.* **2021**, *59*, 4403–4415. [CrossRef]
11. Firme, B.; Figueiredo, J.; Sousa, J.M.; Vieira, S.M. Agent-based hybrid tabu-search heuristic for dynamic scheduling. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107146. [CrossRef]
12. Fontes, D.B.; Homayouni, S.M.; Gonçalves, J.F. A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources. *Eur. J. Oper. Res.* **2023**, *306*, 1140–1157. [CrossRef]

13. Sun, M.; Cai, Z.; Zhang, H. A teaching-learning-based optimization with feedback for LR fuzzy flexible assembly job shop scheduling problem with batch splitting. *Expert Syst. Appl.* **2023**, *224*, 120043. [CrossRef]
14. Cheng, L.; Tang, Q.; Zhang, L. Mathematical model and adaptive simulated annealing algorithm for mixed-model assembly job-shop scheduling with lot streaming. *J. Manuf. Syst.* **2023**, *70*, 484–500. [CrossRef]
15. Zhu, K.; Gong, G.; Peng, N.; Zhang, L.; Huang, D.; Luo, Q.; Li, X. Dynamic distributed flexible job-shop scheduling problem considering operation inspection. *Expert Syst. Appl.* **2023**, *224*, 119840. [CrossRef]
16. Zhang, B.; Pan, Q.K.; Meng, L.L.; Zhang, X.L.; Jiang, X.C. A decomposition-based multi-objective evolutionary algorithm for hybrid flowshop rescheduling problem with consistent sublots. *Int. J. Prod. Res.* **2023**, *61*, 1013–1038. [CrossRef]
17. Zhang, H.; Buchmeister, B.; Li, X.; Ojstersek, R. An Efficient Metaheuristic Algorithm for Job Shop Scheduling in a Dynamic Environment. *Mathematics* **2023**, *11*, 2336. [CrossRef]
18. Zhang, F.; Mei, Y.; Nguyen, S.; Zhang, M. Collaborative multifidelity-based surrogate models for genetic programming in dynamic flexible job shop scheduling. *IEEE Trans. Cybern.* **2021**, *52*, 8142–8156. [CrossRef]
19. Wang, H.; Peng, T.; Nassehi, A.; Tang, R. A data-driven simulation-optimization framework for generating priority dispatching rules in dynamic job shop scheduling with uncertainties. *J. Manuf. Syst.* **2023**, *70*, 288–308. [CrossRef]
20. Xiong, H.; Wang, H.; Shi, S.; Chen, K. Comparison study of dispatching rules and heuristics for online scheduling of single machine scheduling problem with predicted release time jobs. *Expert Syst. Appl.* **2024**, *243*, 122752. [CrossRef]
21. Panwalkar, S.S.; Iskander, W. A survey of scheduling rules. *Oper. Res.* **1977**, *25*, 45–61. [CrossRef]
22. Zhang, F.; Mei, Y.; Nguyen, S.; Zhang, M. Correlation coefficient-based recombinative guidance for genetic programming hyperheuristics in dynamic flexible job shop scheduling. *IEEE Trans. Evol. Comput.* **2021**, *25*, 552–566. [CrossRef]
23. Zhou, Y.; Yang, J.J.; Huang, Z. Automatic design of scheduling policies for dynamic flexible job shop scheduling via surrogate-assisted cooperative co-evolution genetic programming. *Int. J. Prod. Res.* **2020**, *58*, 2561–2580. [CrossRef]
24. Fan, H.; Xiong, H.; Goh, M. Genetic programming-based hyper-heuristic approach for solving dynamic job shop scheduling problem with extended technical precedence constraints. *Comput. Oper. Res.* **2021**, *134*, 105401. [CrossRef]
25. Rosca, J.P.; Ballard, D.H. *Genetic Programming with Adaptive Representations*; Department of Computer Science, University of Rochester: Rochester, NY, USA, 1994.
26. Mei, Y.; Nguyen, S.; Xue, B.; Zhang, M. An efficient feature selection algorithm for evolving job shop scheduling rules with genetic programming. *IEEE Trans. Emerg. Top. Comput. Intell.* **2017**, *1*, 339–353. [CrossRef]
27. Sitahong, A.; Yuan, Y.; Li, M.; Ma, J.; Ba, Z.; Lu, Y. Learning dispatching rules via novel genetic programming with feature selection in energy-aware dynamic job-shop scheduling. *Sci. Rep.* **2023**, *13*, 8558. [CrossRef] [PubMed]
28. Friedlander, A.; Neshatian, K.; Zhang, M. Meta-learning and feature ranking using genetic programming for classification: Variable terminal weighting. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 941–948.
29. Mei, Y.; Zhang, M.; Nyugen, S. Feature selection in evolving job shop dispatching rules with genetic programming. In Proceedings of the Genetic and Evolutionary Computation Conference, Denver, CO, USA, 20–24 July 2016; pp. 365–372.
30. Zhang, F.; Mei, Y.; Zhang, M. A two-stage genetic programming hyper-heuristic approach with feature selection for dynamic flexible job shop scheduling. In Proceedings of the Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 13–17 July 2019; pp. 347–355.
31. Zhang, F.; Mei, Y.; Nguyen, S.; Zhang, M. Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling. *IEEE Trans. Cybern.* **2020**, *51*, 1797–1811. [CrossRef] [PubMed]
32. Jia, S.; Yang, Y.; Li, S.; Wang, S.; Li, A.; Cai, W.; Liu, Y.; Hao, J.; Hu, L. The Green Flexible Job-Shop Scheduling Problem Considering Cost, Carbon Emissions, and Customer Satisfaction under Time-of-Use Electricity Pricing. *Sustainability* **2024**, *16*, 2443. [CrossRef]
33. Xu, B.; Mei, Y.; Wang, Y.; Ji, Z.; Zhang, M. Genetic programming with delayed routing for multiobjective dynamic flexible job shop scheduling. *Evol. Comput.* **2021**, *29*, 75–105. [CrossRef]
34. Shen, L.; Dauzère-Pérès, S.; Maecker, S. Energy cost efficient scheduling in flexible job-shop manufacturing systems. *Eur. J. Oper. Res.* **2023**, *310*, 992–1016. [CrossRef]
35. Branke, J.; Nguyen, S.; Pickardt, C.W.; Zhang, M. Automated design of production scheduling heuristics: A review. *IEEE Trans. Evol. Comput.* **2015**, *20*, 110–124. [CrossRef]
36. Tian, Y.; Gao, Z.; Zhang, L.; Chen, Y.; Wang, T. A Multi-Objective Optimization Method for Flexible Job Shop Scheduling Considering Cutting-Tool Degradation with Energy-Saving Measures. *Mathematics* **2023**, *11*, 324. [CrossRef]
37. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
38. Luan, F.; Cai, Z.; Wu, S.; Liu, S.Q.; He, Y. Optimizing the low-carbon flexible job shop scheduling problem with discrete whale optimization algorithm. *Mathematics* **2019**, *7*, 688. [CrossRef]
39. Zhang, F.; Mei, Y.; Zhang, M. Genetic programming with multi-tree representation for dynamic flexible job shop scheduling. In Proceedings of the AI 2018: Advances in Artificial Intelligence: 31st Australasian Joint Conference, Proceedings 31, Wellington, New Zealand, 11–14 December 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 472–484.
40. Tay, J.C.; Ho, N.B. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Comput. Ind. Eng.* **2008**, *54*, 453–473. [CrossRef]

41. Nguyen, S.; Zhang, M.; Johnston, M.; Tan, K.C. Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming. *IEEE Trans. Evol. Comput.* **2013**, *18*, 193–208. [CrossRef]
42. Zeiträg, Y.; Figueira, J.R.; Horta, N.; Neves, R. Surrogate-assisted automatic evolving of dispatching rules for multi-objective dynamic job shop scheduling using genetic programming. *Expert Syst. Appl.* **2022**, *209*, 118194. [CrossRef]
43. Ishibuchi, H.; Imada, R.; Setoguchi, Y.; Nojima, Y. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evol. Comput.* **2018**, *26*, 411–440. [CrossRef]
44. Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 196–202.
45. Shady, S.; Kaihara, T.; Fujii, N.; Kokuryo, D. A novel feature selection for evolving compact dispatching rules using genetic programming for dynamic job shop scheduling. *Int. J. Prod. Res.* **2022**, *60*, 4025–4048. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

An Improved YOLOv7-Based Model for Real-Time Meter Reading with PConv and Attention Mechanisms

Xiancheng Peng [1], Yangzhuo Chen [2], Xiaowen Cai [2,*] and Jun Liu [1]

[1] Artificial Intelligence, Xiangtan University, Xiangtan 411100, China; 202105570214@smail.xtu.edu.cn (X.P.); 202105570211@smail.xtu.edu.cn (J.L.)
[2] School of Automation and Electronic Information, Xiangtan University, Xiangtan 411100, China; chenyz5@xtu.edu.cn
* Correspondence: caixiaowen@xtu.edu.cn; Tel.: +86-181-0732-4331

Abstract: With the increasing complexity of the grid meter dial, precise feature extraction is becoming more and more difficult. Many automatic recognition solutions have been proposed for grid meter readings. However, traditional inspection methods cannot guarantee detection accuracy in complex environments. So, deep-learning methods are combined with grid meter recognition. Existing recognition systems that utilize segmentation models exhibit very high computation. It is challenging to ensure high real-time performance in edge computing devices. Therefore, an improved meter recognition model based on YOLOv7 is proposed in this paper. Partial convolution (PConv) is introduced into YOLOv7 to create a lighter network. Different PConv introduction locations on the base module have been used in order to find the optimal approach for reducing the parameters and floating point of operations (FLOPs). Meanwhile, the dynamic head (DyHead) module is utilized to enhance the attention mechanism for the YOLOv7 model. It can improve the detection accuracy of striped objects. As a result, this paper achieves mAP^{val}_{50} of 97.87% and $mAP^{val}_{50:90}$ of 62.4% with only 5.37 M parameters. The improved model's inference speed can reach 108 frames per second (FPS). It enables detection accuracy that can reach ±0.1 degrees in the grid meter.

Keywords: yolov7; meter-sensor reading; real-time detection; deep learning

1. Introduction

As grid industrial automation advances, intelligent inspections are repeatedly propelled into the spotlight. Automating the inspection of various old-fashioned instruments is challenging in grid meter detection. A single medium-sized substation may be equipped with as many as 800 to 1000 pointer-type meters. High-frequency manual inspections inevitably result in an increased error rate. Therefore, it is imperative to propose an effective and intelligent solution to alleviate this repetitive and burdensome work.

Currently, there has been much research in automatic grid meter recognition. In the past, traditional computer vision and digital image processing were commonly used for reading meter data. Zhang et al. [1] improved the visual saliency of pointer meter regions by establishing the localized pixel inhomogeneity factor (LPIF) model. And, Hough transform [2] was utilized to recognize the meter readings. Meng et al. [3] employed segmentation of pointers through thresholding and the area growth method. They used the Bresenham algorithm to calculate the deflection angle between the center line of the pointer and the zero scale. This method incorporates dynamic thresholding and pixel restoration mechanisms into the recognition process. This can cause it to be robust. However, in some environments with significant lighting changes, reflections of the meter dial face can affect the recognition of the pointer features.

Because of the proliferation of deep learning, numerous methods have emerged. He et al. [4] utilized the Mask R-CNN [5] for grid meter detection. Target detection was

employed to locate the dial, followed by semantic segmentation to delineate the pointer region. Principal Component Analysis (PCA) was used to fit the slope of the pointer, and accurate meter readings were achieved. Advanced deep-learning techniques were used in this method for accurate detection. However, the high computing overhead of the semantic segmentation model results in it being difficult to deploy on edge devices, resulting in a low real-time performance. Fan et al. [6] improved YOLOv5 based on the Global Context (GC) [7] module. The meter was located through target detection. The U-net [8] was employed for pointer segmentation and keypoint detection. They also used PCA to fit the slope of the pointer. This approach can adapt to a certain level of complexity in the environment. However, it requires two separate model inferences—the YOLOv5 target detection and U-net semantic segmentation. It results in significant computational overhead and longer processing time. Therefore, a grid pointer recognizer that combines high accuracy and robustness with real-time performance is still needed.

Regarding the problem mentioned above, an improved model based on YOLOv7 [9] is implemented that combines Partial Convolution (PConv) [10] and Dynamic Head (DyHead) [11] for grid meter detection. Additionally, an adaptive calibration algorithm has been incorporated for pointer reading calculation. This paper achieves relatively accurate readings in complex environments while maintaining a high real-time performance and lower parameters. Meanwhile, it demonstrates stable recognition when detecting pointers in dark, blurry, and small target scenarios.

Over 1000 photographs of grid pointer meters were utilized as datasets. These were collected at a substation, as shown in Figure 1. The datasets comprised images of meters under various working conditions, including shots taken at different angles and periods for the same meters. Data augmentation was also performed in varying degrees. These practices enriched the diversity of the dataset features, enabling the model to learn more generalized characteristics. It also helped reduce the risk of overfitting and improved overall model performance.



Figure 1. Overview of images in the dataset with different noise conditions.

The main contributions of our work are as follows:

(1) In grid meter detection, PConv was introduced into the base module of YOLOv7 and it obtained a relatively optimal replacement structure through multiple experiments, reducing the model's FLOPs and parameters.
(2) An adaptive calibration algorithm was proposed for pointer reading calculation, which exhibited a relatively high robustness and good detection accuracy.
(3) DyHead was integrated into YOLOv7, incorporating an attention mechanism that boosted the accuracy when detecting small target features.

The remainder of this paper is organized as follows: Section 2 provides the development of the critical technologies. Section 3 discusses the main technical improvements. The results of the experiments and ablation experiments are presented in Section 4. Finally, Sections 5 and 6 summarize the paper, offering insights into the approach's merits and outlining directions for future work.

2. Related Work

2.1. Real-Time Object Detection

Currently, there are two-stage detectors with a high detection accuracy, such as Fast R-CNN [12] and Faster R-CNN [13]. There are also high real-time one-stage object detectors, such as SSD [14] and the YOLO [15] series. Similarly, there are excellent semantic segmentation models like Mask R-CNN [5], which integrates the achievements of deep learning in computer vision.

The real-time performance becomes increasingly important as deep-learning vision models are implemented into industrial production. Typically, techniques such as pruning, quantization, and knowledge distillation [16] can be employed to reduce the model's complexity. This can improve the inference speed and enhance the real-time performance. Additionally, improved convolution modules can also be applied to reduce the model's parameters and FLOPs, such as depthwise convolution (DWConv) [17] and PConv. These modules can reduce the parameters, accelerate the model's inference performance, and enhance the real-time capabilities to some extent.

2.2. Pointer Meter Reading

Past researchers have achieved substantial work in the grid meter reading field. Some researchers have employed traditional visual solutions in their work. They have utilized feature extraction methods to detect the pointer meter. Others have adopted deep learning, detecting meters through object detection and using semantic segmentation to delineate the regions of the pointers. Our team has accumulated a wealth of experience in meter reading. Past works have attempted to use image classification by detecting meters through sliding windows. Feature point detection has also been used to calibrate the dial to achieve meter readings. However, these methods could achieve a better real-time performance.

In this paper, PConv is introduced into the base module of the YOLOv7 backbone, replacing regular convolutions to reduce the computational load and increase the frames per second (FPS). Eisting work such as Sun et al. [18] has introduced PConv into the YOLO series networks. However, this work does not provide details on how the integration is performed, the number of replacements, and crucial aspects like the replacement positions. Therefore, our work experiments with various replacement approaches in the base module of the YOLOv7 backbone and identifies the most effective replacement through experimentation. Moreover, DyHead with an attention mechanism is added to the detection head of YOLOv7 to enhance the model's performance in strip object recognition.

3. Methods

The algorithmic process is illustrated in Figure 2. The meter image is initially captured and fed into the improved YOLOV7$_{tiny}$ object detector to detect relevant elements in the meter dial. YOLOv7 is used to extract the positions of the short pointers and the regions

corresponding to the left and right starting points of the instrument's scale. The area of the detected pointer is cropped from object detection.

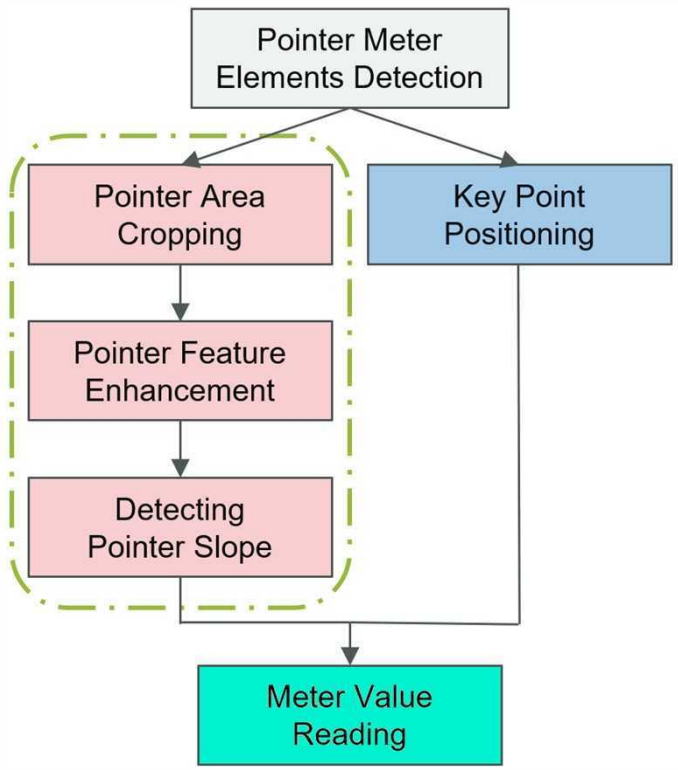


Figure 2. The entire algorithmic process.

The detected pointer region and key point coordinates are input into the adaptive calibration program for pointer reading. As the linear features of the pointer are very distinct in the cropped image, the adaptive calibration algorithm is used to estimate the slope of the pointer relative to the dial center coordinate system.

The meter reading can be calculated through a proportional relationship using the pointer's slope (k), dial center coordinate (x_c, y_c), and dual boundary location (x_{b1}, y_{b1}), (x_{b2}, y_{b2}).

The direction vector of the pointer is

$$\vec{P} = (1, k) \tag{1}$$

The vector of the dual boundary location to the center is

$$\begin{aligned} \vec{V}_{left} &= (x_{b1} - x_c, y_{b1} - y_c) \\ \vec{V}_{right} &= (x_{b2} - x_c, y_{b2} - y_c) \end{aligned} \tag{2}$$

So, the angle from the pointer to the nearest boundary of the dial range can be calculated as

$$Angle = \min(\arccos(\frac{\vec{P} \cdot \vec{V}_{left}}{|\vec{P}| \cdot |\vec{V}_{left}|}),\ \arccos(\frac{\vec{P} \cdot \vec{V}_{right}}{|\vec{P}| \cdot |\vec{V}_{right}|})) \tag{3}$$

Using this angle, the meter's reading can be calculated through the proportional relationship.

The entire reading process is typically faster compared with using semantic segmentation, and it also exhibits a strong adaptability to different environments.

3.1. Data Augmentation

The pointer dataset for this paper was manually collected by team members at the substation, including meter images from various shooting angles, different periods, diverse lighting conditions, and varying pointer angles. Approximately 1000 images were annotated. The meter images were captured at distances ranging from 0.5 m to 2 m, showcasing a multi-scale characteristics.

Different levels of data augmentation techniques were incorporated into the meter datasets. Salt-and-pepper noise and random region blur were added to the meter images. Also, stains and old meters were simulated to enhance the network's ability to generalize such patterns. Random region blur is a technique used to increase the variety of a dataset. It calculates a value by multiplying the shorter side of the image and a specific ratio. This value is used as the length and width of a square region. The square is randomly applied to a particular area of the image, and a large-kernel mean filter is employed within this region, resulting in an image with region blur. The number of squares is generally limited to three or fewer per image, and the ratio is typically selected from a range of 0.2 to 0.3. Experiments have shown this data augmentation technique has a specific positive impact on the detection of striped objects. Different angles and various aspect ratios of pointer meters were stimulated by applying perspective transformation and random rotation. Additionally, the diversity of the datasets was enhanced by randomly adjusting the images' color space, brightness, and saturation within predefined ranges.

3.2. ELAN-PConv

The original YOLOv7 base module was structured according to the ELAN module, as depicted in Figure 3. The entire backbone underwent hierarchical feature extraction through four layers of ELAN modules. The number of image channels progressed from 128 to 256, then to 512, and finally to 1024. 1×1 and 3×3 standard convolutions were utilized in each ELAN module. So, in each computation step, a large number of convolutional kernel operations were required. Taking the example of the ELAN module with an output channel of 256, a single computation between the two 3×3 convolutional layers utilized 4096 convolutional kernels. This undoubtedly imposed a significant computational burden, particularly for the edge devices. This resulted in a substantial computation overhead.

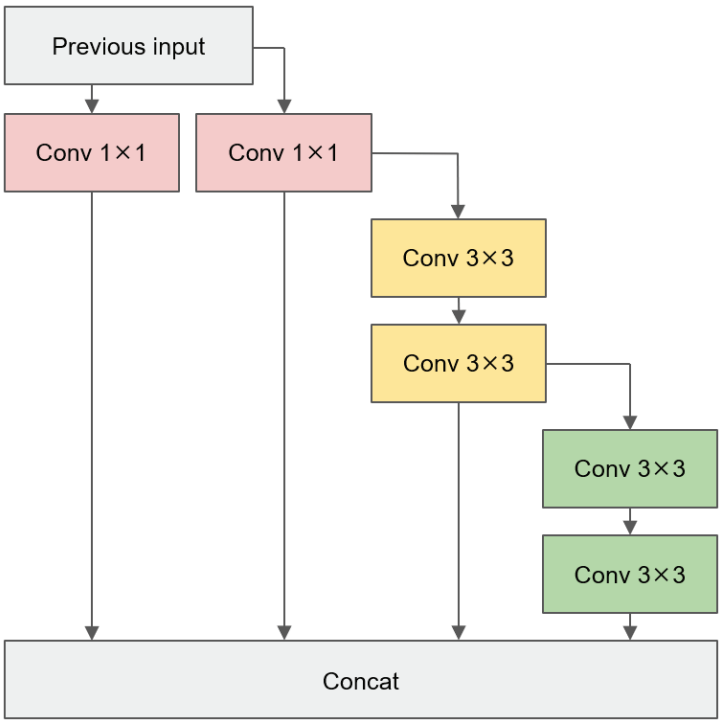


Figure 3. ELAN module structure in YOLOv7.

Inspired by the PConv proposed in FasterNet [10], Partial Convolution (PConv) represented a recent improvement in standard convolution. Its primary design goal was to reduce memory access and computational redundancy. The use of PConv was motivated by the observation that feature maps exhibited a high similarity across different channels, indicating a significant redundancy between channels. Therefore, leveraging the redundancy of the feature maps, PConv selectively applied standard convolution on a subset of channels while keeping the remaining channels unchanged, as shown in Figure 4. This reduced the computational complexity and contributed to realizing a fast and efficient neural network.

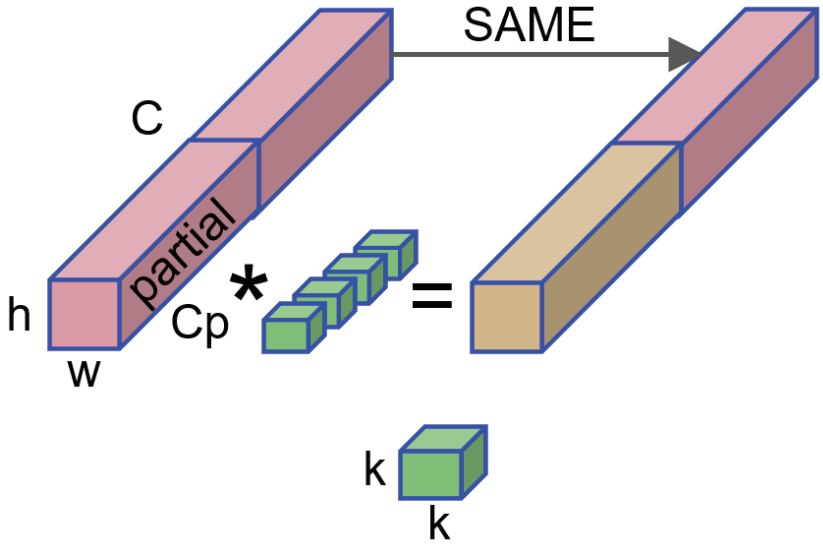


Figure 4. PConv computation diagram.

PConv was introduced at different positions in this paper to replace the standard convolutions in the ELAN modules. Validations were conducted to determine the best approach for introducing PConv in the Experiments Section.

PConv typically selects a certain proportion of channels for standard convolution, usually with $r = \frac{1}{4}$. When the channel numbers of input and output are the same, the FLOPs for using PConv are

$$h \times w \times k^2 \times c_p^2. \tag{4}$$

When $r = \frac{1}{4}$, PConv is simply $\frac{1}{16}$ of the standard convolution, it requires less memory accesses. The memory accesses for using PConv are

$$h \times w \times 2c_p + k^2 \times c_p^2 \approx h \times w \times 2c_p, \tag{5}$$

where h and w represent the feature map's width and height, respectively, and k represents the size of the convolution kernel. c_p refers to a subset of the feature channels.

When $r = \frac{1}{4}$, memory access is only $\frac{1}{4}$ of the standard convolution.

To integrate the feature information from different channels more effectively, PConv is typically followed by a Point-Wise Convolution (PWConv) [10] connection.

3.3. Change to DyHead

The IDetect module [9] of YOLOV7 was improved in this paper. IDetect is the prediction head of the YOLOV7 network, and is responsible for classification and regression. In the experiments of pointer detection in YOLOv7, the pointer scale and aspect ratio in each meter varied greatly. The performance was suboptimal for some relatively short and thin pointers. The features learned by the model were scattered, leading to a certain degree of deviation in the regression prediction boxes. So, DyHead was added before IDetect to enhance its perception of the scale and space, as shown in Figure 5.

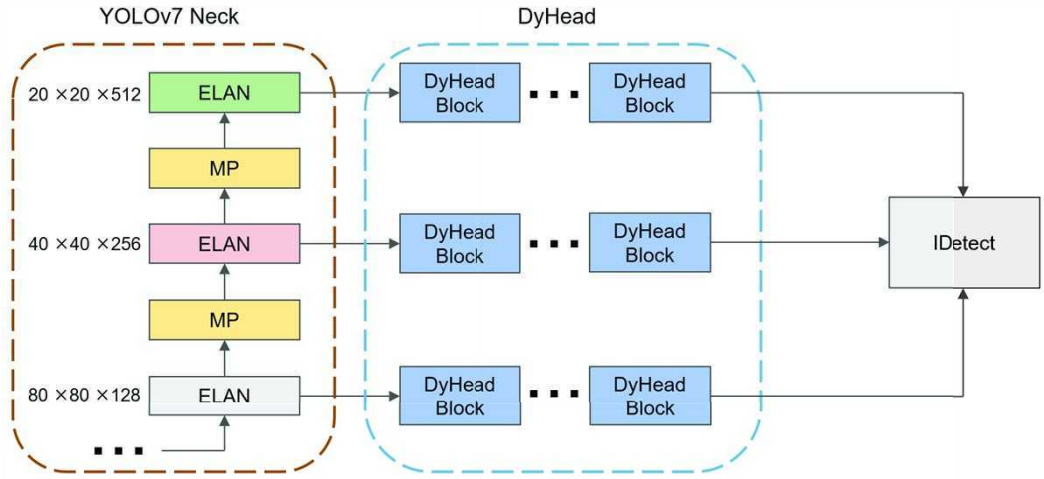


Figure 5. Dyhead added position in YOLOv7.

Dynamic Head (DyHead) is a dynamic target detection head, as shown in Figure 6. It unified scale awareness, spatial awareness, and task-aware attention, constructing a self-attention mechanism. After feature extraction by the backbone network of YOLOV7, the obtained feature pyramids at three different scales were fed into DyHead. This involved utilizing a self-attention mechanism for spatial, scale, and task perception. The outputs were provided to IDetect for the classification and regression. This approach benefited the network by paying more attention to small objects or objects with significant aspect ratio differences. It enhanced the detection the performance of pointers.

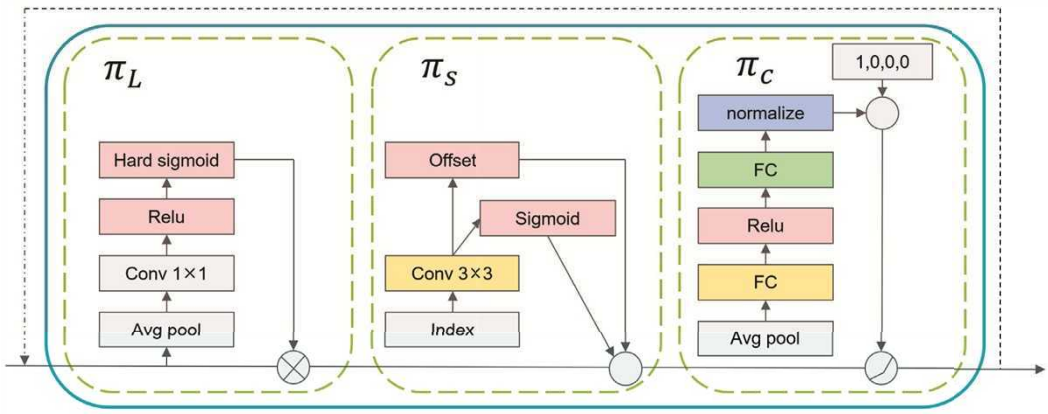


Figure 6. The internal structure of a single DyHead block.

The heatmap displayed in Figure 7 shows that DyHead's multi-layer stacking enabled the feature layers to gradually converge onto the pointers without interference from the other dial components. Therefore, incorporating an attention mechanism into the model could enhance its ability to detect pointers with significant variations in spatial scales and aspect ratios, thereby improving its detection capability in complex environments.

3.4. Adaptive Calibration Pointer

After object detection, the region and key points of the pointer were input into the adaptive calibration program for meter reading. The detected key points mainly included the start and end positions on both sides of the meter range. This paper achieved this by cropping and duplicating a portion of the pointer region and applying Gaussian smoothing for noise reduction. The maximum interclass variance method (OTSU's method) [19] was used to segment the pixels of the pointer. This is an adaptive threshold segmentation method. As short and thin pointers are typically in small regions on the dial face, the complexity of the dial face does not significantly impact the accurate segmentation of the

pointer from the background. Moreover, OTSU's method performs segmentation based on interclass variance and it is not influenced by image brightness and contrast. When obtaining the segmented image, we employed Canny edge detection [20], resulting in two distinct edges on the pointer. Hough line transform was applied to obtain the equations of these two lines. The complete process of extracting the pointer is schematized in Figure 8.

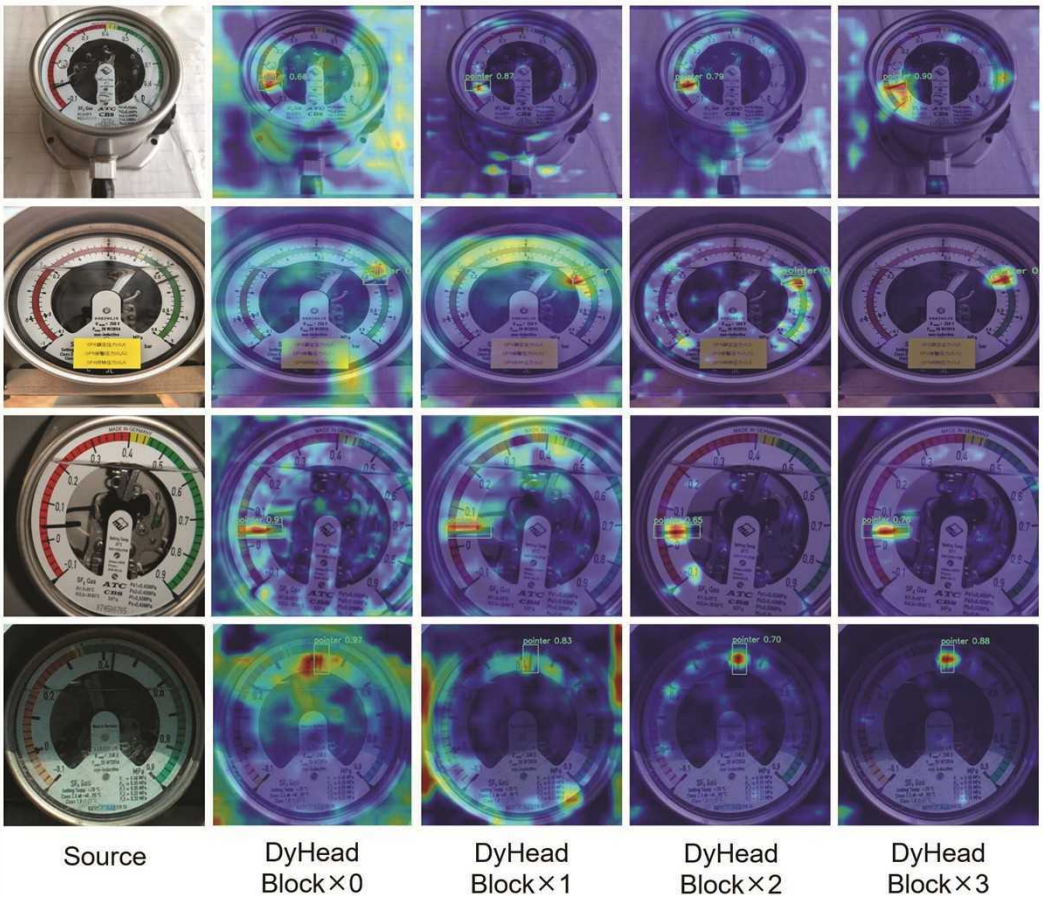


Figure 7. Stacking different numbers of DyHead blocks results in the heatmap.

If the two lines are parallel, the program can arbitrarily select one as the fitted slope for the pointer. In the case of a sharp-tipped pointer, the average of the slopes of these two lines can be calculated and used as the fitted slope. The meter reading can be determined through proportional relationships by using the key points on both sides and the slope of the pointer.

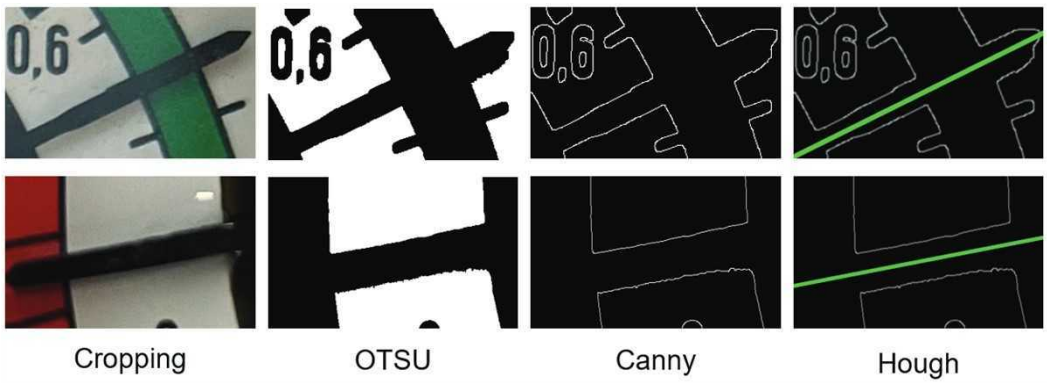


Figure 8. Process of extracting pointer features and fitting the slope.

4. Experiments

4.1. Experimental Setup

Experiments were conducted on Ubuntu 20.04. The software stack was configured with Python 3.8 and PyTorch 1.13.1. For hardware, training was performed on a Tesla P100 GPU. In the inference, the RTX 3060 Laptop was used to simulate the computational capabilities of the edge devices. The training code was based on the YOLOv7 project, which WongKinYiu opens sources on GitHub.

Regarding the model training parameter settings, Stochastic Gradient Descent (SGD) is the optimizer with an initial learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. The training was conducted on a batch size of 64 for 500 epochs of the dataset. Additionally, pre-trained weights were not utilized for a fair performance comparison between the different networks.

Mean Average Precision (mAP) was utilized as the accuracy evaluation metric. It is computed as the average precision (AP) average across all categories. Specifically, mAP^{val}_{50} calculates the average precision when the Intersection over Union (IoU) threshold is 50%.

$$\mathrm{mAP} = \frac{1}{C}\sum_{c=1}^{C} \mathrm{AP}_c \tag{6}$$

where C represents the number of categories and c represents the specific category.

Using this evaluation metric helps better reflect the model's accuracy in recognizing features.

4.2. Ablation Study

4.2.1. PConv Replacement Position in ELAN

Because of the characteristics of PConv, only a subset of channels underwent convolution, while the rest remained unchanged. Therefore, the position of replacing standard convolutions with PConv in the ELAN module was bound to affect the network's performance. Figure 9 shows four replaceable convolution positions for the ELAN structure. Five replacement structures are shown in Table 1.

YOLOv7$_{tiny}$ + PConv-x refers to a different replacement position for PConv in the YOLOv7 ELAN module. Comparative experiments were conducted on different positions and quantities of PConv introduction and identified the most effective positional structure. The PConv module was configured to default to connecting a PWConv for feature fusion and channel transformation. All of the YOLOv7$_{tiny}$ + PConv-x experiments were conducted directly on the original YOLOv7$_{tiny}$ model.

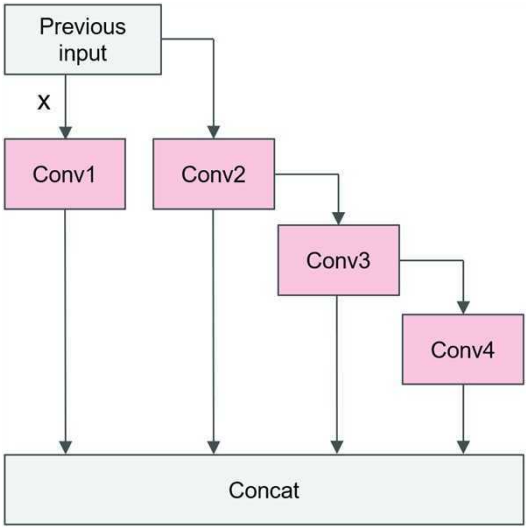


Figure 9. Location of the four Conv (standard convolutions) in the ELAN module of YOLOv7$_{tiny}$.

Table 1. PConv replacement position ("✓" means this Conv in Figure 9 will be replaced).

Model	Conv1	Conv2	Conv3	Conv4
YOLOv7$_{tiny}$ + PConv-1	✓	✓	✓	✓
YOLOv7$_{tiny}$ + PConv-2	✓	✓		
YOLOv7$_{tiny}$ + PConv-3			✓	✓
YOLOv7$_{tiny}$ + PConv-4		✓		✓
YOLOv7$_{tiny}$ + PConv-5		✓	✓	✓

Through experiments, YOLOv7$_{tiny}$ + PConv-1 was found to have fewer parameters and FLOPs, as shown in Table 2. This is because all the Conv1-4 were replaced standard convolutions with PConv. This implies that only a subset of channels underwent convolution, resulting in fewer convolutional kernels. However, as the 3×3 convolution in ELAN was the primary feature extraction layer, replacing it may have led to insufficient feature extraction in some cases.

Table 2. Comparison of model complexity based on different PConv replacements.

Model	Params$_{(Net)}$	Params$_{(Block)}$	FLOPs
YOLOv7$_{tiny}$	6.02 M	1.31 M	13.2 G
YOLOv7$_{tiny}$ + PConv-1	**4.73 M**	0.41 M	10.5 G
YOLOv7$_{tiny}$ + PConv-2	6.12 M	1.38 M	13.6 G
YOLOv7$_{tiny}$ + PConv-3	4.83 M	0.33 M	10.1 G
YOLOv7$_{tiny}$ + PConv-4	5.35 M	0.86 M	11.9 G
YOLOv7$_{tiny}$ + PConv-5	4.78 M	0.37 M	10.3 G

From Table 3, it is observed that YOLOv7$_{tiny}$ + PConv-1 had the lowest mAP$^{val}_{50}$, demonstrating insufficient feature extraction due to the reduction in the number of convolutional kernels. YOLOv7$_{tiny}$ + PConv-4 exhibited the greatesr down inference time with a batch size 32, and it achieved a good mAP$^{val}_{50:95}$. It was mainly replaced with PConv in the Conv2 and Conv4 positions.

Table 3. Performance comparison based on different PConv replacements.

Model	Infer$_{(b32)}$	FPS	mAP$^{val}_{50}$	mAP$^{val}_{50:95}$
YOLOv7$_{tiny}$	7.6 ms	104	97.13%	60.33%
YOLOv7$_{tiny}$ + PConv-1	6.2 ms	109	97.09%	59.61%
YOLOv7$_{tiny}$ + PConv-2	5.9 ms	114	98.12%	59.23%
YOLOv7$_{tiny}$ + PConv-3	7.1 ms	104	98.27%	59.05%
YOLOv7$_{tiny}$ + PConv-4	6.5 ms	108	97.75%	**60.48%**
YOLOv7$_{tiny}$ + PConv-5	6.1 ms	110	97.55%	59.34%

4.2.2. Module Ablation Study

Ablation experiments were conducted on the selected improved modules, as shown in Table 4. YOLOv7$_{tiny}$-based PConv-1 had the most minor net parameters, and YOLOv7$_{tiny}$-based PConv-4 had the relatively best mAP$^{val}_{50:95}$. So, experiments were conducted by adding Dyhead to the YOLOv7$_{tiny}$ + PConv-1 and YOLOv7$_{tiny}$ + PConv-4 to determine which structure was more suitable.

From the experimental results, it can be observed that YOLOv7$_{tiny}$ + PConv-4 + DyHead demonstrated a better performance than the other improved models. Despite the increase in parameters with the addition of DyHead, there was a reduction in FLOPs. The mAP also reached a relatively superior level.

Table 4. Ablation experiments of the network improvement modules.

Model	Params	FLOPs	FPS	mAP^{val}_{50}	$mAP^{val}_{50:95}$
$YOLOv7_{tiny}$	6.02 M	13.2 G	104	97.13%	60.33%
$YOLOv7_{tiny}$ + Dyhead	5.96 M	13.0 G	106	97.83%	62.43%
$YOLOv7_{tiny}$ + Pconv-1	4.73 M	10.5 G	109	97.09%	59.61%
$YOLOv7_{tiny}$ + Pconv-4	5.35 M	11.9 G	108	97.75%	60.48%
$YOLOv7_{tiny}$ + Pconv-1 + Dyhead	4.77 M	10.5 G	110	97.80%	60.96%
$YOLOv7_{tiny}$ + Pconv-4 + Dyhead	5.37 M	11.8 G	108	**97.87%**	**62.67%**

4.2.3. Meter Reading Comparison

The meter reading errors using different YOLO models were compared. The pointer and keypoint features extracted by each YOLO model were fed into the reading program to calculate the meter value. The error was calculated by comparing the predicted meter values with the ground truth. This experiment processed all images in the test set to obtain each model's average error and inference speed, as shown in Figure 10.

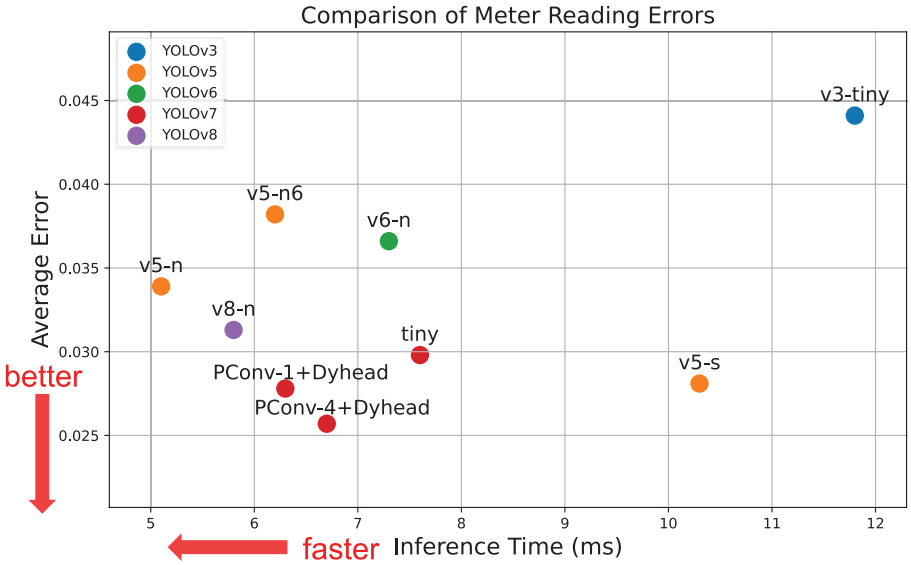


Figure 10. Comparison of meter reading with different models for feature extraction.

The models that were closer to the bottom-left corner were considered better. Compared with the YOLOv8n model, the original YOLOv7-tiny model had a higher accuracy, while YOLOv8n was faster. However, through field testing, our team concluded that accuracy was of a higher priority at the same parameter level. Therefore, we chose YOLOv7 as the base model.

5. Discussion

$YOLOv7_{tiny}$ + PConv-4 + DyHead was validated as being more suitable for meter datasets, providing a relatively optimal improvement structure through multiple experiments.

The improved network achieved a relatively optimal performance in mAP values compared with the other networks, as shown in Table 5. The mAP^{val}_{50} reached 97.87% and $mAP^{val}_{50:95}$ reached 62.67%. Meanwhile, the parameters and FLOPs were maintained at a low level. This demonstrates the effectiveness of reducing the network volume through PConv and enhancing the feature perception with DyHead for meter detection.

Table 5. Comparison of real-time model performances.

Model	Params	FLOPs	mAP^{val}_{50}	$mAP^{val}_{50:95}$
$YOLOv7_{tiny}$	6.02 M	13.2 G	97.13%	60.33%
$YOLOv7_{tiny}$ + Pconv-1 + Dyhead	4.77 M	10.5 G	97.80%	60.96%
$YOLOv7_{tiny}$ + Pconv-4 + Dyhead	5.37 M	11.8 G	**97.87%**	**62.67%**
$YOLOv3_{tiny}$	12.1 M	19.0 G	95.38%	55.78%
YOLOv5n	1.7 M	4.2 G	96.62%	59.32%
YOLOv5n6	3.1 M	4.3 G	96.72%	58.94%
YOLOv5s	7.2 M	17.0 G	97.18%	59.35%
YOLOv6n	4.2 M	11.9 G	96.77%	59.51%
YOLOv8n	3.0 M	8.2 G	96.51%	59.73%

6. Conclusions

Nowadays, numerous methods in grid meter reading are being used. However, these methods could be better in terms of both real-time performance and accuracy. This paper enhanced the speed and accuracy of grid meter detection in the improved YOLOv7 model. The ELAN module replaced the Standard convolutions with PConv to reduce the network's parameters and FLOPs, enhancing the real-time inference on the edge devices. In addition, an attention mechanism was introduced by incorporating the DyHead Block into the neck output of YOLOv7 IDetect. It improved the network's recognition of pointers and enhanced the meter detection accuracy. The mAP^{val}_{50} reached 97.87% and the $mAP^{val}_{50:95}$ reached 62.67% in the improved model. The improved model took 6.4 ms to infer a single image, achieving a frames-per-second (FPS) rate of 108. It exhibited an excellent performance in terms of both accuracy and real-time capabilities.

In future work, our team will further explore the challenges in grid meter detection. Lighter-weight networks will be investigated to expand the applicability of our work. We will explore how to achieve simple visual detection of meters on microcontrollers such as STM32H7 or MCX N Series. Our team will continue to advance the construction of industrial intelligence in the future, striving towards comprehensive intelligent manufacturing and intelligent inspection (Figure 11).

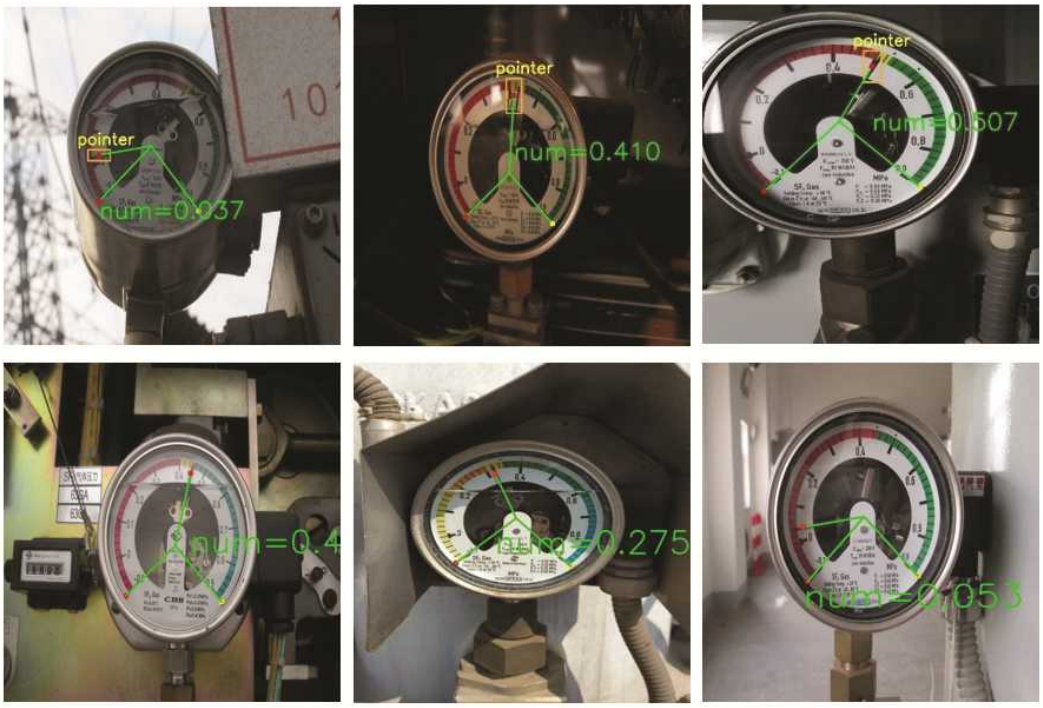


Figure 11. Actual industrial application examples of meter reading in the grid industry.

Author Contributions: Conceptualization, X.P.; methodology, X.P.; software, X.P.; validation, X.P. and J.L.; investigation, X.P., Y.C. and J.L.; resources, Y.C.; data curation, X.P.; writing—original draft preparation, X.P.; writing—review and editing, X.C.; visualization, X.P.; supervision, X.C.; project administration, X.C. and Y.C.; funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Research and Development Program of Hunan Province of China (Grant No. 2023GK2032).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The main code can be accessed from GitHub: https://github.com/Flora233333/Pointer-Reading-based-on-YOLO (accessed on 20 May 2024).

Acknowledgments: This paper is supported by the "Beidou High-Precision Navigation and Positioning Technology Innovation Center" of Hunan Shaofeng Applied Mathematics Research Institute, and the School-Enterprise Cooperation Innovation and Entrepreneurship Education Base for Intelligent Railway Equipment Design in Hunan Province.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Zhang, L.; Fang, B.; Zhao, X.; Zhang, H.S. Pointer-type meter automatic reading from complex environment based on visual saliency. In Proceedings of the 2016 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), Jeju Island, Republic of Korea, 10–13 July 2016; pp. 264–269. [CrossRef]
2. Illingworth, J.; Kittler, J. A survey of the Hough transform. *Comput. Vis. Graph. Image Process.* **1988**, *44*, 87–116. [CrossRef]
3. Meng, L.; Cheng, J. Research on the Visual Recognition Method of Pointer Water Meter Reading. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021; Volume 5, pp. 909–914. [CrossRef]
4. He, P.; Zuo, L.; Zhang, C.; Zhang, Z. A Value Recognition Algorithm for Pointer Meter Based on Improved Mask-RCNN. In Proceedings of the 2019 9th International Conference on Information Science and Technology (ICIST), Hulunbuir, China, 2–5 August 2019; pp. 108–113. [CrossRef]
5. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
6. Fan, Z.; Shi, L.; Xi, C.; Wang, H.; Wang, S.; Wu, G. Real Time Power Equipment Meter Recognition Based on Deep Learning. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–15. [CrossRef]
7. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
8. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III 18; Springer: Berlin, Germany, 2015; pp. 234–241.
9. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
10. Chen, J.; Kao, S.h.; He, H.; Zhuo, W.; Wen, S.; Lee, C.H.; Chan, S.H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
11. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L. Dynamic Head: Unifying Object Detection Heads With Attentions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7373–7382.
12. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2015; Volume 28.
14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin, Germany, 2016; pp. 21–37.

15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
16. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
17. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861
18. Sun, Y.; Li, Y.; Li, S.; Duan, Z.; Ning, H.; Zhang, Y. PBA-YOLOv7: An Object Detection Method Based on an Improved YOLOv7 Network. *Appl. Sci.* **2023**, *13*, 10436. [CrossRef]
19. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
20. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Advanced Monitoring of Manufacturing Process through Video Analytics

Nisar Hakam [1,2,*], Khaled Benfriha [1], Vincent Meyrueis [1] and Cyril Liotard [2]

[1] Arts et Métiers, Institute of Technology (AMIT), 75013 Paris, France; khaled.benfriha@ensam.eu (K.B.); vincent.meyrueis@ensam.eu (V.M.)
[2] ERM Automatismes, 84200 Carpentras, France; c.liotard@erm-automatismes.com
* Correspondence: nisar.hakam@ensam.eu

Abstract: The digitization of production systems has revolutionized industrial monitoring. Analyzing real-time bottom-up data enables the dynamic monitoring of industrial processes. Data are collected in various types, like video frames and time signals. This article focuses on leveraging images from a vision system to monitor the manufacturing process on a computer numerical control (CNC) lathe machine. We propose a method for designing and integrating these video modules on the edge of a production line. This approach detects the presence of raw parts, measures process parameters, assesses tool status, and checks roughness in real time using image processing techniques. The efficiency is evaluated by checking the deployment, the accuracy, the responsiveness, and the limitations. Finally, a perspective is offered to use the metadata off the edge in a more complex artificial-intelligence (AI) method for predictive maintenance.

Keywords: video analytics; digitization; data deviation; detection; machine; visual; industry 4.0

1. Introduction

The framework of Industry 4.0 focuses on deploying three computing approaches: edge, fog, and cloud. These structures enable the integration of image analysis studies on the shop floor, as seen in Figure 1. Edge computing includes all modules, servers, or desktops situated on the operation technology (OT) network. Fog computing allows the usage of the information technology (IT) network, offering better performance due to high-performance equipment. Cloud computing is considered an open configuration platform, providing fast response and preconfigured processing blocks.

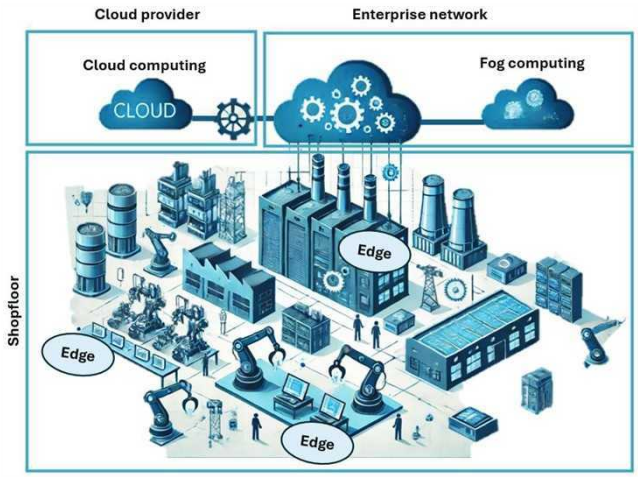


Figure 1. Computing architectures inside the framework of Industry 4.0.

1.1. Industrial Data

Data mining is fundamental in modern industrial sectors, aiming to optimize and adjust operations, as explained by Tambare et al. (2022) [1]. However, essential challenges such as system integration, communication, and collaboration must be tackled. The goal is to maintain a continuous production cycle, preventing unexpected machine breakdowns. This objective aligns with smart supervision and edge computing, where data are contextualized into performance indicators, expanding system monitoring.

Several studies have explored the potential of operational images to extract metadata from isolated systems. Various image-based methods allow the measurement of operational and environmental data from system operations. However, image processing requires parameter tuning and stable lighting, as presented by Mirbod et al. (2022) [2]. Simultaneously, AI approaches facilitate metadata prediction if a sufficient, unbiased dataset is available, as illustrated by Caggiano et al. (2019) [3].

1.2. Video Analytics in Edge Computing

Considering the objective of Industry 4.0, Bazhenov et Korzun (2019) discusses the application of event-driven video services for monitoring in edge-centric Internet of Things (IoT) environments [4]. It emphasizes that video analytics in IoT environments should not rely on transferring raw video data to a central processing center (like cloud-based systems). Instead, it proposes a model where multiple edge video capture devices process data streams locally for event detection. These events can be simple occurrences within a single video stream or complex events derived by correlating events across multiple streams and incorporating contextual information from the IoT environment. The authors highlight the importance of microservice architecture, event-driven models, and context-based mechanisms in constructing these video services. They also present a demo implementation of a video service for personnel recognition around production equipment, showcasing its use in digital monitoring for smart manufacturing processes.

Simultaneously, Kim et al. (2019) present a novel monitoring approach for small and medium-sized enterprises (SMEs) called KEM (keep an eye on your machine) [5]. The system uses a low-cost vision device like a webcam and open-source technologies to collect and process operational data from computer numerical control (CNC) machine tools. A prototype was tested with a 3-axis CNC milling machine, and the results showed that the system was easy to integrate and could be applied to legacy machine tools without significant changes or cost barriers. The authors believe that this approach can change monitoring operations, reduce operational time and energy consumption, and lessen the environmental impact of sustainable production in SMEs.

These two applications present initial roadmaps for video analytics. Bazhenov et Korzun (2019) [4] suggest that their event-driven model, which is a key element of their overall concept, can be extended to other applications beyond manufacturing, such as object recognition and recognition of interaction between people and physical objects. On the other hand, Kim et al. (2019) [5] consider the application of their approach with other manufacturing equipment, such as turning centers, parts feeders, and 3D printers.

1.3. Scope

This paper aims to reinforce the models of Bazhenov et al. (2019) and Kim et al. (2019) [4,5]. This reinforcement consists of optimizing the processing power for the edge module by preventing data latency, machine reconfiguration, and security risks. Section 2 highlights some video analytics in manufacturing, considering the limitations, the structural implementation, and the accuracy of the algorithm. The approach is tested on a CNC lathe machine by contextualizing video frames into operational data.

2. Image Processing and AI Vision

Image processing is based on filters and pixel calculations, while AI vision is a modern analysis technique that learns pixel deviations and correlates unseen images with pre-

trained ones. Both techniques are considered artificial intelligence (AI) methods. Presently, they are used in manufacturing processes to contextualize video data, such as operative parameters and process faults.

An investigation of 11 research studies shows the importance of video analytics in manufacturing processes according to five criteria, as seen in Table 1.

Table 1. A comparative review of image-based studies.

Study	Process	Algorithm	Network	Accuracy	Gap
Mazumder (2015) [6]	AM *	SVM * PCA *	Edge/Fog	<80%	Nonsufficient
Abid Hasan et Ko (2016) [7]	Detection	Canny	Edge	>95%	Applicability in manufacturing
Nuchitprasitchai et al. (2017) [8]	AM *	SIFT * RANSAC *	Edge	>90%	Code modification
Ceruti et al. (2017) [9]	AM *	SURF *	Edge	>90%	Refine error
Delli et Chang (2018) [10]	AM *	SVM *	Edge/Fog	90%	Add checkpoints
Caggiano et al. (2019) [3]	AM *	DCNN *	Fog/Cloud	99.4%	Single process
Kalbasi et Nikmehr (2020) [11]	Detection	Canny	Edge	--	Applicability in manufacturing
Giusti et al. (2020) [12]	SM *	CNN *	Fog/Cloud	98%	Apply in situ automatic control
Cai et al. (2021) [13]	Detection	PNGAN *	Fog/Cloud	>95%	Applicability in manufacturing
Wang et al. (2022) [14]	AM *	CNN * DCNN *	Fog/Cloud	98%	Include more capabilities
Halawi Ghoson et al. (2023) [15]	SM *	Mask	Edge	99%	Nonadopted for online monitor

* AM—additive manufacturing; SM—subtractive manufacturing; SVM—support vector machine; PCA—principal component analysis; SIFT—scale invariant feature transform; RANSAC—random sample consensus; SURF—speed up robust features; CNN—convolutive neural network; PNGAN—pixel-level noise-aware generative adversarial network; DCNN—deconvolutional neural network.

2.1. Additive and Subtractive Manufacturing

The study by Mazumder (2015) proposed a design for a metal additive manufacturing machine [6]. They incorporated a video sensor for constant defect monitoring. While suitable for zero-point design, the approach did not discuss advanced image analysis, achieving less than 80% efficiency due to limitations in integrating reliable measurements using edge computing.

Feature point localization is also an image processing technique. Nuchitprasitchai et al. (2017) used the RANSAC and SIFT algorithms to identify key points between a 3D-printed part and its virtual representation [8]. This method controlled printing accuracy and ensured continuous operation without faults, but it slowed down printing by a factor of four.

Augmented reality (AR) is another technology for assessing printing faults. Ceruti et al. (2017) implemented an innovative approach using AR technology [9]. They projected a simulation from the CAD file onto the printing bed, detecting 1-mm deviations and running filament defects between the projected and real product. However, integration failed when not enough feature points were recognized between real and simulated products using the SURF algorithm.

In addition to classical recognition, AI algorithms predict data. Delli et Chang (2018) trained an SVM, a machine learning method trained to detect running filaments and fault prints during the process [10]. They installed a camera with an adapted graphic card, achieving 90% accuracy and less than 1 s response time, but reproducing the work required, including print pauses and high-performance servers.

Caggiano et al. (2019) developed an approach to transform images from a selective laser-melting (SLM) printer into operational data [3]. Points, such as the applicability of

other machines and overfitting probability, were not discussed. Simultaneously, training and deployment required computational resources, making it only applicable to an integrated fog or cloud network.

Along with additive manufacturing, Wang et al. (2022) used existing methodologies to implement a robot welding process [14]. They utilized deep learning methods, like CNN and DCNN, to measure operational parameters: welding speed, current, and torch angle. Surface measurement for the welded area was necessary to calculate process parameters using the DCNN, requiring fog or cloud computing for optimized training and deployment.

AM presents only one type of creation of new products from 3D CAD files. In industry, these manufacturing machines are not always deployed due to uncertainty in final production. Thus, a discussion about SM is fundamental to understanding video analytics' applicability in the field.

Video analysis on SM machines focuses on a detection model for the machining context, such as parameters, clipping angle, etc. A deeper investigation is crucial to understand the requirements for integration with real production data. Currently, AI vision approaches are not often deployed due to biased datasets. For example, Giusti et al. (2020) trained a CNN model to predict the surface roughness (Ra) of a manufactured product [12]. The model was trained on publicly available datasets like W300, ASP, STAVAX, and V720, achieving 98% accuracy. However, the authors noted that Ra and binary quality classification are insufficient for quality control. Future work aims to extend this model to predict more measurements about product surface quality.

With low computational hardware, it is possible to manipulate a manufacturing machine, as demonstrated by Halawi Ghoson et al. (2023) [15]. They used an embedded vision sensor on a robot to manipulate a 3-axis milling machine. The robot initiated a production cycle with only the smart-mounted camera, using image processing approaches. Unlike Wang et al. (2022) [14], this application had limitations due to lighting effects and machine security.

2.2. Dataset Creation

Training AI methods require non-biased datasets. Cai et al. (2021) proposed a generative noisy image model [13]. Their objective was to train a model that adds random noise to an image to augment the dataset, making it more suitable for training. Simultaneously, an operator cannot distinguish if the generated image presents an environmental issue or a process issue. Hence, it is necessary to develop an online edge computing device capable of extracting relevant parameters to create datasets. These datasets might serve as references to enhance AI models since they are based on real images and contextual data.

2.3. Edge Detection

Edge detection is considered an old-fashioned recognition method. It depends on filters like Sobel, Prewitt, etc. The common approach is the Canny edge detector. Abid Hasan et Ko (2016) proposed a method for segmentation using 3D measurement technologies and the Canny edge detector [7]. Images are passed through erosion and dilation filters to remove noise before being processed by the Canny detector. The detector measures edges inside the image, creating a depth map that segments the image into sections corresponding to the tool, part, and machine interior. This technique is under evolution due to false predictions from unwanted random noises, addressed in a detailed study by Kalbasi et Nikmehr (2020) [11]. They developed an adaptable Canny edge detector, increasing contour detection even with noisy images. These results suggest researching a technique using the Canny detector instead of more sophisticated and complex SIFT, SURF, and AI algorithms.

In conclusion, video analytics have the potential to integrate smart manufacturing. According to the studies, three conditions are required: effective edge computing modules, performance devices, and stable lighting. A general expansion method is proposed to allow machine data extraction using classical recognition algorithms. This method is then tested on an SM machine, presenting critical aspects for data measurement in a production line.

3. Proposed Methodology

The review explores the conditions for building an effective image data contextualization method. One requirement is data measurement using edge computing modules.

The data of a non-connected CNC machine are enclosed within the control loop, requiring a generic method to measure essential characteristics. A methodology is proposed to extract relevant information. It is a 4-stage phase, starting with the specific needs, refer to Figure 2.

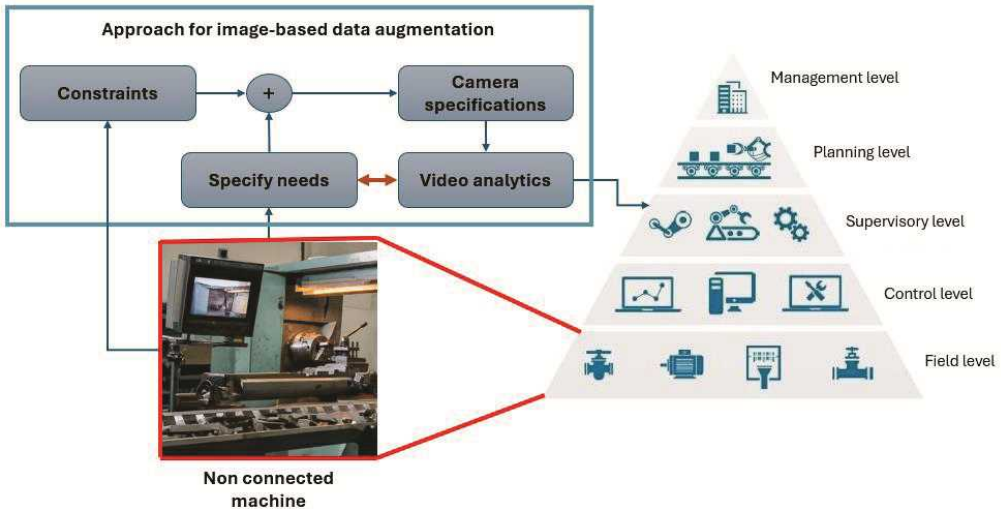


Figure 2. Proposed methodology for an edge computing module.

3.1. Specific Needs

Initially, it is essential to specify the data to identify. This objective depends on each system, as they differ in shape, software, and operational data. Based on these needs, a vision system is investigated. For example, one camera is not sufficient for 3D printer supervision, so two cameras are required to supervise the printed product. The vision module also assesses tool quality or raw part presence.

An optimal place is chosen to mount the camera without affecting the process or data collection. The camera's view angle must prioritize capturing the process. As a result, the machine presents several constraints requiring technical implementations.

3.2. Constraints

Machine constraints play a role in finding the operational data. These constraints are examined to prevent false installations that might affect the supervision device. It is beneficial to correlate between the needs and the constraints. For example, vibration is an important parameter to monitor during manufacturing. As a result, the camera is placed in front of the process to detect the chatter of the workpiece. Additionally, noisy images are prevented by mounting the module on a rigid body. Finally, cable management is also crucial to prevent interference with the production process.

3.3. Camera Specifications

The aim is to ensure data measurement using only a camera and edge detection. Thus, a suitable camera is essential for monitoring a machine and communicating the metadata to the supervision level. Industrial vision systems are often not accessible due to budget or technical support constraints.

The design of a custom vision sensor should fit industry standards. For example, Costa (2018) has identified the Raspberry Pi with its camera as a customizable system [16]. Additionally, this vision module satisfies working-area constraints, according to Kumar et al. (2018) [17]. The module is sized $65 \times 56 \times 12.5$ mm, making it possible to add support or fix it to the walls of a machine without perturbing the process.

Moreover, the literature review points out that lighting has a major effect on images. Thus, it is important to overexpose the scene, ensuring good image quality with no noise [8].

A final condition is the capturing rate. Using the image acquisition ratio (IAR), it is possible to understand the camera's capability to capture subsections of a process. IAR represents the proportionality between the camera's sampling frequency and the machine's working speed (refer to Equation (1)).

$$IAR = \frac{FPS_{camera}(frame\ per\ second)}{Speed_{machine}(rotation\ per\ second)} \quad (1)$$

3.4. Video Analytics

The algorithmic configuration varies between low-level analysis and AI vision. The focus is on low-level analysis, integrating the concepts of edge detection.

For a manufacturing machine, several properties are required. Data contextualization is either on the edge or on remote servers. This approach introduces edge integration. Edge computing is then configured on the vision module using a low computational algorithm, like the Canny detector.

The Canny edge detector is a multi-stage algorithm that combines several techniques to detect edges in images:

1. Noise reduction with Gaussian Blur

A Gaussian filter is applied to smooth the image and reduce noise. The filter kernel is defined by the Gaussian function

$$G(x,y) = \left(\frac{1}{2\pi\sigma^2}\right) * e^{\frac{-(x^2+y^2)}{2\sigma^2}} \quad (2)$$

where sigma controls the degree of smoothing. A larger sigma results in more blurring.

2. Gradient calculation using Sobel filters

Sobel filters are applied to calculate the intensity gradient of the image. Two filters are used, one for the horizontal gradient (G_x) and one for the vertical gradient (G_y). The gradient magnitude (G) and direction (θ) are then calculated:

$$G = \sqrt{\left(G_x^2 + G_y^2\right)} \quad (3)$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (4)$$

3. Non-maxima suppression

Thinning of edges by suppressing non-maximum gradient magnitudes. Only pixels where the gradient magnitude is a local maximum in the direction of the gradient are kept.

4. Double thresholding

Two thresholds are used—a high threshold ($thresh_{high}$) and a low threshold ($thresh_{low}$).

Strong edges (G above $thresh_{high}$) are marked as definite edges. Weak edges (C between $thresh_{low}$ and $thresh_{high}$) are potential edges.

5. Edge tracking by hysteresis

Weak edges are connected to strong edges to form continuous contours. Weak edges not connected to strong edges are discarded.

Binary images reveal a strong ratio for object detection and recognition. Additionally, it serves as a refined image for the Canny detector, refer to Figure 3. In the end, these contextual data are relatively small packets, making it advantageous to use the MQTT protocol for transmission.

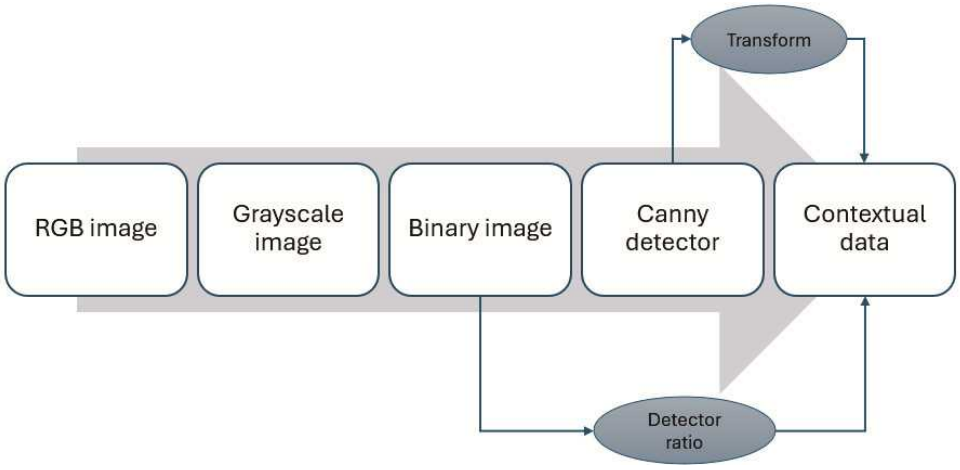


Figure 3. Image flow to extract contextual data.

Some calculations are presented in the next section since these techniques are based on the needs and the parameters to monitor. It is important to validate that the obtained data responds to the identified needs.

The deployment focuses on the CNC lathe machine, which generates noisy images due to environmental disturbances. These machines are commonly deployed in manufacturing lines because they produce products faster.

4. Experimentation

The experimentation is conducted on a pedagogic platform to develop technological bricks for the Industry 4.0 framework. The focus is on the CNC lathe machine of this platform.

4.1. Objectives and Constraints

The methodology is tested on a lathe machine. It treats the case of how to properly integrate the vision system.

First, this machine is an automated unit designed to operate by an operator without the ability to make external connections. The data are enclosed inside the internal closed control loop. This concept applies to many machines used throughout the industrial manufacturing lines. Thus, it is relevant to have external monitoring units, like a vision system, to track the process and ensure its integration within the Industry 4.0 framework. The metadata aids in scheduling maintenance operations, minimizing machine breakdown, and enhancing supervision. Therefore, the vision module must:

- Measure the process parameters
- Assess the quality of the tool
- Check the roughness
- Detect part presence

These needs are common and require an adapted analytical approach. To proceed with sensor configuration, machine constraints are exploited in Table 2.

The edge module is placed inside the machine to guarantee a continuous view of the process. It also prevents missing out on important scenes due to the turret displacement. Figure 4 resumes the general conception where the essential mechanical systems, like the spindle speed and the turret, are presented.

Table 2. Constraints for physical integration of vision system.

Type of Constraint	Description	Resolution
GEOMETRICAL	• Machine compact design. • Little space for operator manipulation.	• Camera facing process. • See machine Figure 4. • Minimal distance 30 cm.
ENERGY	• Input power is 220Vac. • Vision modules require 5Vdc with 1.5Adc.	• Adapter with a rating of 5Vdc and 2 A
CABLE MANAGEMENT	• High magnetic field causing errors in data.	• Wi-Fi connection. • Only power supply.
AIR PRESSURE	• Air pressure to clean the product. • Vibration errors on camera.	• Use a rigid support to fix the camera module.
ACCESS	• Security door closed during manufacturing.	• Camera placed inside the machine. • Refer to Figure 4.

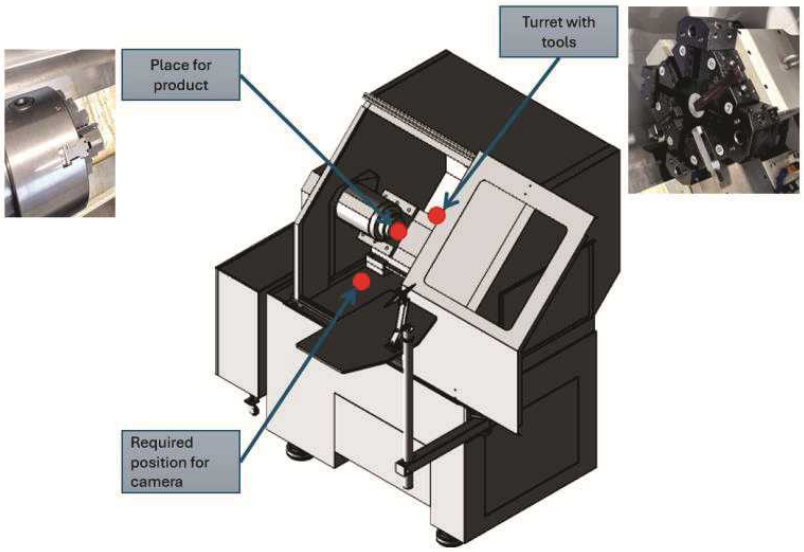


Figure 4. Geometrical constraints of CNC lathe machine from TORMACH, Madison, WI, USA.

4.2. Camera Selection and Validation

A 2-dimensional analysis with one camera is sufficient for a lathe machine. The camera selection, introduced in Section 3.4, defines the sampling frequency and the ability to process the image. The ratio is found by looking at the highest machining parameters of the lathe. In our case, it is 2500 rotations per minute (RPM). For an adequate sampling frequency, IAR should be greater than 1.1, allowing the capturing of an image every rotation (refer to Equation (2)).

$$IAR = \frac{FPS_{camera}}{Speed_{machine}} \tag{5}$$

$$IAR = 1.1 = \frac{FPS_{camera}}{2500(rpm) * \frac{1}{60}} = \frac{FPS_{camera}}{41.66(rotation/second)} \tag{6}$$

$$FPS_{camera} = 1.1 * 41.66 = 45.825\ fps \tag{7}$$

As a result, the selected camera should be rated at least 46 fps. According to Costa (2018), the pi-camera of Raspberry Pi achieves a sampling rate of 60 fps.

After considering the constraints of Table 2 and selecting the Raspberry Pi 4 with its camera module, the installation is realized, as shown in Figure 5.

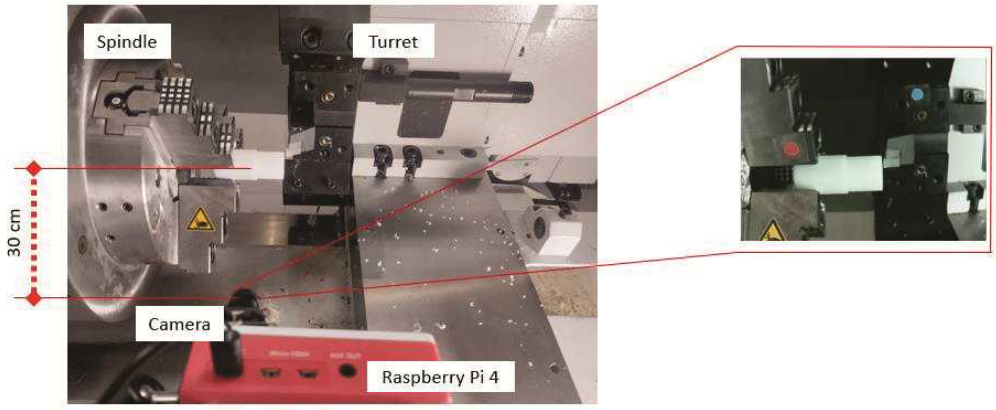


Figure 5. Hardware integration on the lathe machine.

This implementation achieves a sampling frequency of 60 Hz with a control processing unit (CPU) usage of 40%. This means that the potential is not fully used, and the board can handle some video analytics. Simultaneously, the operating system, Raspbian, is fully customizable and adaptable for multi-threading with its 4 cores and 4 Gigabytes (GB) of random-access memory (RAM).

At this stage, the initial steps of the approach are well integrated for the supervision of a lathe machine. The next phase is to define the data treatment flow, allowing the investigation of the process parameters.

4.3. Video Analytics

According to the objective in Section 4.1, a data flow is created to visualize and ensure valid data extraction (refer to Figure 6).

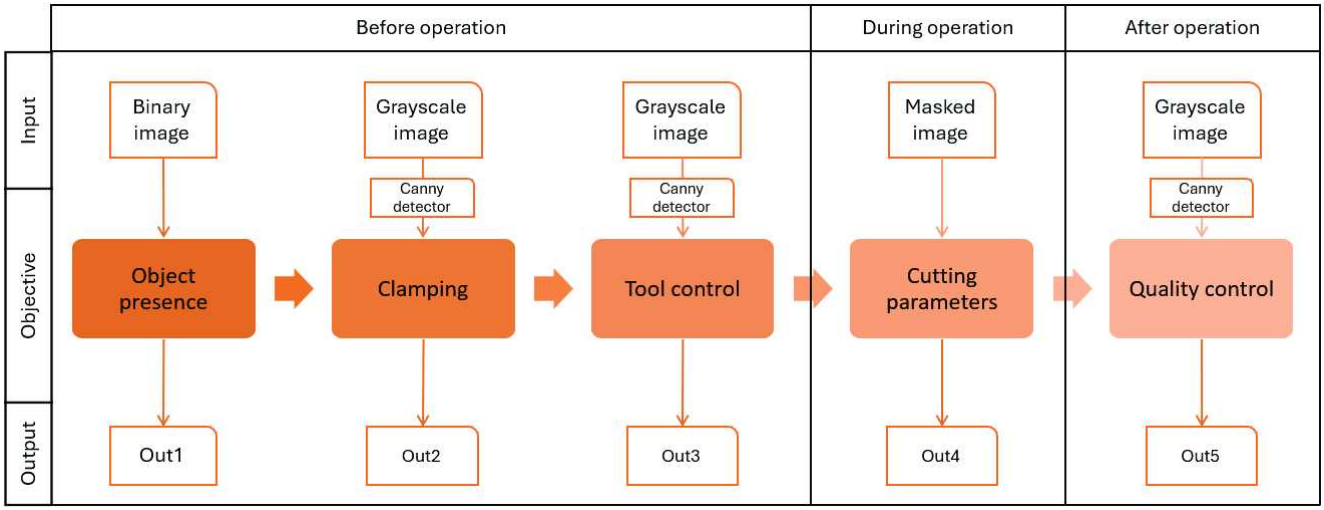


Figure 6. Objectives of video analytics for the lathe machine.

The expected outputs of the detection sequence are summarized in Table 3. These outputs are transferred to the supervisory control level in text format to facilitate the operator's interpretations.

Table 3. Output characterization for video analytics.

Output	Positive Detection	Alternative Detection
OUT 1	Raw part detected	Manufactured part detected
OUT 2	Material right alignment	Misalignment detected
OUT 3	Good tool quality	Bad tool quality
OUT 4	Cutting speed; Depth of cut; Feed rate	No landmarks detected
OUT 5	Acceptable roughness	Too many surface glitches

4.3.1. Object Presence

Detecting objects in the chuck allows verification if the automatic feeding system has performed its task correctly or if the workpiece has been ejected during the production process. This detection permits the control unit to stop the machine, preventing unwanted time loss and potential malfunctions.

The described studies are based on CNN for object detection. However, they require high-performance servers to turn efficiently, which is not available in most manufacturing machines. A less computational approach is to use the black-to-white ratio (BWR), defined as follows:

$$BWR = \frac{black\ pixels}{white\ pixels} \tag{8}$$

This ratio is only applicable to binary images. The original image is transformed into grayscale and then converted to a binary image. This approach efficiently detects material presence with a total computational time of 10 milliseconds per frame (refer to Figure 7).

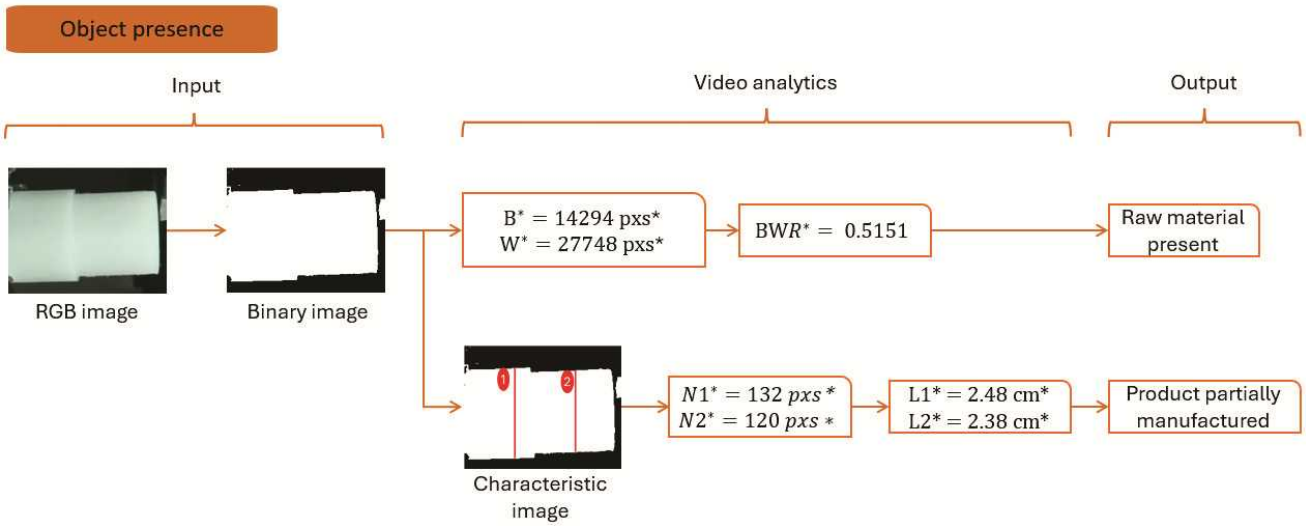


Figure 7. Analytics to detect object presence and manufactured parts. * B—black pixels; W—white pixels; BWR—black-to-white ratio; N1—number of pixels for segment 1; N2—number of pixels for segment 2; L1—length of segment 1; L2—length of segment 2; pxs—abbreviation for pixels; cm—abbreviation for centimeters.

If the BWR is lower than 0.80, it means a product is detected inside the chuck. Otherwise, no material is detected. Correlating this result with a start operation button enhances machine intelligence by alerting any material conditions and object ejection during the process.

Simultaneously, the transformation from pixel to centimeter is given using the theorem of Thales, explained in Figure 8.

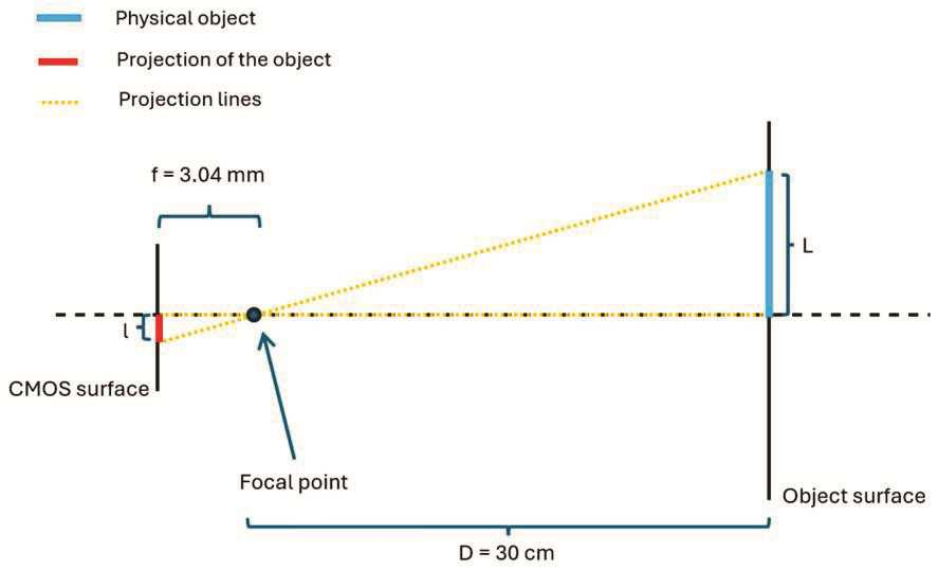


Figure 8. Length proportionality between real object and image.

This diagram shows the detection of lengths L1 and L2; see Figure 7. Using the theorem of Thales, it is given:

$$\frac{f}{D} = \frac{l}{L} \tag{9}$$

$$L = \frac{l * D}{f} = l * \frac{30}{3.04 * 10^{-1}}$$

To obtain L in cm, it is crucial to know the camera specifications about the pixel size. It allows the transformation of the length measured in pixels into micrometers. In this case, it is around 1.12 μm.

4.3.2. Clamping

Clamping issues affect the final product quality. Therefore, detecting any misalignment of the raw part before manufacturing is advantageous. Clamping is detected using the Hough transform and geometric formulas. The Hough transform involves five steps successive steps:

- Step 1: edge detection using the optimized Canny detector algorithm.
- Step 2: extract the line parameters from the edge image by creating a voting function.
- Step 3: select the thresholds to consider true lines and omit false positives.
- Step 4: apply the threshold to the extracted lines and obtain the true positives.
- Step 5: draw the line on the image and save it locally.

The presence of a part indicates that the horizontal surface of the raw material receives the most votes as a line. Simultaneously, it is crucial to know the object's presence to avoid algorithm misinterpretation.

Applying the transform to the video stream, five lines are detected: three horizontals and two verticals (refer to Figure 9). The vertical lines showcase a well-placed part, while the horizontal lines highlight a 3° deviation. The algorithm sends a warning to the operator to consider adjustments.

Additional experiments are conducted to detect the tolerance, which is around $\pm 0.14°$ on average. The angle order is validated to prevent misinterpretation. The measurement is deployed at two thresholds, 50 and 80, for accurate identification. The total response time is found to be 2 milliseconds.

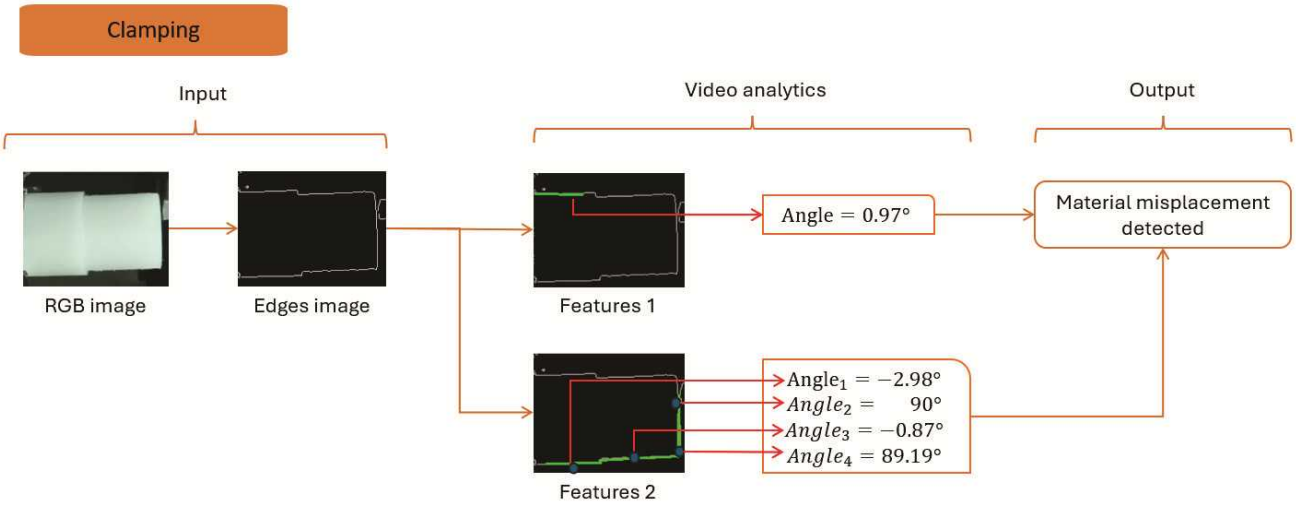


Figure 9. Analytics to part position and orientation.

Based on the lighting effects, it is possible to tune the thresholds of the Hough transform. In this use case, we overexposed the scene. Then, the algorithm is parameterized by setting the threshold of votes, where a line is defined as a true line.

4.3.3. Tool Control

Tool supervision is the capability to predict whether a tool performs machining well. In AI vision, the Siamese Neural Network (SNN) compares a reference tool state and the current tool state to verify tool wear. However, it is possible to perform edge detection combined with the Hough transform to find the tool angle and deduce the tool wear. For example, the tools of a lathe machine have specific geometric reference profiles. If the tool sustains damage, the angle will vary drastically; see Figure 10.

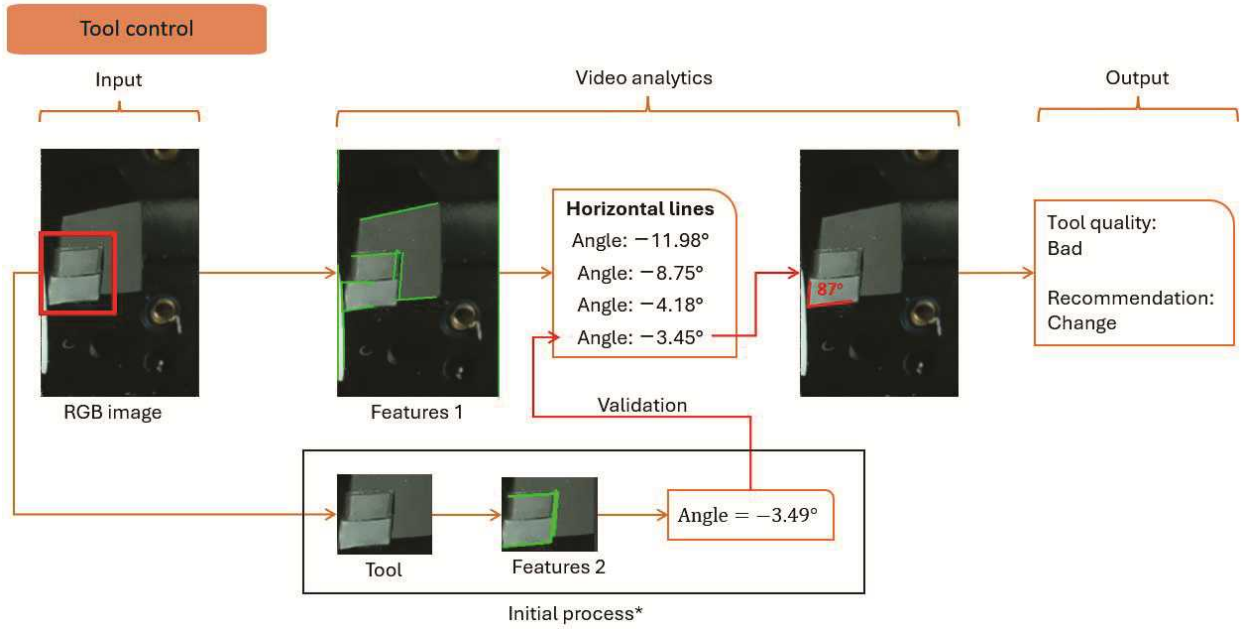


Figure 10. Analytical approach to measure the tool angle. * Initial process is only launched once to validate the position of angles extracted from Features 1.

For the detection of the tool angle, it is essential to validate the output angle order. Another experiment focuses on the detection of the tool and detecting the angle. The

measured angle is found to be always the last one predicted by the essential algorithm. The detection time is measured around 1.5 milliseconds.

4.3.4. Cutting Parameters

Measuring machine parameters is crucial since the machine does not allow external communication. The lathe machine has three parameters: spindle rotation, cutting speed, and depth of cut.

A blue landmark is placed to detect the tool. The vision module detects the blue color based on specified masks. The cutting speed is measured by detecting the tool displacement in the image. For example, at time T1, the tool is at position P2; at time T2, the tool moved to position P2′ (refer to Figure 11).

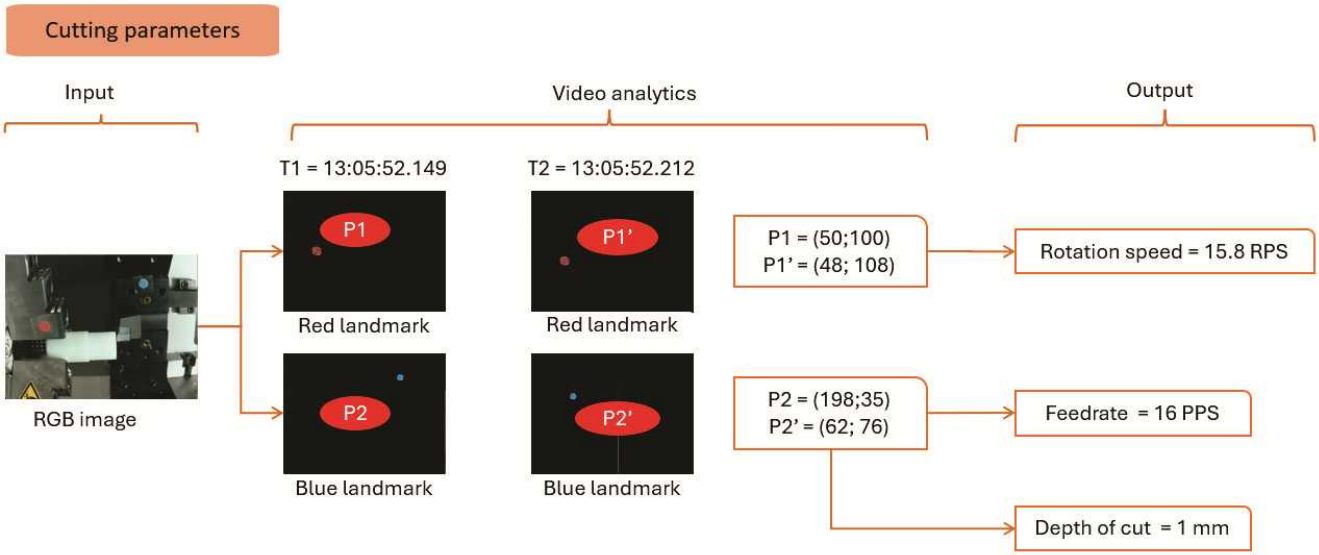


Figure 11. Analytical approach to measure the operational parameters.

The cutting speed is found:

$$cutting\ speed = \frac{distance}{time}\ in\ PPS\ \left(\frac{pixels}{second}\right) \tag{10}$$

where

$$distance = \sqrt{(x_{P2'} - x_{P2})^2 + (y_{P2'} - y_{P2})^2}\ in\ pixels \tag{11}$$

$$time = T2 - T1\ in\ seconds$$

The unit does not conform to the physical value. Based on the camera configuration, mapping allows the discovery of the speed of rotation per minute.

The machine's rotational speed is measured using the chuck's diameter and the time for one rotation. With an IAR greater than 1, one image is captured per turn. A red landmark is placed on it to detect the spindle. The vision module measures the rotation by timing the red circle's exact position in the image.

Finally, the depth of cut is found by evaluating the pixel difference between two consecutive horizontal passes. In this case, it is found to be 2 pixels equivalent to 1 mm.

4.3.5. Quality Control

Quality control is a critical evaluation that presents a wide study area to automate the process. Giusti et al. (2020) used CNN models to find the roughness [12]. However, deploying it on an edge device reduces performance drastically. The Hough transform

method detects lines in an image, representing deviations from the final surface (refer to Figure 12).

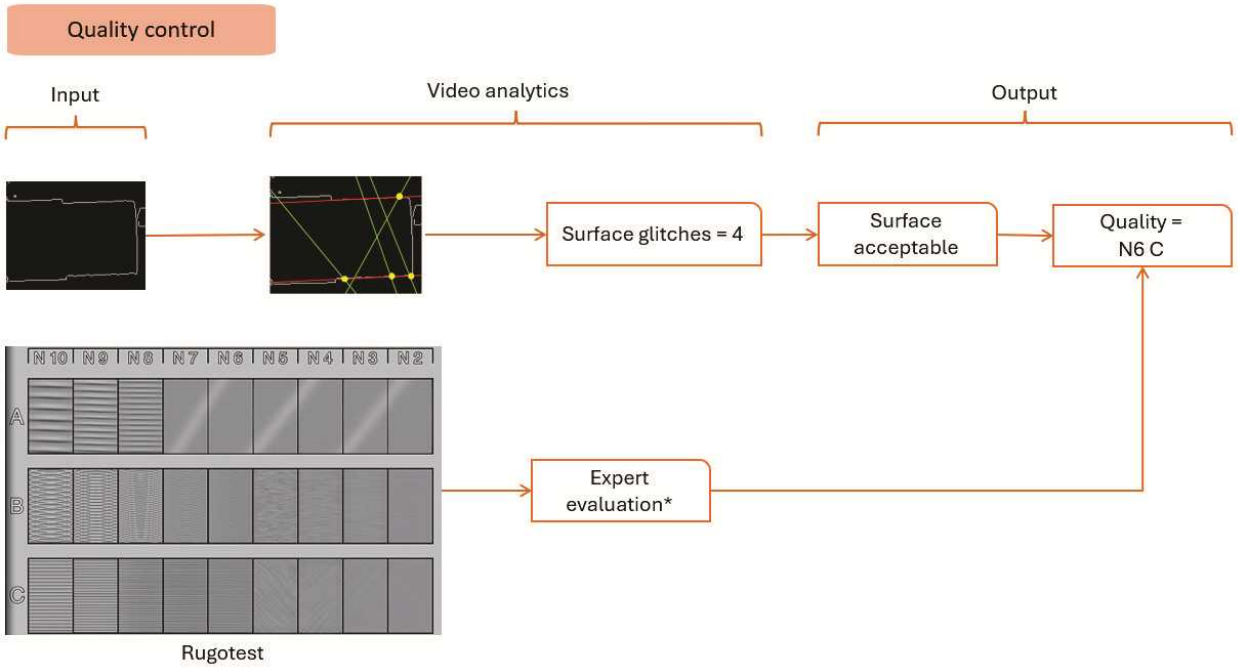


Figure 12. Analytical approach to measuring the quality of the surface. * Expert evaluation is still done manually since the detection of stripes is being researched as a future extension.

The detection sequence is divided into idle and production states. Figures 7–12 summarize the detection approaches during these phases. The data measured are contextual and require a total time of 100 milliseconds per image. The following section investigates the importance of the approach relative to other studies.

5. Discussion

The proposed method augments data communication for unconnected CNC machines, evaluating camera integration limitations. Additionally, a suitable edge algorithm responds to the defined needs.

Our setup detects the chuck holding a partially machined product with an incorrect clamping angle. The machine tool is then examined, and a notice is sent to the machinist about its lifecycle. This preprocess analysis serves as initialization before manufacturing. During the process, the camera measures the cutting parameters, validating correct execution. Finally, the system assesses product quality, detecting possible surface imperfections and estimating roughness based on expert opinion.

The proposed approach responds to the five identified needs for machine supervision. The algorithm has a total timespan of 100 milliseconds, ensuring real-time response and data contextualization. Compared to other approaches like SIFT, SURF, RANSAC, and CNN, our model performs better on an edge device with limited resources. Table 4 summarizes the comparison between literature methods and our approach.

Table 4. A comparison between recent studies and our approach.

Approach	Reference	Hardware	Time	Accuracy
SIFT + RANSAC	[8]	CPU @ 1.5 GHz 4 Gb RAM	900 ms	81%
CNN	[9]	CPU @ 1.5 GHz 4 Gb RAM	2 s	98%
SURF	[10]	CPU @ 1.5 GHz 4 Gb RAM	800 ms	82%
Our approach	--	CPU @ 1.5 GHz 4 Gb RAM	100 ms	95%

The properties are obtained by reimplementing their methods on our machine with the Raspberry Pi of CPU @1.5 GHz and 4 GB RAM. The accuracy for feature point detection is low, with an execution time near 1 s. With AI vision, the controller heated up, presenting an execution time of 2 s. Our approach is suitable for the edge setup and is known to have low computational power.

6. Limitations

The algorithm can fail to assess machine parameters without scene overexposure. Additionally, some algorithm glitches occur during angle measurements and quality assessment. These deviations are avoided by considering all environmental constraints, such as lighting, shadows, noise, and offsets.

Another major limitation occurs when the manufacturing machine's shape changes. Since the camera extracts features from a 2D process considering the symmetrical shape of products, for a 3-axis milling machine, we require updating the 5-stage algorithm to fit geometrical specifications.

The transition toward Industry 4.0 is limited by the lack of data availability from old manufacturing machines. Our approach proves the concept of data extraction from such a system, helping industries achieve a smoother transition with less financial burden.

7. Conclusions

The paper aimed to define an approach to improve the integration of video analytics in manufacturing. We were able to implement a 4-stage method that allows the digitization of any non-connected machine. This scope is defined by the Reference Architecture Model Interface 4.0 (RAMI 4.0). Thus, it was possible to transfer essential data to the supervisory control level of a production line. These data serve to remotely monitor machines and automate critical scenarios.

The implementation focuses on precisely defining the objective to prevent over-calculation. In addition, it is optimized to use the full potential of a low-cost edge computing device. In short, the Raspberry Pi turned with a total computational power of 75% and near real-time processing with 100 milliseconds per image. The standards of the industry are also considered, so the MQTT protocol is applied for data transfer.

In conclusion, the external smart vision system upgrades closed control systems like CNC machines. We explained the deployment of a vision system, extracting metadata. In our use case, the lathe machine is now equipped with an advanced monitoring device describing object presence, clamping, tool state, operational parameters, and surface roughness. With the wireless edge device, the machine becomes interoperable, transforming it into a CPS. This approach allows a gradual evolution toward a digitalized version with adequate communication and data processing.

8. Perspectives

Using the data extracted from a CNC machine, a data pool is created in real time. This data pool serves as a reference to train more advanced approaches, like CNN and DCNN, which help to reorchestrate the manufacturing process. These approaches require fog and

cloud computing. Simultaneously, more complex AI models can be trained to refine the quality control procedure.

Author Contributions: Conceptualization, N.H. and K.B.; methodology, N.H., K.B. and V.M.; software, N.H.; validation, K.B., C.L. and V.M.; formal analysis, K.B.; investigation, V.M.; resources, C.L.; data curation, N.H.; writing—original draft preparation, N.H.; writing—review and editing, K.B. and N.H.; visualization, N.H.; supervision, K.B. and V.M.; project administration, K.B.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All codes and configurations are made available upon request for confidentiality reasons.

Conflicts of Interest: The authors declare no conflict of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Tambare, P.; Meshram, C.; Lee, C.-C.; Ramteke, R.J.; Imoize, A.L. Performance Measurement System and Quality Management in Data-Driven Industry 4.0: A Review. *Sensors* **2021**, *22*, 224. [CrossRef] [PubMed]
2. Mirbod, M.; Ghatari, A.R.; Saati, S.; Shoar, M. Industrial parts change recognition model using machine vision, image processing in the framework of industrial information integration. *J. Ind. Inf. Integr.* **2022**, *26*, 100277. [CrossRef]
3. Caggiano, A.; Zhang, J.; Alfieri, V.; Caiazzo, F.; Gao, R.; Teti, R. Machine learning-based image processing for on-line defect recognition in additive manufacturing. *CIRP Annals* **2019**, *68*, 451–454. [CrossRef]
4. Korzun, N.B.E.D. Event-Driven Video Services for Monitoring in Edge-Centric Internet of Things Environments. In Proceedings of the 2019 25th Conference of Open Innovations Association (FRUCT), Helsinki, Finland, 5–8 November 2019; pp. 47–56. [CrossRef]
5. Kim, H.; Jung, W.-K.; Choi, I.-G.; Ahn, S.-H. A Low-Cost Vision-Based Monitoring of Computer Numerical Control (CNC) Machine Tools for Small and Medium-Sized Enterprises (SMEs). *Sensors* **2019**, *19*, 4506. [CrossRef] [PubMed]
6. Mazumder, J. Design for Metallic Additive Manufacturing Machine with Capability for "Certify as You Build". *Procedia CIRP* **2015**, *36*, 187–192. [CrossRef]
7. Hasan, S.M.A.; Ko, K. Depth edge detection by image-based smoothing and morphological operations. *J. Comput. Des. Eng.* **2016**, *3*, 191–197. [CrossRef]
8. Nuchitprasitchai, S.; Roggemann, M.C.; Pearce, J.M. Three Hundred and Sixty Degree Real-Time Monitoring of 3-D Printing Using Computer Analysis of Two Camera Views. *J. Manuf. Mater. Process.* **2017**, *1*, 2. [CrossRef]
9. Ceruti, A.; Liverani, A.; Bombardi, T. Augmented vision and interactive monitoring in 3D printing process. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2016**, *11*, 385–395. [CrossRef]
10. Delli, U.; Chang, S. Automated Process Monitoring in 3D Printing Using Supervised Machine Learning. *Procedia Manuf.* **2018**, *26*, 865–870. [CrossRef]
11. Kalbasi, M.; Nikmehr, H. Noise-Robust, Reconfigurable Canny Edge Detection and its Hardware Realization. *IEEE Access* **2020**, *8*, 39934–39945. [CrossRef]
12. Giusti, A.; Dotta, M.; Maradia, U.; Boccadoro, M.; Gambardella, L.M.; Nasciuti, A. Image-based Measurement of Material Roughness using Machine Learning Techniques. *Procedia CIRP* **2020**, *95*, 377–382. [CrossRef]
13. Cai, Y.; Hu, X.; Wang, H.; Zhang, Y.; Pfister, H.; Wei, D. Learning to Generate Realistic Noisy Images via Pixel-level Noise-aware Adversarial Training. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 3259–3270. [CrossRef]
14. Wang, Z.; Xu, Y.; Ma, X.; Thomson, G. Towards Smart Remanufacturing and Maintenance of Machinery—Review of Automated Inspection, Condition Monitoring and Production Optimisation. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; pp. 1731–1738. [CrossRef]
15. Ghoson, N.H.; Hakam, N.; Shakeri, Z.; Meyrueis, V.; Loubère, S.; Benfriha, K. Towards Remote Control of Manufacturing Machines through Robot Vision Sensors. *Proc. Des. Soc.* **2023**, *3*, 3601–3610. [CrossRef]
16. Costa, D.G. On the Development of Visual Sensors with Raspberry Pi. In Proceedings of the 24th Brazilian Symposium on Multimedia and the Web 2018, New York, NY, USA, 16 October 2018; pp. 19–22. [CrossRef]
17. Kumar, V.; Wang, Q.; Wang, M.; Rizwan, S.; Ali, S.; Liu, X. Computer Vision BASED object Grasping 6DoF Robotic Arm Using Picamera. In Proceedings of the 2018 4th International Conference on Control, Automation and Robotics (ICCAR), Auckland, New Zealand, 20–23 April 2018. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

A Novel Model for Instance Segmentation and Quantification of Bridge Surface Cracks—The YOLOv8-AFPN-MPD-IoU

Chenqin Xiong [1], Tarek Zayed [1], Xingyu Jiang [2], Ghasan Alfalah [3] and Eslam Mohammed Abelkader [4,*]

[1] Department of Building and Real Estate, Faculty of Construction and Environment, The Hong Kong Polytechnic University, Kowloon, Hong Kong 999077, China; chenqin.xiong@connect.polyu.hk (C.X.); tarek.zayed@polyu.edu.hk (T.Z.)
[2] College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China; jiangxy@spacestar.com.cn
[3] Department of Architecture and Building Science, College of Architecture and Planning, King Saud University, Riyadh 145111, Saudi Arabia; galfalah@ksu.edu.sa
[4] Structural Engineering Department, Faculty of Engineering, Cairo University, Giza 12613, Egypt
* Correspondence: eslam_ahmed1990@cu.edu.eg

Abstract: Surface cracks are alluded to as one of the early signs of potential damage to infrastructures. In the same vein, their detection is an imperative task to preserve the structural health and safety of bridges. Human-based visual inspection is acknowledged as the most prevalent means of assessing infrastructures' performance conditions. Nonetheless, it is unreliable, tedious, hazardous, and labor-intensive. This state of affairs calls for the development of a novel YOLOv8-AFPN-MPD-IoU model for instance segmentation and quantification of bridge surface cracks. Firstly, YOLOv8s-Seg is selected as the backbone network to carry out instance segmentation. In addition, an asymptotic feature pyramid network (AFPN) is incorporated to ameliorate feature fusion and overall performance. Thirdly, the minimum point distance (MPD) is introduced as a loss function as a way to better explore the geometric features of surface cracks. Finally, the middle aisle transformation is amalgamated with Euclidean distance to compute the length and width of segmented cracks. Analytical comparisons reveal that this developed deep learning network surpasses several contemporary models, including YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and Mask-RCNN. The YOLOv8s + AFPN + MPDIoU model attains a precision rate of 90.7%, a recall of 70.4%, an F1-score of 79.27%, mAP50 of 75.3%, and mAP75 of 74.80%. In contrast to alternative models, our proposed approach exhibits enhancements across performance metrics, with the F1-score, mAP50, and mAP75 increasing by a minimum of 0.46%, 1.3%, and 1.4%, respectively. The margin of error in the measurement model calculations is maintained at or below 5%. Therefore, the developed model can serve as a useful tool for the accurate characterization and quantification of different types of bridge surface cracks.

Keywords: surface cracks; bridges; YOLOv8s-Seg; asymptotic feature pyramid network; minimum point distance; middle aisle transformation

1. Introduction

Hong Kong has emerged as one of the most economically advanced regions in the past five decades due to its unique cultural heritage and strategic location. This remarkable progress has been accompanied by significant improvements in urban infrastructure and facilities, including the construction of bridges and roads [1]. While urban development has undeniably brought convenience to the lives of citizens, it has also given rise to a pressing issue—the maintenance of bridges. In our daily lives, numerous factors, such as cyclic loading, fatigue stresses, and adverse long-term environmental conditions, have hastened the deterioration of bridge surfaces, resulting in problems like cracks, ruts, and potholes [2]. Among these concerns, cracks hold particular significance due to their direct impact on concrete structures' safety, functionality, and durability [3]. Cracks provide an

entry point for corrosive chemicals within the concrete, allowing water and de-icing salts to penetrate bridge decks, potentially causing damage to superstructures and aesthetic elements [4]. Since cracks are a critical indicator of a structure's condition, they play a pivotal role in structural health monitoring [5]. Consequently, conducting a rigorous study to assess the extent and severity of cracks is essential for evaluating the condition of bridges and maintaining a comprehensive database for long-term bridge inspections and analysis. As a result, numerous countries have established comprehensive maintenance plans for their bridges. For example, the United States mandates biennial bridge inspections in compliance with American Association of State Highway and Transportation Officials (AASHTO) requirements [6]. In the United Kingdom, bridge inspections occur every one to three years in accordance with established standards [7]. In China, specific standards dictate the detection and quantification of bridge cracks wider than 0.2 mm [8].

In practical terms, due to its simplicity, human visual inspection remains the predominant method for monitoring bridge health [9]. However, this method heavily relies on expertise, experience, and subjectivity, which can result in imprecise and unreliable assessment outcomes [10,11]. Furthermore, visual inspection is often impractical for inaccessible areas of a bridge, such as columns and intersections [12]. In such cases, bridge inspectors resort to deploying inspection vehicles as manual operation platforms to measure the size of bridge cracks. Nevertheless, this approach has its own set of limitations, including traffic disruptions, time consumption, and high maintenance costs [13]. The advent of computer vision models has paved the way for automatic inspection technologies, offering enhanced efficiency and precision in contrast to manual measurements, which are susceptible to inaccuracies and subjectivity [14,15].

However, the application of these technologies presents various challenges and limitations. Firstly, many studies in this field focus on crack detection under ideal conditions, often on surfaces consisting entirely of concrete or asphalt. In real-world scenarios, crack detection tasks frequently entail complex backgrounds with diverse elements, such as tree leaves and varying lighting conditions, rendering the accurate distinction of cracks challenging [16]. Furthermore, traditional bridge distress inspection commonly involves image processing techniques employing edge detection and image thresholding. Cracks are identified based on changes in edge gradients derived from the intensity difference relative to the background and are extracted through threshold segmentation [17]. However, this method is significantly affected by environmental factors during image collection, including variations in lighting conditions and the presence of oil stains [3]. The current study introduces an integrated framework tailored explicitly for the segmentation of road cracks and the quantification of surface features, with a particular emphasis on addressing the challenges posed by complex backgrounds. In contrast to existing research in the same domain, our study presents several noteworthy contributions:

(1) We provide a comprehensive literature review that delves into the realms of bridge crack detection and segmentation, offering a thorough understanding of the existing body of knowledge in this field.

(2) To enrich the research landscape, we have meticulously curated a new bridge crack dataset customized to the distinctive context of Hong Kong. This dataset encompasses a diverse array of crack samples, faithfully representing the complexities encountered in real-world scenarios, including various crack sizes and intricate crack backgrounds.

(3) Our study pioneers the development of a novel YOLOv8-AFPN-MPD-IoU, an advanced approach facilitating precise and timely detection and segmentation of cracks within Hong Kong's bridges with complex backgrounds, thus enhancing process efficiency. The AFPN is incorporated into the YOLOv8 neck, effectively reducing information gaps between non-adjacent layers and facilitating improved feature fusion. The MPD-IoU (minimum point distance intersection over union) loss function is deployed to solve the problem that the predicted bounding box possesses the same aspect ratio as the ground-truth bounding. In terms of recognition precision, model size, and reference times, the proposed model emerges as the optimal choice for deploying portable devices in practical applications.

(4) Furthermore, we employ a distance transform method (DTM) to accurately measure the length and width of segmented road cracks at the pixel level. This approach enhances precision and minimizes the impact of varying lighting conditions, a crucial aspect of our methodology.

2. Literature Review

Crack identification is mainly divided into traditional methods based on digital image processing and methods based on deep learning. Digital image processing methods based on edge detection and threshold segmentation are highly susceptible to ambient environmental conditions and image quality requirements. Additionally, these techniques may fail when encountered with real-world scenarios such as tree leaves and varying lighting conditions [18]. Likewise, they remain highly dependent on the design of feature descriptors, and therefore, they are difficult to be generalized [19–21]. In this respect, some previous research attempts relied on the use of threshold segmentation [22], Otsu [23], improved Otsu [24–26], edge detectors [27,28], improved watersheds [29], artificial neural networks [30,31], hybrid artificial neural networks [32], and support vector machines [33–35] for crack detection and assessment.

Aside from data dependence, deep learning addresses the limitations inherent in the typical machine learning models mentioned above. The advancement of graphical processing units (GPUs) has significantly accelerated image processing on computers. Consequently, deep learning techniques can now be effectively employed for tasks such as image object detection and segmentation, including the management of building construction [36]. Convolutional neural networks (CNNs) are among the popular techniques used to extract category and location information from images in object detection tasks. However, the process of generating region proposals using selective search remains slow and exhibits limited accuracy for two-stage detectors [37]. Rosso et al. [38] presented a ResNet-50-based framework for defect classification in ground penetrating radar profiles. The authors implemented bi-dimensional Fourier transform as a preprocessing convolution operation, and vision transformer architecture was incorporated, resulting in better localization of defects. In another research effort, Park et al. [39] introduced a machine-learning-based model for predicting depths of concrete cracks in thermal images. The four machine learning models included AdaBoost, gradient boosting, random forest, and multilayer perceptron. The AdaBoost model demonstrated the highest prediction accuracy with a determination coefficient and mean absolute percentage error of 98.96% and 0.6%, respectively. They also showed that the combination of principal component analysis with AdaBoost or gradient could properly detect microcracks.

Among the bounding box detection models, Zhang et al. [40] created a single-stage detector based on you only look once (YOLOv3) to locate bridge surface defects like cracks, spalling, pop-outs, and exposed rebar. In another study, Deng et al. [41] utilized YOLOv2 to locate cracks in real-world images contaminated with handwriting scripts. In a third study, Teng et al. [42] investigated the use of 11 pre-trained deep learning networks to be leveraged as feature extractors of YOLOv2. Among them, there were Alex Net, VGG16, VGG19, ResNet50, ResNet101, Google Net, etc. They deduced that the integration of ResNet18 and YOLOv2 was deemed appropriate in terms of computational accuracy and efficiency. Similarly, Qiu and Lau [43] elucidated that ResNet50-based YOLOv2 and YOLOv4-tiny demonstrate exceptional accuracy, fast processing speed, and notable proficiency in detecting small cracks in tiled sidewalks for crack detection. Xiang et al. [44] addressed various challenges encountered by UAV-based crack detection networks, including low efficiency and accuracy stemming from road noise such as shadows, occlusions, and low contrast. Consequently, the YOLO architecture is recognized for its capability to identify cracks amidst complex backgrounds.

However, the bounding box outcomes provided by crack recognition lack the potential for further quantitative analysis, thereby rendering crack recognition less suitable as a crack measurement algorithm [11]. The task of object segmentation evolves from object detection, requiring a comprehensive understanding of the image scene at the pixel level. In crack segmentation tasks, the input image typically contains only cracks and background. The resulting output consists of binary values of 1 and 0, representing crack pixels and background pixels,

respectively [45]. As a result, the outcomes of crack segmentation can facilitate subsequent geometric calculations of crack length and width. The human brain rapidly processes distinct shapes and irregular contours present in objects, similar to an objective segmentation task. Consequently, creating a detection frame outside the target object is deemed unnecessary and incongruent with human visual interpretations [46]. Therefore, the utilization of the YOLOv8-seg model in bridge inspection endeavors represents a promising initiative.

3. Research Methodology

This research study comprises three distinct phases within its framework: a systematic review, data collection, and crack measurement. Illustrated in Figure 1, this framework delineates a structured approach to bridge crack segmentation and measurement. The model developed is grounded in a conceptualization revolving around three essential modules. The initial module involves a comprehensive examination of existing models for detecting and quantifying surface cracks on bridge decks. This critical review not only identified research gaps, future research directions, and clear objectives but also prompted the utilization of a publicly available database to augment the diversity and complexity of image scenarios used in experimental settings. However, currently, the images provided lack sufficient power to fully capture the complex diversity of real-world scenes. Therefore, a set of supplementary images featuring various bridge defects was curated specifically for this purpose, utilizing data augmentation—an extensively employed method to mitigate overfitting in deep learning models while enhancing their diversity and generalizability [47]. The images were captured at various locations across Hong Kong, encompassing a diverse array of environmental factors. These include variations in sunlight conditions, diverse surface textures, different bridge locations, instances of graffiti, and occurrences of water damage. Various transformations, such as lighting adjustments, random element addition, noise introduction, flipping, and image bounding box manipulation, were applied to augment the Hong Kong dataset. At this juncture, the integrated dataset combines the publicly accessible benchmark dataset of bridge surface cracks (BSD) sourced from [48] alongside the Hong Kong bridge crack dataset (HKBCD). The third module of this study focuses on developing a novel YOLOv8-AFPN-MPDIoU model for surface crack segmentation, complemented by the utilization of the distance transform method (DTM) for precise calculation of bridge crack width and length.

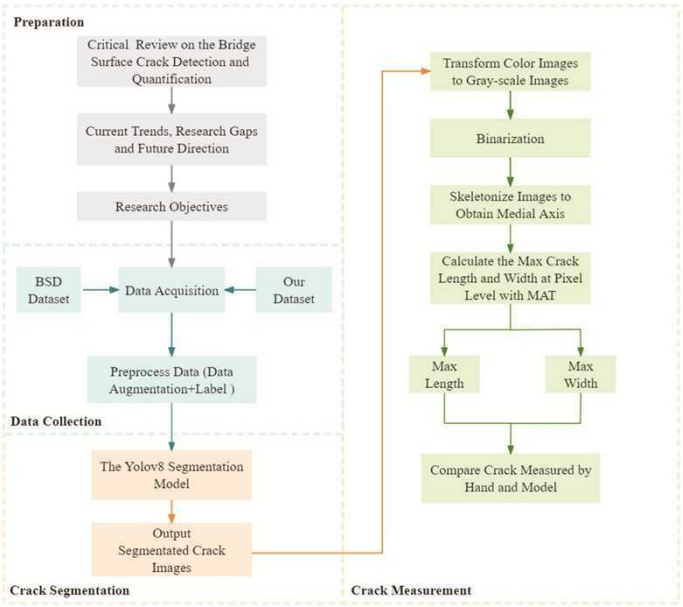


Figure 1. Framework of the developed YOLOv8-APF-MPDIOU model for crack instance segmentation.

4. Model Development

This section delineates the basic components of the developed YOLOv8-AFPN-MPDIoU model.

4.1. Basics of YOLOv8-Segmentation

Architecture of the developed YOLOv8-AFPN-MPDIoU model is depicted in Figure 2. Its distinctive features lie in the following:

1. The YOLOv8 network offers support for object detection, tracking, and various additional tasks, including instance segmentation, image classification, and key-point detection. Similar to YOLOv5, YOLOv8 presents five distinct scales of models (n, s, m, l, x) with increasing depth and width from left to right [49].
2. In alignment with the efficient layer aggregation network (ELAN) design philosophy, YOLOv8 replaces the C3 structure within the YOLOv5 backbone network with a C2f structure. This modification enables YOLOv8 to retain its lightweight characteristics while enhancing the flow of gradients [50]. In comparison to YOLOv5, YOLOv8 demonstrates more pronounced disparities in its head section due to the integration of the widely adopted decoupled head structure.
3. YOLOv8 adopts the Task-Aligned-Assigner strategy for positive sample assignment in loss function calculation [51]. Additionally, it introduces the distribution focal loss (DFL). During training, the strategy of disabling mosaic augmentation in the final ten epochs is incorporated, as inspired by YOLOX, to effectively enhance precision in the data augmentation process.
4. YOLOv8s-Seg is an extension of the YOLOv8 object detection model specifically tailored for performing segmentation tasks. This network draws upon the principles of the YOLACT network to achieve real-time instance segmentation of objects while maintaining a high segment mean average precision [52]. The structural overview of the YOLACT network is presented in Figure 3.
5. The YOLOv8-Seg network, version ultralytics 8.0.201, comprises three primary components: the backbone, the neck, and the head. In the associated GitHub repository, five distinct scale models of the network are available, specifically YOLOv8n-Seg, YOLOv8s-Seg, YOLOv8m-Seg, YOLOv8l-Seg, and YOLOv8x-Seg. In this study, experiments were conducted using YOLOv8-Seg models at various scales to assess the segment mAP50 and model size. Given the minimal presence of cracks in each image, the first four scales of models were utilized to identify the most suitable scale.

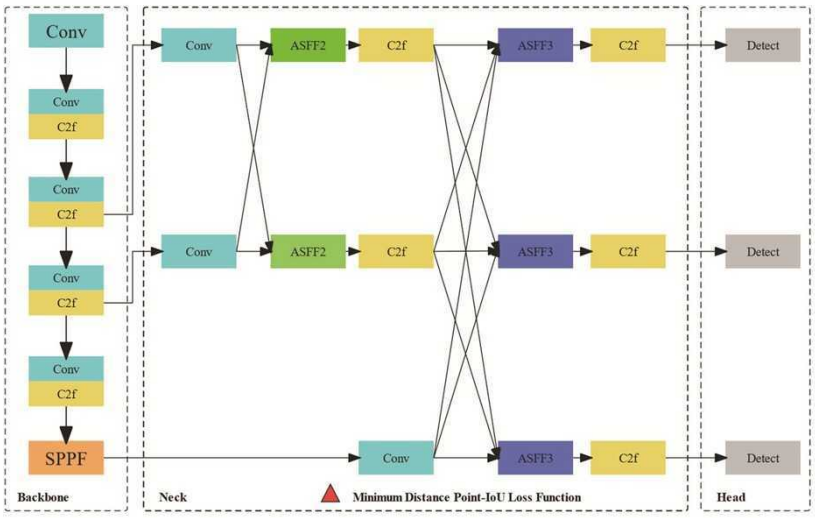


Figure 2. Architecture of the developed YOLOv8-AFPN-MPDIoU model.

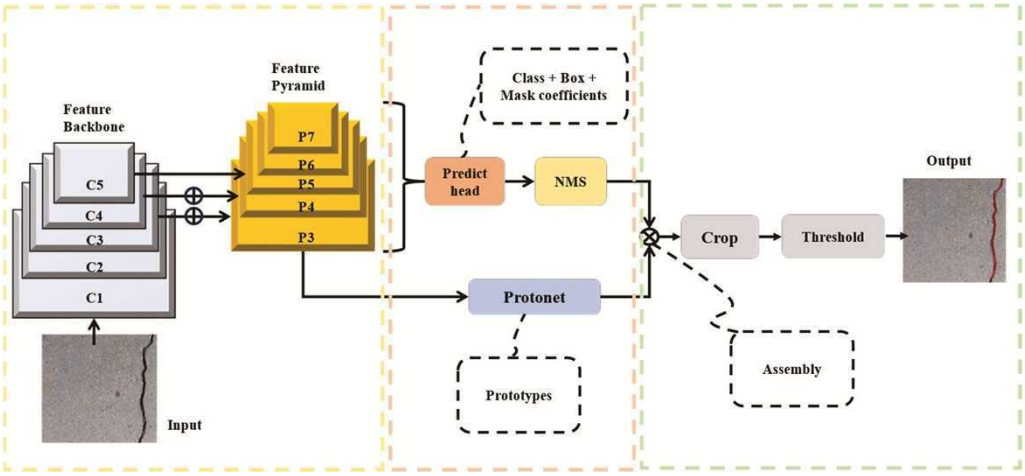


Figure 3. Structure of YOLACT network.

4.2. Asymptotic Feature Pyramid Network

In tasks related to object detection and segmentation, the multi-scale feature extraction process plays a key role in object encoding and scale change processing. A common approach is to utilize well-established top-down or bottom-up feature pyramid networks. However, these methods are susceptible to the problem of feature information loss and deterioration, which in turn adversely affects the integration of information between two non-adjacent layers. Within the prevailing architectures of feature pyramids, the challenge lies in the necessity for high-level features located at the pinnacle of the pyramid to traverse multiple intermediate scales and engage with features at these intermediate levels before finally fusing with the low-level features at the base. Throughout this process of propagation and interaction, there is a risk of the semantic information contained within the high-level features being lost or compromised. In contrast, the bottom-up pathway of the PAFPN (pyramid attention feature pyramid network) introduces a converse challenge: the detailed information originating from the lower-level features could potentially suffer loss or deterioration during the process of propagation and interaction.

4.2.1. Asymptotic Architecture

To address this challenge, the asymptotic feature pyramid network (AFPN) was developed by Yang et al. [53]. The issue stems from the substantial semantic gap between non-adjacent hierarchical features, which is especially pronounced between the lowest- and highest-level features. This gap hinders the effective fusion of non-adjacent hierarchical features. However, the AFPN architecture is designed to be asymptotic, resulting in a closer alignment of semantic information across different hierarchical feature levels during the asymptotic fusion process. This, in turn, mitigates the aforementioned issues. As illustrated in Figure 4, during the bottom-up feature extraction process of the backbone network, the AFPN gradually integrates low-level features, followed by intermediate-level features, before culminating in the fusion with the highest-level feature, which is the most abstract. Black arrows denote convolutions, while blue arrows indicate adaptive spatial fusion. In this proposed model, the AFPN is incorporated into the YOLOv8 neck, effectively reducing information gaps between non-adjacent layers and facilitating improved feature fusion.

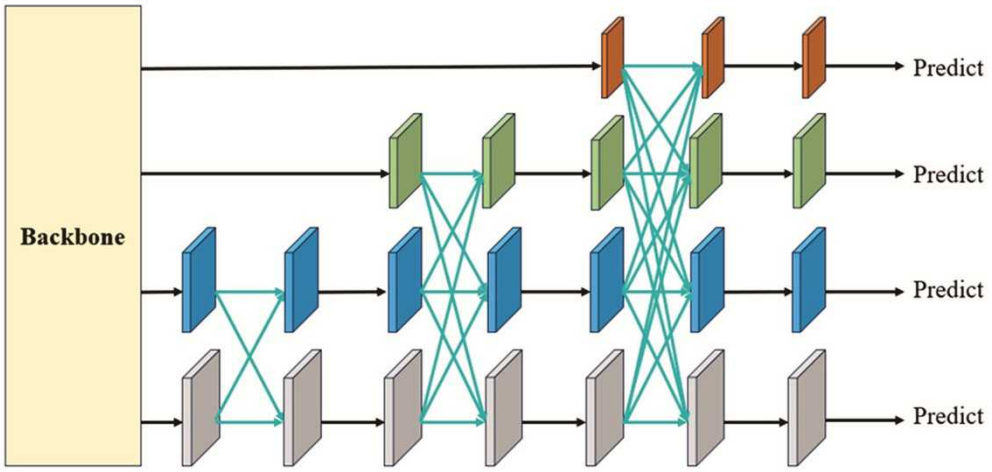


Figure 4. Architecture of the proposed asymptotic feature pyramid network.

4.2.2. Adaptive Spatial Fusion

The ASFF technique is utilized to assign variable spatial weights to different feature levels during multi-level feature fusion, enhancing the importance of critical levels while minimizing the impact of conflicting information from diverse objects. Illustrated in Figure 5, features from three different levels are combined. This Figure serves as an illustration of feature fusion at three different levels, but we can adapt the method as needed for cases with more or fewer levels. Representing the feature vector transitioning from level n to level l at position (i, j) as $x_{ij}^{n \to l}$, the resultant feature vector y_{ij}^{l} is attained through adaptive spatial fusion, formally defined as the linear combination of feature vectors $x_{ij}^{1 \to l}$, $x_{ij}^{2 \to l}$, and $x_{ij}^{3 \to l}$ (refer to Equation (1)).

$$y_{ij}^{l} = \alpha_{ij}^{l} \cdot x_{ij}^{1 \to l} + \beta_{ij}^{l} \cdot x_{ij}^{2 \to l} + \gamma_{ij}^{l} \cdot x_{ij}^{3 \to l} \tag{1}$$

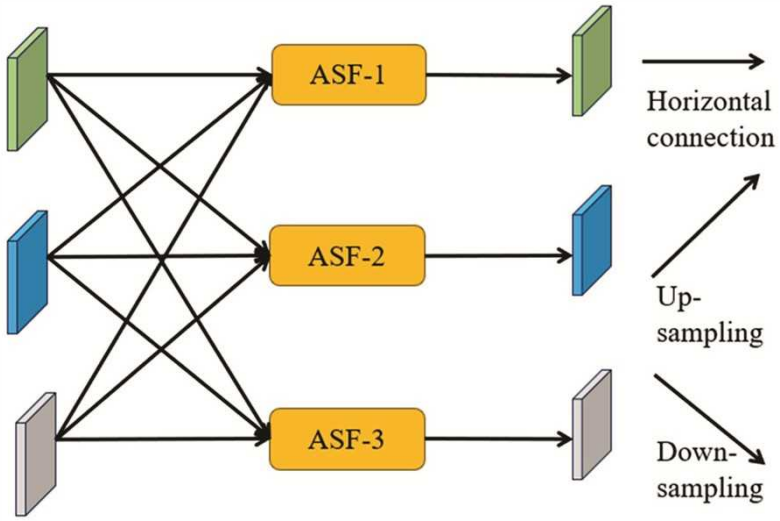


Figure 5. Adaptive spatial fusion operation.

At level α_{ij}^{l}, β_{ij}^{l} and γ_{ij}^{l} represent the spatial weights of the features across the three levels, with the constraint $\alpha_{ij}^{l} + \beta_{ij}^{l} + \gamma_{ij}^{l} = 1$. Due to variations in the number of fused features at each stage of the AFPN, we introduce stage-specific adaptive spatial fusion modules [54]. The adaptive spatial fusion operation is shown in Figure 5.

4.3. Minimum Point Distance-IoU Loss Function

The optimization process of the YOLOv8 model occurs across two dimensions: classification and regression. While the classification loss continues to employ binary cross-entropy loss (BCEL), the regression component integrates focal distribution loss (DFL) and bounding box regression loss (BBRL). Equation (2) comprehensively represents the loss function:

$$f_{\text{Loss}} = \lambda_1 f_{BCEL} + \lambda_2 f_{DFL} + \lambda_3 f_{BBRL} \tag{2}$$

Here, f_{Loss} denotes the total loss; λ_1, λ_2, and λ_3 represent the weighting factors assigned to each loss term; and f_{BCEL}, f_{DFL}, and f_{BBRL} are individual loss functions for binary cross-entropy, focal distribution loss, and bounding box regression loss, respectively. BCEL loss is utilized for classification, assessing the dissimilarity between predicted class probabilities and ground-truth labels. DFL is applied for regression, considering the distribution of predicted bounding box values and focusing on challenging samples. Meanwhile, BBRL aims to minimize the difference between predicted and ground-truth bounding box coordinates.

Bounding box regression (BBR) has established itself as a crucial component in object detection and instance segmentation, serving as a pivotal step in precise object localization. Nevertheless, a substantial limitation of most prevailing loss functions for bounding box regression lies in their inability to be effectively optimized when the predicted bounding box possesses the same aspect ratio as the ground-truth bounding box but exhibits distinct width and height values. To address these aforementioned challenges, we conducted a comprehensive exploration of the geometric attributes associated with horizontal rectangles. Consequently, we introduce a novel bounding box similarity comparison metric, denoted as MPD-IoU (minimum point distance intersection over union), which encapsulates all the relevant factors considered in existing loss functions. This includes considerations such as the overlapping or non-overlapping area, the distance between central points, and variations in width and height, while concurrently simplifying the computational process.

In contrast to the complete intersection over union loss function (C-IoU loss function) employed in YOLOv8, the minimum point distance-IoU loss function (*MPD-IoU* loss function) is employed for comparison against other loss functions based on IoU [55]. Figure 6 demonstrates a visual presentation of the proposed MPD-IoU loss function. Figure 7 provides visual examples of predicted bounding boxes and ground-truth bounding boxes. They share the same aspect ratio but differ in width and height, where k > 1 and k ∈ R. In these visualizations, the yellow box signifies the ground-truth bounding box, while the red box and black box represent the predicted bounding boxes. Mathematically, the key components of the *MPD-IoU* metric are defined using Equations (3)–(8).

$$d_1^2 = \left(x_1^{prd} - x_1^{gt}\right)^2 + \left(y_1^{prd} - y_1^{gt}\right)^2 \tag{3}$$

$$d_2^2 = \left(x_2^{prd} - x_2^{gt}\right)^2 + \left(y_2^{prd} - y_2^{gt}\right)^2 \tag{4}$$

$$w_{gt} = x_2^{gt} - x_1^{gt}, h_{gt} = y_2^{gt} - y_1^{gt}, w_{prd} = x_2^{prd} - x_1^{prd}, h_{prd} = y_2^{prd} - y_1^{prd} \tag{5}$$

$$MPD - IoU = IoU - \frac{d_1^2}{h_{gt}^2 + w_{gt}^2} - \frac{d_2^2}{h_{prd}^2 + w_{prd}^2} \tag{6}$$

$$\mathcal{L}_{MPDIoU} = 1 - MPD - IoU \tag{7}$$

$$f_{Loss} = \lambda_1 f_{BCEL} + \lambda_2 f_{DFL} + \lambda_3 f_{MPD-IoU} \tag{8}$$

where, the overall loss function, denoted as f_{Loss}, is assigned weights (λ_1, λ_2, λ_3) to balance the contributions of binary cross-entropy loss f_{BCEL}, distance-based focal loss (f_{DFL}), and the proposed MPD-IoU-based loss ($f_{MPD-IoU}$). Here, (x_c^{gt}, y_c^{gt}) and (x_c^{prd}, y_c^{prd}) denote the coordinates of the central points of the ground-truth bounding box and the predicted bounding box, respectively. Similarly, w_{gt} and h_{gt} represent the width and height of the ground-truth bounding box, while w_{prd} and h_{prd} denote the width and height of the predicted bounding box.

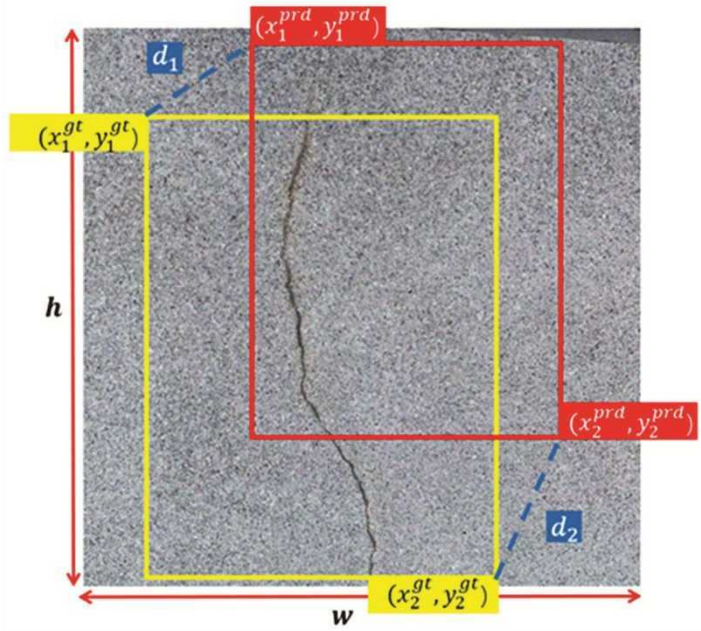


Figure 6. Visual illustration of the proposed $\mathcal{L}_{\text{MPDIoU}}$.

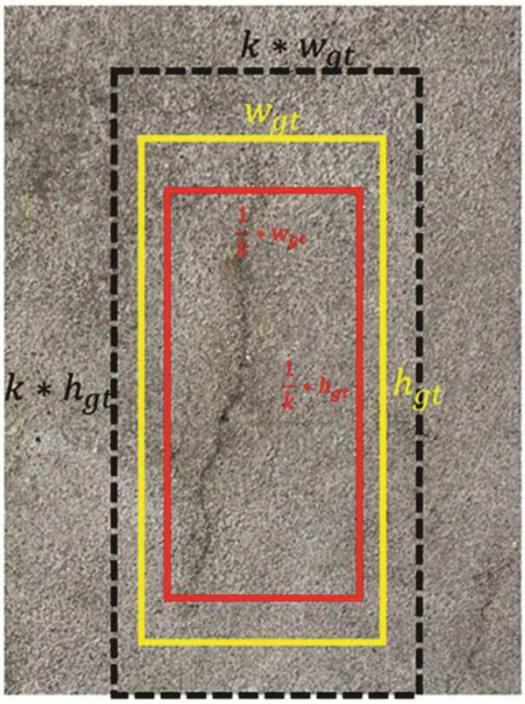


Figure 7. Visual presentation of predicted bounding boxes and ground-truth bounding boxes.

The expressiveness of our $MPD - IoU$ metric is underscored by the fact that all the factors considered in existing loss functions can be determined by the coordinates of the

top-left and bottom-right points, encompassing the non-overlapping area, central point distance, and deviations in width and height. As a result, our proposed $MPD - IoU$ metric not only offers a comprehensive consideration of these factors but also streamlines the calculation process. It is worth noting that when the aspect ratio of the predicted bounding box matches that of the ground-truth bounding box, the $MPD - IoU$ metric ensures that the predicted bounding box's value is lower when it is contained within the ground-truth bounding box, compared to cases where the predicted bounding box extends beyond the boundaries of the ground-truth bounding box. This unique characteristic enhances the precision of bounding box regression, ultimately reducing redundancy in the predicted bounding boxes.

4.4. Crack Skeleton Extraction and Measurement

The extraction of crack skeletons serves as a crucial step in deriving geometric properties from segmented crack images, encompassing measurements such as length, width, and area. In this investigation, we employ the medial axis transformation (MAT) method to delineate the outline of the target crack based on binary images [56]. This process involves selecting a point P_t within the interior region (A) of any crack and identifying the point (P_t') on the crack boundary (A^C) that is closest to P_t. If multiple P_t' points are found, P_t is designated as the skeleton point of the crack. The search for P_t' is formalized in Equation (9):

$$d_t\left(P_t, A^C\right) = inf\left\{d(P_t, z) \mid z \in A^C\right\} \tag{9}$$

where inf denotes the lower bound, d_t and d represent Euclidean distances, and z represents any point on A^C. Points with multiple neighbors are excluded from consideration. Subsequently, a distance transformation is executed from all foreground pixels to the background, with "1" representing the foreground and "0" denoting the background. This process is illustrated in Figure 8. Finally, the target is reduced to a skeleton based on the outcomes of the distance transformation. Essentially, this process mirrors the boundary erosion process.

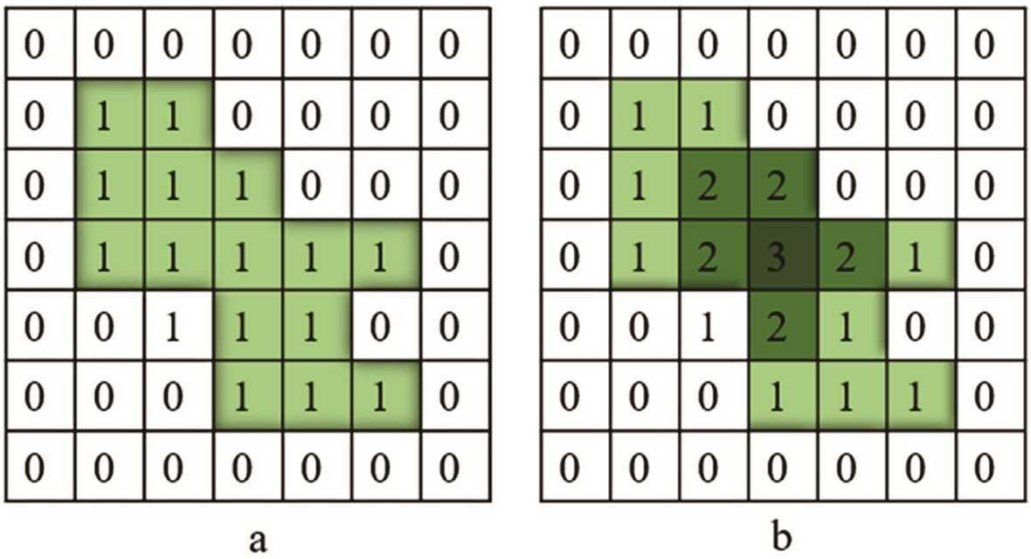


Figure 8. Representation of the incorporated distance transform. (**a**) Before applying distance transform and (**b**) After applying distance transform.

To describe MAT clearly, a diagram is used to explain how the algorithm calculates crack width and length. The coordinate system is shown in Figure 9. Each pixel in the crack image is our coordinate [57]. Therefore, a pixel coordinate system is established with red and yellow line segments representing the boundaries of the crack. The black line segment

in the middle is the crack skeleton, calculated based on the central skeleton algorithm, and $W(x)$ is the calculated average width of the crack. Part is dl. Within the scope of this study, the width of each crack in the workpiece is represented by pixel values. Thus, each pixel represents the width or length, making it easy to calculate the number of pixels contained in the crack. The calculation formula for determining the crack length (L) is shown in Formula (10).

$$L = \int_c f(x,y)dl \cong \sum f(x,y)dl \tag{10}$$

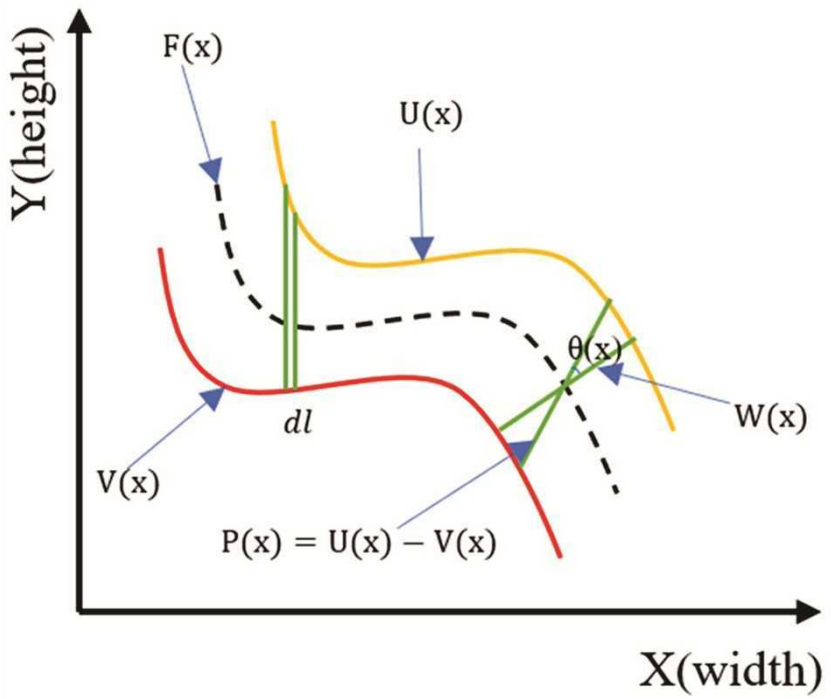


Figure 9. Schematic diagram of calculation method of crack length, width.

The function $f(x,y)$ acts as a geometric calibration index, aligning the detected displacements of points within the image. Simultaneously, dl denotes the finite length of each skeleton unit. c represents the path of the crack skeleton. The crack skeleton is a central line that traces the main path of the crack through the image. This skeleton is derived from the central skeleton algorithm, which identifies the midline of the crack. Integration or summation along the crack skeleton accumulates the lengths of small segments dl corrected by $f(x,y)$.

The pixel-level crack width value is computed by the pixel distance calculation between the crack pixel (1) and the intact pixel (0) using the Euclidean distance transform (EDT) algorithm. The Euclidean distance (d) of the two pixels, P_t and P'_t, is expressed by Equation (11) [58].

$$(P_t, P'_t) = \sqrt{\left(x_{P_t} - x_{P'_t}\right)^2 + \left(y_{P_t} - y_{P'_t}\right)^2}\left(P_t \in A, P'_t \in A^C\right) \tag{11}$$

where t is an indexing variable used to uniquely identify each pixel within the respective sets A and A^C to calculate the Euclidean distance between a cracked pixel and an intact pixel. A is the set of green pixels represented by 1 in Figure 8, and A^C is the white pixel, complement of the green pixel set, A.

4.5. Performance Evaluation

The selection of suitable evaluation metrics holds paramount importance when assessing various segmentation models. While accuracy remains a prominent benchmark for existing instance segmentation models, the advent of real-time and lightweight models has

introduced new considerations for practical deployment on equipment. Consequently, our proposed model shall undergo evaluation with a multifaceted approach, scrutinizing not only its accuracy but also its runtime performance and model complexity to ascertain its suitability for practical applications.

4.5.1. Accuracy

In this study, we conducted a comprehensive evaluation of the enhanced YOLOv8s-Seg model, employing a range of performance metrics, including precision, recall, *F*1-score, and segment mAP at both 50% and 75% intersection over union (IoU) thresholds [59–61]. Our assessment focused on two distinct aspects: the accuracy of crack detection, which was quantified using precision, recall, and *F*1-score, and the quality of the segmentation results, which were appraised through segment mAP.

$$\text{precision} = \frac{TP}{(TP + FP)} \times 100\% \tag{12}$$

$$\text{recall} = \frac{TP}{(TP + FN)} \times 100\% \tag{13}$$

$$F1 = 2 \times \text{precision} \times \frac{\text{recall}}{\text{precision} + \text{recall}} \tag{14}$$

$$seg_{mAP} = \sum_{i=1}^{c} \frac{AP(i)}{C} \tag{15}$$

Here, *TP* represents true positive samples, which are actual positive instances correctly predicted as such. *FP* indicates false positive samples, signifying actual negative instances erroneously identified as positive, while *FN* stands for false negatives, representing actual positive instances wrongly classified as negative. $AP(i)$ corresponds to the average precision for each segmentation category, and *C* represents the total number of segmentation categories. Notably, the segmentation model's performance is directly linked to the cumulative *AP* score, and a higher *AP* score is indicative of improved segmentation quality.

4.5.2. Computational Time

In the realm of model evaluation, two critical indicators to gauge efficiency are training times and inference times. Training time denotes the duration required for a model to converge during the training process, while inference time signifies the time taken to process an image, thereby reflecting the model's suitability for practical industrial applications. To streamline the assessment process, the use of FPS (frames per second) has become a prevalent metric for evaluating the efficiency of instance segmentation methods. This metric offers a straightforward means for researchers to identify and select faster inference models when operating under comparable conditions [62,63].

4.5.3. Model Complexity

Model complexity plays a crucial role in evaluating the practical usability of a model, as it directly influences storage requirements and computational demands. In our study, it specifically refers to the storage needs linked to model parameters, a factor closely tied to the implementation of algorithms on mobile devices.

5. Model Implementation

This section addresses data collection and analysis of bridge surface cracks.

5.1. Data Collection

The model training was conducted on a high-performance computing workstation equipped with specific hardware, including GPUs (GeForce RTX 3090, NVIDIA, Santa Clara, CA, USA), a CPU (Intel(R) Core (TM) i9-10900 CPU @ 2.80 GHz, Intel, Santa Clara, CA,

USA), and 22.6 GB of RAM. The system operated on Windows 10 and utilized PyTorch 2.0.0 as the deep learning framework, with the software environment incorporating CUDA 11.7 and Python 3.8. Detailed specifications of the hardware and software setups are provided in the accompanying table. The data collection and processing methodologies for this study are detailed in Figure 10. The dataset comprises a total of 1600 cases, meticulously annotated for accuracy. Among these, 240 images were sourced from Hong Kong bridges and their associated data enhancements, with the remaining 1360 images obtained from the publicly available database of BSD. The latter subset of 1360 bridge crack images, each sized 224 × 224 pixels, was randomly selected from the BSD dataset. It encompasses images with cracks of variable size and width, stains, shadows, complex backgrounds, and non-uniform illumination. The selected images contain instances of transverse cracks, diagonal cracks, and reticular cracks.

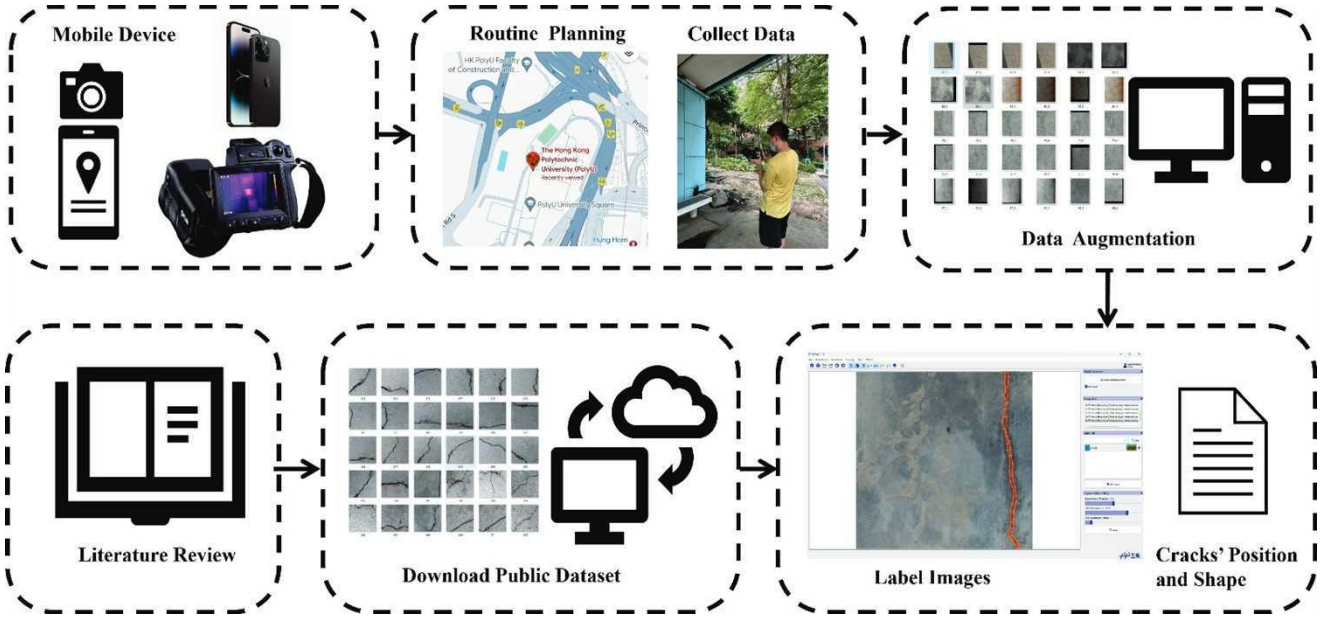


Figure 10. The procedures of crack image capture and preprocessing.

Additionally, 60 crack images were captured from bridges near the Hong Kong Polytechnic University. The images in the Hong Kong bridge crack dataset (HKBCD) were obtained using an iPhone 14 Pro and a FLIR thermal camera capable of producing RGB images. The Hong Kong subset used in this study was collected from two distinct locations: the footbridge surrounding the Hong Kong Polytechnic University campus and the K74 flyover bridge at Pik Wan road, depicted in Figure 11. To enrich the diversity of the dataset, images were captured under a variety of conditions, encompassing different crack shapes, backgrounds, brightness levels, and noise settings. As illustrated in Figure 12, images (a)–(d) showcase four types of cracks: transversal, longitudinal, diagonal, and reticular. Images (e)–(h) depict cracks on different bridge materials, including concrete and asphalt. Images (i)–(l) display bridge cracks under varying light intensities. Images (m)-(p) are influenced by different types of noise, such as graffiti, water stains, moss, and other obstructions. By incorporating these variations, the dataset ensures a comprehensive representation of real-world scenarios, enhancing the robustness and generalizability of the analysis. The augmentation of the HKBCD added an additional 240 crack images, bringing the total compiled dataset to 1600 images. This dataset was then divided into training, validation, and test sets in a 7:1:2 ratio, resulting in 1120 training images, 160 validation images, and 320 test images for the crack segmentation task. Further details on the dataset partitioning are provided in Table 1. Notably, the dataset encompasses various types of concrete cracks, categorized as follows: approximately 20% of the images depict reticular cracks, 40% feature diagonal cracks, and the remaining 40% display linear cracks. Reticular

cracks form a mesh-like pattern and are often attributed to concrete shrinkage or restrained thermal movements. Diagonal cracks appear diagonally across concrete surfaces and are commonly caused by structural overloading or differential settlement. Transverse cracks run perpendicular to the direction of tensile stress and are frequently associated with drying shrinkage, thermal contraction, or excessive applied loads.



(**a**) (**b**)

Figure 11. Sources of the HKBCD dataset. (**a**) Footbridge across the Hong Kong Polytechnic University; (**b**) flyover bridge at Pik Wan road.

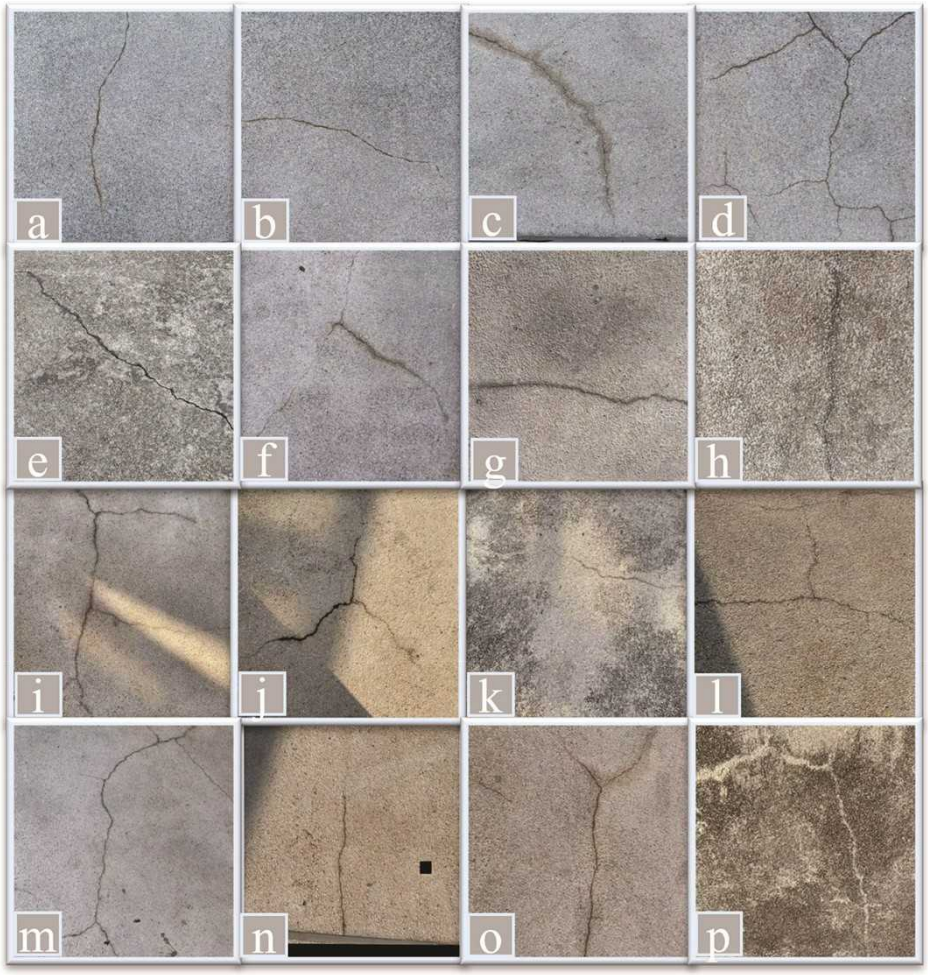


Figure 12. Sample of images from HKBCD datasets. (**a**–**d**) Different Types of cracks, (**e**–**h**) Different types of bridge materials, (**i**–**l**) Different lighting conditions and (**m**–**p**) Different types of noises.

Table 1. Partitioning of bridge crack dataset.

Task	Dataset	Training	Validation	Testing	Total Number
Segmentation	BSD	952	136	272	1360
	ours	168	24	48	240
	New dataset	1120	160	320	1600

Four data augmentation techniques were implemented to bolster the model's generalization capacity, encompassing adjustments in brightness, panning, mirroring, and rotation. It is worth noting that the level of added noise was meticulously regulated to prevent adverse effects on the model's output. Consequently, a total of 240 crack images were generated for the dataset. The EIseg software 1.1.1 was employed to annotate this dataset. The results of the data augmentation techniques are presented in Figure 13, illustrating four methods: brightness, panning, mirroring, and rotation. The developed YOLOv8-AFPN-MPD-IoU model underwent training using the following hyperparameters: a learning rate of 0.01, momentum of 0.937, weight decay of 0.0005, and a batch size of 8. A stochastic gradient descent (SGD) optimizer was utilized for training the model over 300 epochs in total. Hardware and software requirements for this research study are detailed in Table 2.

Table 2. Hardware and software specifications.

Hardware	Information	Software	Information
GPUs	GeForce RTX 3090	Operating System	Windows 10 (Version 23H2)
CPU	Intel(R) Core (TM) i9-10900 CPU @ 2.80 GHz	Deep Learning Framework	PyTorch
RAM	22.6 GB	CUDA Version	11.7
		Python Version	3.8



Figure 13. Samples of data augmentation, adjustments in brightness, panning, mirroring, and rotation.

5.2. Performance Comparison of YOLOv8 Architectures

In this research study, we conducted a series of experiments employing YOLOv8-Seg models at various scales with the objective of evaluating segment mAP50 and model size. Given the limited occurrence of cracks within each image, the initial four scales of models were examined to determine the most suitable scale. Similar to YOLOv5, YOLOv8 incorporates an early stopping mechanism to prevent model overfitting, whereby training is terminated if no significant improvement is observed over a specified number of epochs. The "patience" parameter is indicative of the number of epochs for which no

discernible progress is tolerated before training halts. In our case, "patience" was set at 50 epochs, and the "fitness" criterion was defined as a weighted combination of metrics, with (0.1 × mAP50 + 0.9 × mAP50-95) utilized to assess the performance of the CNNs model. Following 300 epochs of training, YOLOv8s exhibited its best performance when the model depth was a multiple of 0.33, width was a multiple of 0.5, and the maximum number of channels was set to 1024. The results of these experiments are presented in Table 3, with YOLOv8s outperforming the other three models, achieving the highest mAP50 at 74% and the highest fitness score at 37.64%. Consequently, YOLOv8s was selected as the baseline model for further analysis.

Table 3. Performance comparison of YOLOv8 architectures based on test dataset.

Model	Epoch	Size	Running Time	P	R	mAP50	mAP75	Map50-95	F1	Model Fitness
YOLOv8n	300	6.43 M	4.028 h	88.80%	71.10%	72.60%	71.60%	32.90%	78.97%	0.3687
YOLOv8s	300	22.7 M	3.385 h	92.60%	68.60%	74.00%	73.40%	33.60%	78.81%	0.3764
YOLOv8m	300	67.8 M	3.783 h	90.20%	68.10%	71.30%	70.40%	32.40%	77.61%	0.3629
YOLOv8l	300	151 M	6.164 h	88.90%	70.20%	73.70%	73.20%	33.50%	78.45%	0.3752

5.3. Performance Comparison of Different Loss Functions

To assess the superiority of the MPD-IoU loss function in the recognition of the entire crack dataset, we conducted a comparative analysis involving three different loss functions: C-IoU, W-IoU, and MPD-IoU, all within the YOLOv8s framework. These experiments entailed an examination of three distinct bounding box regression loss functions, while maintaining identical network structures across all three schemes, with the sole variation being the choice of loss function. Furthermore, all three schemes were trained using the same parameters. In the context of image segmentation, "recall" signifies the model's capability to accurately detect target objects, representing the proportion of pixels effectively identified by the model as true targets. Conversely, "precision" quantifies the proportion of pixels within the model's segmentation output that are accurately classified as target objects. The relationship between recall and precision in image segmentation often involves a trade-off, necessitating the introduction of the F1-score to strike a balance between these two metrics. Enhancing the model's sensitivity to target objects (improving recall) may lead to the inclusion of more target regions in the segmentation results but can also result in an increase in false positives. Hence, reducing false positives (enhancing precision) may lower recall as the model adopts a more conservative approach, becoming less inclined to classify boundary regions as target objects.

Table 4 reports a performance comparative analysis based on the test dataset. Notably, in the specific context of bridge crack detection, the ramifications of false negatives can be exceedingly detrimental, potentially causing significant financial losses and even posing risks to human safety. Therefore, when their F1-scores are approximately equal, prioritizing recall over precision is advisable during bridge inspection. As a result, the "YOLOv8s + AFPN + MPDIoU" combination emerges as the preferred choice. Furthermore, it exhibits improvements of 1.3% in terms of mAP50 and mAP75 when compared to "YOLOv8s + AFPN + CIOU" on the test dataset, as well as enhancements of 0.3% and 0.8% in mAP50 and mAP75 compared to "YOLOv8s + AFPN + WIOU" on the same test dataset.

Table 4. Performance comparison of loss functions based on testing dataset.

Model	Epoch	Model Size	Running Time	P	R	mAP50	mAP75	F1
YOLOv8s + AFPN + CIoU	300	23.7 MB	3.596 h	0.935	0.694	0.74	0.735	79.69%
YOLOv8s + AFPN + MPDIoU	300	22.7 MB	3.661 h	0.907	0.704	0.753	0.748	79.27%
YOLOv8s + AFPN + WIoU	300	23.7 MB	3.575 h	0.936	0.677	0.75	0.74	78.57%

5.4. Performance Comparison of Instance Segmentation Models

To assess the effectiveness of the proposed YOLOv8s + AFPN + MPDIoU model, we conducted a comprehensive comparative analysis alongside three established instance

segmentation models, with a focus on key performance metrics including precision, recall, F1-score, mAP50, and mAP75. Results of the conducted performance comparative analysis are given in Table 5. They highlight the superiority of the YOLOv8s + AFPN + MPDIoU model, which achieved a precision rate of 90.7%, a recall of 70.4%, an F1-score of 79.27%, mAP50 of 75.3%, and mAP75 of 74.80%. In comparison to the YOLOv8s model, our proposed model exhibited a minor decrease in precision but demonstrated improvements in other performance indicators, with recall, F1-score, mAP50, and mAP75 increasing by 1.8%, 0.46%, 1.3%, and 1.4%, respectively. Regarding model complexity, the proposed model has a relatively compact size, requiring only 22.7 M of computer storage, making it larger only than the YOLOv5s-seg model. This compact size facilitates the installation of the model on portable equipment for practical bridge inspections. Notably, Mask-RCNN, being a two-stage instance segmentation model with the most parameters in comparison to other models in this study, consumes the highest amount of computer storage at 335 M. Additionally, for the YOLOv8-seg model, an increase in channel width and depth corresponds to an escalation in model size, with YOLOv8s-seg at 22.7 M, YOLOv8m-seg at 67.8 M, and YOLOv8l-seg at 151 M. Moreover, the model size also influences training and inference times. The most complex model, Mask-RCNN, requires the longest training time (approximately 11.5 h) and inference time (36 ms per image). In contrast, our model, YOLOv8s + AFPN + MPDIoU, takes the second place in terms of model complexity and requires 12.2 ms to recognize bridge cracks in an image. Consequently, YOLOv8s + AFPN + MPDIoU emerges as the optimal choice, effectively balancing precision, time efficiency, and ease of installation.

Table 5. Performance comparison of instance segmentation models based on testing dataset.

Model	Epoch	Model Size	Inference Time	Running Time	MASK				
					P	R	F1	mAP50	mAP75
YOLOv5s-seg	282	18.79 M	11.5 ms	2.397 h	90.50%	68.80%	78.17%	72.30%	70.60%
YOLOv8s-seg	300	22.7 M	8.9 ms	3.385 h	92.60%	68.60%	78.81%	74.00%	73.40%
YOLOv8m-seg	300	67.8 M	12.5 ms	3.783 h	90.20%	68.10%	77.61%	71.30%	70.40%
YOLOv8l-seg	300	151 M	12.5 ms	6.164 h	88.90%	70.20%	78.45%	73.70%	73.20%
Mask-RCNN	300	335 M	36 ms	11.5 h	49.82%	26.10%	34.25%	49.80%	7.60%
YOLOv8s + AFPN + MPDIoU (ours)	300	22.7 MB	12.2 ms	3.661 h	90.70%	70.40%	79.27%	75.30%	74.80%

5.5. Crack Measurement Performance

In the final section, the proposed model effectively segmented various types of cracks in diverse structural and environmental contexts (see Figure 14). To further augment its functionality, we introduced a width measurement module designed to quantitatively assess the segmentation outcomes. While the geometric characteristics of cracks are typically described in pixel terms, determining the actual width of a crack necessitates the conversion of pixel measurements into internationally recognized standard units of measurement. Furthermore, potential correction factors, such as accounting for the angle between the camera and the photographed surface, must be taken into consideration to ensure precise width measurements. These considerations will be a focal point in our future research endeavors, aimed at refining and enhancing the model's measurement capabilities. As observed in Table 6, it is clear that the discrepancy between the actual values and the segmentation results is quite minimal for the majority of diagonal and transverse cracks. However, owing to their intricate nature, the model frequently generates misidentifications and incorrect predictions when dealing with reticular bridges. This highlights the imperative need for further refinement in enhancing the model's accuracy in the detection and prediction of reticular cracks within bridge structures. Tackling this specific challenge will be a primary focus of our forthcoming research and development endeavors, with the ultimate aim of bolstering the model's capabilities and ensuring precise segmentation outcomes across all types of cracks, even within complex structures such as reticular bridges.

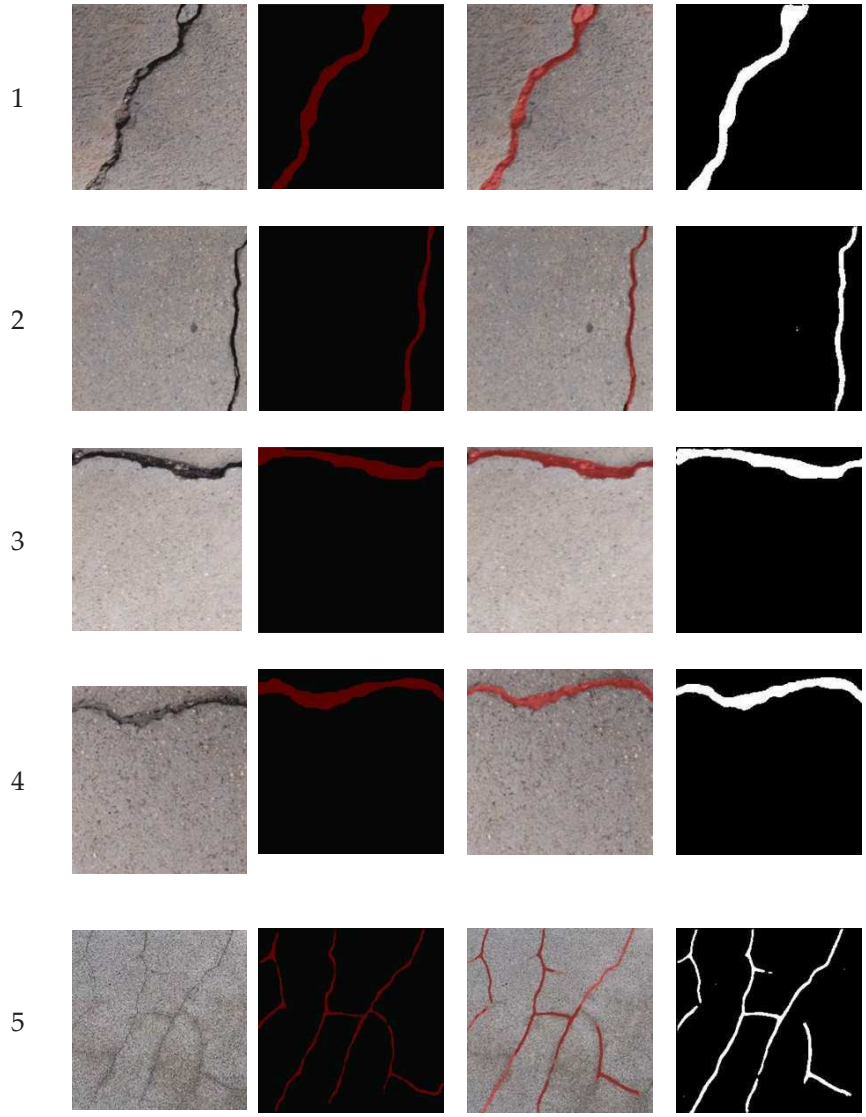


Figure 14. Samples of the crack measurement. (**a**) Original crack images; (**b**) manually segmented crack images; (**c**) segmented crack images with original background; and (**d**) segmented crack images. 1: Diagonal crack, 2–4: Linear cracks and 5: eticular crack.

Table 6. Comparison of manually measured and segmentation.

		Ground Truth		Segmentation		Error	
Image Number	**Type**	**Max Length**	**Max Width**	**Max Length**	**Max Width**	**Max Length**	**Max Width**
1	Diagonal	254.00	23.75	259.00	24.83	−1.97%	−4.56%
2	Diagonal	225.00	7.64	229.00	7.64	−1.78%	0.00%
3	Transverse	230.00	21.08	225.00	21.01	2.17%	0.33%
4	Transverse	227.00	19.93	228.00	18.02	−0.44%	5.31%
5	Reticular	4568.00	40.11	4513.00	39.03	1.20%	2.70%

Figure 15 expounds another portion of segmented images by the developed YOLOv8-AFPN-MPD-IoU model. Through comprehensive analysis of the results from the test dataset, the proposed model demonstrated robust generalization capabilities across various environmental conditions. Specifically, the model effectively identified cracks under different lighting scenarios (e.g., low light, direct sunlight) and weather conditions (e.g., cloudy day, clear skies), as illustrated in Figure 15a–c. Moreover, the model exhibited

strong anti-interference capabilities, accurately distinguishing cracks from external objects such as water stains, graffiti, and shadows, as shown in Figure 15d–f. The model performed well in detecting single horizontal and vertical cracks, as well as simple branched cracks (Figure 15g–i). Overall, these results indicate that the model can generalize well across various environmental conditions, enhancing its applicability in real-world scenarios, where such variability is common.

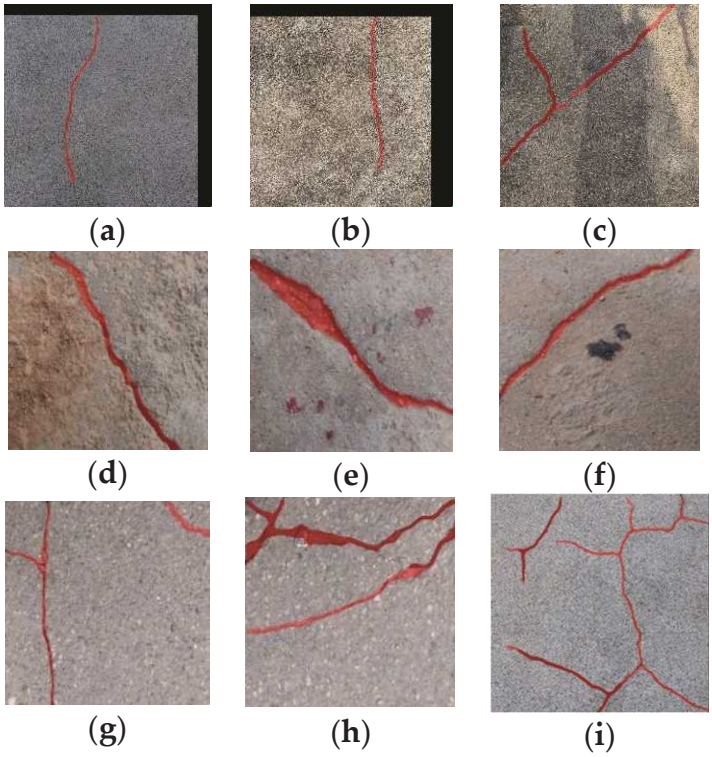


Figure 15. Samples of the segmented images with (**a**–**c**) different lighting conditions, (**d**–**f**) stains, and (**g**–**i**) reticular cracks.

6. Conclusions

This study introduced a novel computer vision model aimed at effectively segmenting bridge cracks and deriving precise geographic information from the segmented crack images at the pixel level. The traditional method of human visual inspection is not only time-consuming and subjective but also poses risks such as falls and injuries. Although existing object detection models have mitigated these issues, they primarily rely on qualitative analyses for crack evaluation. In contrast, our proposed model offers a groundbreaking quantitative analysis, empowering bridge inspectors to make informed decisions. The superiority of the developed YOLOv8s + AFPN + MPDIoU model lies in the following:

1. Balanced accuracy and speed: By employing a one-stage instance model, we strike a crucial balance between segmentation accuracy and processing speed. This approach avoids the limitations of two-stage methods that may not comprehensively capture the interplay between detection and segmentation. Multi-stage methods, often associated with extended processing times, are circumvented, enabling real-time segmentation.
2. Innovative feature fusion: The incorporation of an asymptotic feature pyramid network in the YOLOv8-seg model replaces conventional top-down or bottom-up feature pyramid networks. This innovative choice addresses the loss and deterioration of feature information between non-adjacent layers, facilitating improved feature fusion and enhancing the overall model performance.
3. Specialized loss function: The introduction of the minimum point distance-IoU loss function tackles issues arising when the predicted bounding box possesses the same

aspect ratio as the ground-truth bounding box but exhibits distinct width and height values. This tailored loss function ensures a more accurate and reliable model.

4. Quantitative measurement method: The combination of the middle aisle transformation method and Euclidean distance method to calculate the length and width of bridge cracks in segmented images provides a quantitative basis for maintenance suggestions. This method enhances the precision of crack assessment.

Furthermore, in terms of the loss function selection, the MPD-IoU loss function emerges as the optimal choice, achieving the highest recall score. Given that every crack poses a potential threat to bridge health, this selection is crucial. Comparative analysis with other instance models reveals that the YOLOv8s + AFPN + MPDIoU model outperforms others in terms of model size, inference time, and mAP50, making it the optimal choice for practical industry applications. Additionally, the evaluation of max width and length errors between manual inspection and segmented crack results indicates a minimal discrepancy of around 5%, significantly smaller than the errors associated with visual inspection. This underscores the reliability and precision of our proposed model in assessing bridge cracks. In summary, our computer vision model stands out as an innovative, efficient, and reliable solution for bridge inspection and maintenance in the industry.

Author Contributions: Conceptualization, C.X., X.J., T.Z., G.A. and E.M.A.; methodology, C.X., X.J., T.Z., G.A. and E.M.A.; validation, C.X., X.J., T.Z. and E.M.A.; formal analysis, C.X., X.J., T.Z. and E.M.A.; investigation, C.X., X.J., T.Z., G.A. and E.M.A.; resources, C.X. and E.M.A.; data curation, C.X. and E.M.A.; writing—original draft preparation, C.X., X.J., T.Z., G.A. and E.M.A.; writing—review and editing C.X., X.J., T.Z., G.A. and E.M.A.; visualization, C.X., X.J., T.Z., G.A. and E.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge the support from the Smart Traffic Fund (STF) under grant number PSRI/14/2109/RA.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Some or all the data that support the findings of this research study are available upon request from the corresponding author.

Acknowledgments: The authors express their appreciation to the Researchers Supporting Project number (RSPD2024R899), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Leung, A.; Tanko, M.; Burke, M.; Shui, C.S. Bridges, tunnels, and ferries: Connectivity, transport, and the future of Hong Kong's outlying islands. *Isl. Stud. J.* **2017**, *12*, 61–82. [CrossRef]
2. Weng, X.; Huang, Y.; Wang, W. Segment-based pavement crack quantification. *Autom. Constr.* **2019**, *105*, 102819. [CrossRef]
3. Song, L.; Sun, H.; Liu, J.; Yu, Z.; Cui, C. Automatic segmentation and quantification of global cracks in concrete structures based on deep learning. *Measurement* **2022**, *199*, 111550. [CrossRef]
4. Adhikari, R.S.; Moselhi, O.; Bagchi, A. Image-based retrieval of concrete crack properties for bridge inspection. *Autom. Constr.* **2014**, *39*, 180–194. [CrossRef]
5. Kim, B.; Cho, S. Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. *Sensors* **2018**, *18*, 3452. [CrossRef] [PubMed]
6. AASHTO. *Manual for Maintenance Inspection of Bridges*; AASHTO: Washington, DC, USA, 1983.
7. Sterritt, G. *Review of Bridge Inspection Competence and Training*; Project Report; UK Bridges Board: London, UK, 2009.
8. *JTG/T H21-2011*; Standards for Technical Condition Evaluation of Highway Bridges. China's Ministry of Transportation: Beijing, China, 2011.
9. Park, S.E.; Eem, S.H.; Jeon, H. Concrete crack detection and quantification using deep learning and structured light. *Constr. Build. Mater.* **2020**, *252*, 119096. [CrossRef]
10. Ji, A.; Xue, X.; Wang, Y.; Luo, X.; Xue, W. An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement. *Autom. Constr.* **2020**, *114*, 103176. [CrossRef]
11. Yu, L.; He, S.; Liu, X.; Jiang, S.; Xiang, S. Intelligent Crack Detection and Quantification in the Concrete Bridge: A Deep Learning-Assisted Image Processing Approach. *Adv. Civ. Eng.* **2022**, *2022*, 1813821. [CrossRef]

12. Jang, K.; An, Y.K.; Kim, B.; Cho, S. Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot. *Comput.-Aided Civ. Infrastruct. Eng.* **2021**, *36*, 14–29. [CrossRef]
13. Peng, X.; Zhong, X.; Zhao, C.; Chen, A.; Zhang, T. A UAV-based machine vision method for bridge crack recognition and width quantification through hybrid feature learning. *Constr. Build. Mater.* **2021**, *299*, 123896. [CrossRef]
14. Deng, J.; Singh, A.; Zhou, Y.; Lu, Y.; Lee VC, S. Review on computer vision-based crack detection and quantification methodologies for civil structures. *Constr. Build. Mater.* **2022**, *356*, 129238. [CrossRef]
15. Hamishebahar, Y.; Guan, H.; So, S.; Jo, J. A Comprehensive Review of Deep Learning-Based Crack Detection Approaches. *Appl. Sci.* **2022**, *12*, 1374. [CrossRef]
16. Deng, L.; Zhang, A.; Guo, J.; Liu, Y. An Integrated Method for Road Crack Segmentation and Surface Feature Quantification under Complex Backgrounds. *Remote Sens.* **2023**, *15*, 1530. [CrossRef]
17. Zhu, J.; Zhong, J.; Ma, T.; Huang, X.; Zhang, W.; Zhou, Y. Pavement distress detection using convolutional neural networks with images captured via UAV. *Autom. Constr.* **2022**, *133*, 103991. [CrossRef]
18. Guo, J.; Liu, P.; Xiao, B.; Deng, L.; Wang, Q. Surface defect detection of civil structures using images: Review from data perspective. *Autom. Constr.* **2024**, *158*, 105186. [CrossRef]
19. Chen, K.; Fu, K.; Yan, M.; Gao, X.; Sun, X.; Wei, X. Semantic Segmentation of Aerial Images with Shuffling Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 173–177. [CrossRef]
20. Jiang, H.; Hu, X.; Li, K.; Zhang, J.; Gong, J.; Zhang, M. PGA-SiamNet: Pyramid feature-based attention-guided siamese network for remote sensing orthoimagery building change detection. *Remote Sens.* **2020**, *12*, 484. [CrossRef]
21. Zhan, Y.; Fu, K.; Yan, M.; Sun, X.; Wang, H.; Qiu, X. Change Detection Based on Deep Siamese Convolutional Network for Optical Aerial Images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1845–1849. [CrossRef]
22. Tong, X.; Guo, J.; Ling, Y.; Yin, Z. A New Image-Based Method for Concrete Bridge Bottom Crack Detection. In Proceedings of the 2011 International Conference on Image Analysis and Signal Processing, Wuhan, China, 21–23 October 2011; pp. 1–4. [CrossRef]
23. Chen, B.; Zhang, X.; Wang, R.; Li, Z.; Deng, W. Detect concrete cracks based on OTSU algorithm with differential image. *J. Eng.* **2019**, *2019*, 9088–9091. [CrossRef]
24. Hoang, N.D. Detection of Surface Crack in Building Structures Using Image Processing Technique with an Improved Otsu Method for Image Thresholding. *Adv. Civ. Eng.* **2018**, *2018*, 3924120. [CrossRef]
25. Vivekananthan, V.; Vignesh, R.; Vasanthaseelan, S.; Joel, E.; Kumar, K.S. Concrete bridge crack detection by image processing technique by using the improved OTSU method. *Mater. Today Proc.* **2023**, *74*, 1002–1007. [CrossRef]
26. Wang, Y.; Zhang, J.Y.; Liu, J.X.; Zhang, Y.; Chen, Z.P.; Li, C.G.; He, K.; Yan, R.B. Research on Crack Detection Algorithm of the Concrete Bridge Based on Image Processing. *Procedia Comput. Sci.* **2018**, *154*, 610–616. [CrossRef]
27. Abdel-Qader, I.; Abudayyeh, O.; Kelly, M.E. Analysis of Edge-Detection Techniques for Crack Identification in Bridges. *J. Comput. Civ. Eng.* **2003**, *17*, 255–263. [CrossRef]
28. Xu, X.J.; Zhang, X.N. Crack detection of reinforced concrete bridge using video image. *J. Cent. South Univ.* **2013**, *20*, 2605–2613. [CrossRef]
29. Zhang, L.; Luo, W.; Xu, Y. Bridge Crack Image Segmentation Based on Improved Watershed Algorithm. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 3537–3541. [CrossRef]
30. Mirbod, M.; Shoar, M. Intelligent Concrete Surface Cracks Detection using Computer Vision, Pattern Recognition, and Artificial Neural Networks. *Procedia Comput. Sci.* **2022**, *217*, 52–61. [CrossRef]
31. Moon, H.-G.; Kim, J.-H. Intelligent Crack Detecting Algorithm on The Concrete Crack Image Using Neural Network. In Proceedings of the 28th International Symposium on Automation and Robotics in Construction, Seoul, Republic of Korea, 29 June–2 July 2011; pp. 1–7. [CrossRef]
32. Peng, J.; Zhang, S.; Peng, D.; Liang, K. Research on Bridge Crack Detection with Neural Network Based Image Processing Methods. In Proceedings of the 12th International Conference on Reliability, Maintainability, and Safety, ICRMS, Shanghai, China, 17–19 October 2018; pp. 419–428. [CrossRef]
33. Liang, S.; Jianchun, X.; Xun, Z. An extraction and classification algorithm for concrete cracks based on machine vision. *IEEE Access* **2018**, *6*, 45051–45061. [CrossRef]
34. Prasanna, P.; Dana, K.; Gucunski, N.; Basily, B. Computer Vision Based Crack Detection and Analysis. In *Proceedings Volume 8345, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*; SPIE: Bellingham, WC, USA, 2012; pp. 1–6. [CrossRef]
35. Yao, C.; Tao, M.; Xiaojie, W.; Feng, L.I. A Bridge Crack Image Detection and Classification Method Based on Climbing Robot. In Proceedings of the 35th Chinese Control Conference, Chengdu, China, 27–29 July 2016; pp. 4037–4042. [CrossRef]
36. Li, Z.; Huang, M.; Ji, P.; Zhu, H.; Zhang, Q. One-step deep learning-based method for pixel-level detection of fine cracks in steel girder images. *Smart Struct. Syst.* **2022**, *29*, 153–166. [CrossRef]
37. Zhang, J.; Qian, S.; Tan, C. Automated bridge surface crack detection and segmentation using computer vision-based deep learning model. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105225. [CrossRef]
38. Rosso, M.M.; Marasco, G.; Aiello, S.; Aloisio, A.; Chiaia, B.; Marano, G.C. Convolutional networks and transformers for intelligent road tunnel investigations. *Comput. Struct.* **2023**, *275*, 106918. [CrossRef]
39. Park, M.J.; Kim, J.; Jeong, S.; Jang, A.; Bae, J.; Ju, Y.K. Machine learning-based concrete crack depth prediction using thermal images taken under daylight conditions. *Remote Sens.* **2022**, *14*, 2151. [CrossRef]

40. Zhang, C.; Chang, C.-c.; Jamshidi, M. Concrete bridge surface damage detection using a single-stage detector. *Comput.-Aided Civ. Infrastruct. Eng.* **2020**, *35*, 389–409. [CrossRef]
41. Deng, J.; Lu, Y.; Lee VC, S. Imaging-based crack detection on concrete surfaces using You Only Look Once network. *Struct. Health Monit.* **2021**, *20*, 484–499. [CrossRef]
42. Teng, S.; Liu, Z.; Chen, G.; Cheng, L. Concrete crack detection based on well-known feature extractor model and the YOLO_v2 network. *Appl. Sci.* **2021**, *11*, 813. [CrossRef]
43. Qiu, Q.; Lau, D. Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images. *Autom. Constr.* **2023**, *147*, 104745. [CrossRef]
44. Xiang, X.; Hu, H.; Ding, Y.; Zheng, Y.; Wu, S. GC-YOLOv5s: A Lightweight Detector for UAV Road Crack Detection. *Appl. Sci.* **2023**, *13*, 11030. [CrossRef]
45. Choi, W.; Cha, Y.J. SDDNet: Real-Time Crack Segmentation. *IEEE Trans. Ind. Electron.* **2020**, *67*, 8016–8025. [CrossRef]
46. Zhang, X.; Rajan, D.; Story, B. Concrete crack detection using context-aware deep semantic segmentation network. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 951–971. [CrossRef]
47. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
48. Xu, H.; Su, X.; Wang, Y.; Cai, H.; Cui, K.; Chen, X. Automatic bridge crack detection using a convolutional neural network. *Appl. Sci.* **2019**, *9*, 2867. [CrossRef]
49. Yue, X.; Qi, K.; Na, X.; Zhang, Y.; Liu, Y.; Liu, C. Improved YOLOv8-Seg Network for Instance Segmentation of Healthy and Diseased Tomato Plants in the Growth Stage. *Agriculture* **2023**, *13*, 1643. [CrossRef]
50. Wu, Y.; Han, Q.; Jin, Q.; Li, J.; Zhang, Y. LCA-YOLOv8-Seg: An Improved Lightweight YOLOv8-Seg for Real-Time Pixel-Level Crack Detection of Dams and Bridges. *Appl. Sci.* **2023**, *13*, 10583. [CrossRef]
51. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-aligned One-stage Object Detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 10–17 October 2021; pp. 3490–3499. [CrossRef]
52. Bolya, D.; Fanyi, C.Z.; Yong, X.; Lee, J. YOLACT Real-time Instance Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9157–9166. [CrossRef]
53. Yang, G.; Lei, J.; Zhu, Z.; Cheng, S.; Feng, Z.; Liang, R. AFPN: Asymptotic Feature Pyramid Network for Object Detection. *arXiv* **2023**, arXiv:2306.15988.
54. Liu, S.; Huang, D.; Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. *arXiv* **2019**, arXiv:1911.09516.
55. Siliang, M.; Yong, X. MPDIoU: A Loss for Efficient and Accurate Bounding Box Regression. *arXiv* **2023**, arXiv:2307.07662.
56. Yu, Z.; Shen, Y.; Sun, Z.; Chen, J.; Gang, W. Cracklab: A high-precision and efficient concrete crack segmentation and quantification network. *Dev. Built Environ.* **2022**, *12*, 100088. [CrossRef]
57. Zheng, X.; Zhang, S.; Li, X.; Li, G.; Li, X. Lightweight bridge crack detection method based on segnet and bottleneck depth-separable convolution with residuals. *IEEE Access* **2021**, *9*, 161649–161668. [CrossRef]
58. Bae, H.; An, Y.-K. Computer vision-based statistical crack quantification for concrete structures. *Measurement* **2023**, *211*, 112632. [CrossRef]
59. Nabizadeh, E.; Parghi, A. Automated corrosion detection using deep learning and computer vision. *Asian J. Civ. Eng.* **2023**, *24*, 2911–2923. [CrossRef]
60. Zhou, Y.C.; Hu, Z.Z.; Yan, K.X.; Lin, J.R. Deep learning-based instance segmentation for indoor fire load recognition. *IEEE Access* **2021**, *9*, 148771–148782. [CrossRef]
61. Bai, R.; Wang, M.; Zhang, Z.; Lu, J.; Shen, F. Automated construction site monitoring based on improved YOLOv8-seg instance segmentation algorithm. *IEEE Access* **2023**, *11*, 139082–139096. [CrossRef]
62. Jung, S.; Heo, H.; Park, S.; Jung, S.U.; Lee, K. Benchmarking deep learning models for instance segmentation. *Appl. Sci.* **2022**, *12*, 8856. [CrossRef]
63. Pena-Caballero, C.; Kim, D.; Gonzalez, A.; Castellanos, O.; Cantu, A.; Ho, J. Real-time road hazard information system. *Infrastructures* **2020**, *5*, 75. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

mathematics

MDPI

Article

A New Strategy: Remaining Useful Life Prediction of Wind Power Bearings Based on Deep Learning under Data Missing Conditions

Xuejun Li [1], Xu Lei [1], Lingli Jiang [1,*], Tongguang Yang [2,*] and Zhenyu Ge [1]

[1] The Laboratory for Rotating Vibration Monitoring and Diagnostics Technology in Mechanical Industries, Foshan University, Foshan 528000, China; hnkjdxlxj@163.com (X.L.); 2112202035@stu.fosu.edu.cn (X.L.); 2112202028@stu.fosu.edu.cn (Z.G.)

[2] School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110819, China

* Correspondence: linlyjiang@163.com (L.J.); 2190001@stu.neu.edu.cn (T.Y.)

Abstract: With its formidable nonlinear mapping capabilities, deep learning has been widely applied in bearing remaining useful life (RUL) prediction. Given that equipment in actual work is subject to numerous disturbances, the collected data tends to exhibit random missing values. Furthermore, due to the dynamic nature of wind turbine environments, LSTM models relying on manually set parameters exhibit certain limitations. Considering these factors can lead to issues with the accuracy of predictive models when forecasting the remaining useful life (RUL) of wind turbine bearings. In light of this issue, a novel strategy for predicting the remaining life of wind turbine bearings under data scarcity conditions is proposed. Firstly, the average similarity (AS) is introduced to reconstruct the discriminator of the Generative Adversarial Imputation Nets (GAIN), and the adversarial process between the generative module and the discriminant is strengthened. Based on this, the dung beetle algorithm (DBO) is used to optimize multiple parameters of the long-term and short-term memory network (LSTM), and the complete data after filling is used as the input data of the optimized LSTM to realize the prediction of the remaining life of the wind power bearing. The effectiveness of the proposed method is verified by the full-life data test of bearings. The results show that, under the condition of missing data, the new strategy of AS-GAIN-LSTM is used to predict the RUL of wind turbine bearings, which has a more stable prediction performance.

Keywords: wind power bearing; remaining useful life prediction; data missing; average similarity

MSC: 68T07

1. Introduction

Wind power is a crucial area of development in the modern world, with wind turbines serving as the primary means of harnessing this renewable energy source [1]. Among the core components of wind turbines, wind turbine bearings play a pivotal role in their operation. However, they are vulnerable to various failure modes due to environmental uncertainties and other factors, which can negatively impact the stability and safety of the turbine equipment. In severe cases, this may result in irreparable financial losses and personnel casualties [2]. Therefore, there is an urgent need and significant importance in conducting predictive maintenance research based on real-time wind turbine bearing monitoring data.

Currently, there are three types of methods for predicting the remaining life of rolling bearings: those based on failure mechanism models [3,4], those based on statistical models [5,6], and those based on machine-learning techniques [7–9]. Among these, methods based on failure mechanism models may have difficulty describing the damage mechanisms of complex mechanical systems. Statistical models may not be able to capture

nonlinear information, which makes it challenging to match the current complex and variable operating conditions. Most machine-learning approaches cannot deeply analyze and process state signals [10].

Due to its adaptability to diverse and complex scenarios, deep learning [11] has been extensively applied in the field of predicting the remaining life of bearings. Based on a Recursive Neural Network (RNN), Guo et al. [12] constructed a comprehensive health index RNN-HI considering the monotonicity and correlation of features, which solved the problem that the fault threshold was challenging to determine. They completed the RUL prediction of wind turbine generator bearings through a particle filter algorithm. Chen et al. [13] proposed a model based on the combination of RNN and autoencoder that realized the intelligent extraction of features, the construction of a complete end-to-end model of health indicators, and life prediction. Xiang et al. [14] proposed a lightweight regression operator based on RNN to balance the accuracy and simplicity of the model. Finally, the RUL prediction of aero-engine bearings was realized by using the deep learning framework model based on RNN.

Considering the correlation between bearing degradation process and time, Long Short-Term Memory (LSTM) has been favored by researchers because of its powerful time series processing ability [15]. Li et al. [16] used the Gorilla Troops Optimizer (GTO) to improve the prediction performance of the combined model of the Convolutional Neural Network (CNN) and LSTM. The vibration signal of the bearing was transformed into a frequency domain amplitude signal. The CNN was used to automatically extract feature information, and the deep features were introduced into the LSTM to predict the bearing RUL. Combining the advantages of LSTM and statistical process analysis, Liu et al. [17] proposed a LSS prediction model. Firstly, the time characteristics of aero-engine bearing vibration signals are extracted by statistical process analysis. Then, the multi-stage signals are input into LSS to complete the prediction of RUL. In order to enhance the model's ability to learn features, L. Magadá et al. [18] proposed a combined model of Stacked Variational Denoising Autoencoder (SVDAE) to construct health indicators and Bidirectional Long Short-Term Memory (Bi-LSTM) for bearing RUL prediction. The experimental results show that the model has relatively stable performance. The parameters of LSTM typically consist of those in the input layer, hidden layer, output layer, and memory cells. These parameters play a crucial role in the training process of deep learning models. By adjusting these parameters, one can influence the model's performance, prediction accuracy, and generalization capacity. Li et al. [19] proposed a predictive model based on Bayesian optimization (BO), Spatial-Temporal Attention (STA), and LSTM. This model adjusts the parameters of the LSTM through Bayesian optimization and applies it to building energy consumption prediction. Results showed that Bayesian optimization alone improved the model's performance by 0.0717. Li et al. [20] used the Peafowl Optimization Algorithm (POA) to optimize the parameters of LSTM and built a POA-LSTM regression prediction model. Through experimental verification, the optimized model demonstrated higher accuracy.

At present, the construction of RUL prediction models is mostly based on the assumption that data acquisition is normal and that data is continuous and uniform. However, at the data acquisition site, due to operator errors, sensor failures, etc. [21], data loss or an uneven data interval may occur in the same equipment data acquisition cycle. Scholars have proposed some feasible data filling methods in terms of missing data. Rădulescu et al. [22] proposed the mean imputation model, which is a statistical method used for data interpolation. For numerical type data, the missing part of the column is interpolated by the average of all the existing values in the missing attribute column, so that the missing sequence becomes a complete sequence. Yang et al. [23] proposed a regression interpolation method. Firstly, a regression model is constructed based on the degree of correlation between different attribute elements. Then, the specific parameters of the constructed model are obtained according to the complete sample records in the data set. Finally, the missing values are filled by the model. Han et al. [24] used weighted K-Nearest Neighbor (KNN)

to obtain globally similar instances and capture fluctuations in local regions, completed the filling of missing values under the two-cycle mechanism related to the ranking of missing variables and missing instances, and obtained better filling results on the sewage treatment plant dataset. Liu et al. [25] proposed an incomplete data filling method based on perceptual CNN, which was successfully applied to the prediction of power generation and improved the prediction performance. Wind turbines are usually installed in areas with large amounts of dust, severe sunshine, or high humidity. The working environment is very harsh, and the load range, temperature, and humidity are large [26], which makes it more prone to data loss. However, the above missing value filling method still needs to be improved when dealing with a complex data with high missing rate.

Based on this, a residual life prediction method for wind power bearing based on deep learning is proposed under the condition that data is missing. Aiming at the problem of missing data, in order to describe the correlation before and after missing as much as possible and improve the reliability of the filled values, an improved filling model based on Generative Adversarial Imputation Nets (GAIN) [27] is proposed. This model fills the missing data according to the characteristics of historical data and reduces the distance between the filled value and the actual value through the mutual confrontation process between the generator (G) and the discriminator (D) to train a more accurate generator. Aiming at the problem of poor prediction accuracy of LSTM, an optimization method with fast convergence speed and strong optimization ability is proposed, which greatly improves the prediction accuracy of LSTM. The effectiveness of the proposed method in predicting the life of rolling bearings under the condition of missing data is verified by a full-life cycle test example.

2. Data Imputation and RUL Prediction

2.1. GAIN

GAIN is a data-filling method proposed by Jinsung Yoon et al. [27] based on the Generative Adversarial Network (GAN) structure. Formally, it is not an improvement of GAN but an application of GAN in the direction of missing value processing. In GAIN, the goal of the G is to accurately fill in the missing data, and the goal of the D is to accurately distinguish the authenticity of the data, so the discriminator should minimize the classification error and the generator should maximize the discriminator's discrimination error, so that the two are in a process of mutual confrontation. In order to ensure that the disc D forces the G to learn the required distribution, GAIN will provide some additional information to D in the form of a hint vector H. In order to ensure that the D forces the G to learn the required distribution, GAIN will provide some additional information to D in the form of a prompt vector H. This hint reveals some information about the missing value of the original sample to D, which D uses to focus on the interpolation quality of a specific missing value. This hint reveals some information about the missing value of the original sample to D, which D uses to focus on the interpolation quality of a specific missing value.

2.1.1. Generator

The G takes a Data matrix ($\widetilde{X}$), a Random matrix (Z), and a Mask matrix (M) as inputs and outputs $\overline{X}$. $\widetilde{X}$ is a d-dimensional random variable sampled from the original data X; the size of d is determined by the sampling space; Z is a noise matrix with the same dimension of $\widetilde{X}$; and M is a d-dimensional marker matrix that reveals which element of $\widetilde{X}$ is the actual value. The definition of M is shown in Formula (1):

$$M_i = \begin{cases} 0, & \dot{X}_i = null \\ 1, & otherwise \end{cases} \tag{1}$$

In the formula, $\dot{X}_i$ is the element corresponding to $\widetilde{X}$ in the original data X.

M and $\widetilde{X}$ can be mutually verified by the values of their respective elements. M is similar to an encryption operation that makes $\widetilde{X}$ more vivid. The definition of $\widetilde{X}$ is as shown in Formula (2):

$$\widetilde{X}_i = \begin{cases} \dot{X}_i, \ M_i = 1 \\ 0, \quad otherwise \end{cases} \tag{2}$$

$\overline{X}$ is the output of the G, defined as shown in Formula (3):

$$\overline{X} = G(\widetilde{X}, M, (1 - M) \otimes Z) \tag{3}$$

In the formula, $\otimes$ represents the matrix by element multiplication, and $G(\cdot)$ is the generator function, which is expressed formally as: $\widetilde{X} \otimes \{0,1\}^d \otimes [0,1]^d$, $\{0,1\}^d$ is a matrix with the same dimension (d-dimension) as $\widetilde{X}$ and contains only 0 and 1, $[0,1]^d$ is a matrix with the same dimension as $\widetilde{X}$ and all elements in (0, 1). Specifically, when GAIN is trained, this is equivalent to G replacing the real value with 0, and filling the missing value with the value in the noise matrix Z, thus generating a new matrix $\overline{X}$. $\overline{X}$ does not contain any value in the original data X, G will generate a fill value for each position, even if the position does not exist and is missing.

2.1.2. Discriminator

In GAIN, the discriminator D functions similarly to a standard GAN, serving as an adversary to train the generator G. However, unlike a standard GAN, where the generator produces entirely genuine or entirely fake outputs, the generator of GAIN produces results that are partly genuine and partly interpolated. The discriminator's role is not to discern the veracity of the entire vector but rather to distinguish between genuine values and interpolated ones. This is similar to predicting the value in M, rather than determining the authenticity of the entire output. Formally, the discriminator function can be defined as: $D(\cdot) : A, A \in [0,1]^d$. The hint vector H actively provides additional information to D, which affects the output of the discriminator and strengthens the confrontation process between G and D. Therefore, a new discriminator can be defined as: $D(\cdot) : A \otimes H$, where $H \in [0,1]^d$. After redefining discriminator, the probability of discriminator misjudging $\hat{X}$ can be reduced by adjusting H, forcing the output of generator G to be closer to the actual value.

2.1.3. Antagonism Target

In order to make the output of G closer to the actual value, GAIN trains D by maximizing the probability of correctly predicting M and trains G by minimizing the probability that D can correctly predict M. Similar to GAN, the objective function is defined as Formula (4):

$$V(D,G) = \theta_{\hat{X},M,H}[M^T \log D(\hat{X}, S) + (1 - M)^T \log(1 - D(\hat{X}, S))] \tag{4}$$

Therefore, the training objectives of GAIN are shown in Formula (5):

$$\min_G \max_D V(D,G) \tag{5}$$

For the discriminator, its working mode belongs to a binary classification problem, so the cross entropy is used to define the loss function. The loss function is as shown in (6):

$$\ell(a,b) = \sum_{i=1}^{d} [a_i \log(b_i) + (1 - a_i) \log(1 - b_i)] \tag{6}$$

In the formula, a_i denotes the element in the D, which is predicted in the result matrix, b_i denotes the element in M corresponding to a_i. Let $\hat{M} = D(\hat{X}, S)$, then the training

target is converted into a function of whether D can accurately predict M, as shown in Formula (7):

$$\min_{G} \max_{D} \theta[\ell(M, \hat{M})] \tag{7}$$

2.2. LSTM

In order to overcome the defect that RNN cannot learn the remote dependence of long time series due to the disappearance of gradient, LSTM improves the model's ability to capture long time series sample information by adding three control units on the basis of RNN [28]: the forgetting gate f_t, the input gate i_t, and the output gate o_t. The basic structure of LSTM is shown in Figure 1.

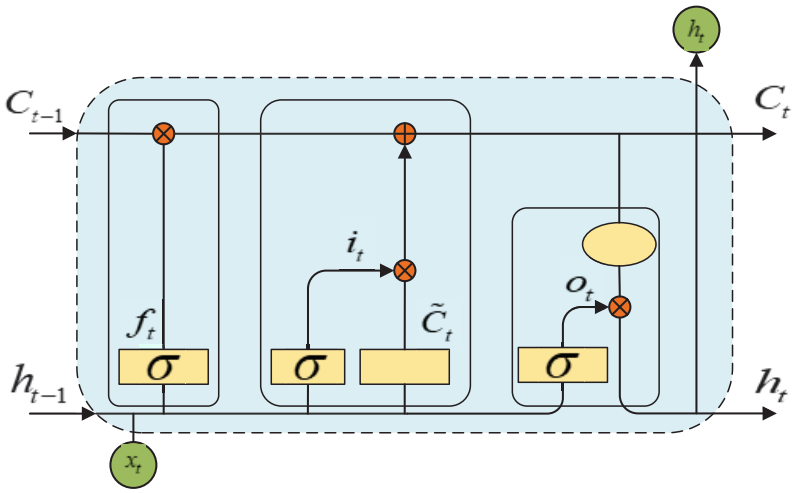


Figure 1. The basic architecture of LSTM.

The three control units of LSTM have different functions: the input gate determines how much information is retained in the memory unit, the output gate controls the output information of the current unit, and the forgetting gate determines the degree of forgetting of the information at the previous time point (memory degree). The three control unit vector calculation formulas are shown in (8):

$$\begin{cases} f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \end{cases} \tag{8}$$

In the formula, σ is the sigmoid function, and the value generated determines whether the information at the previous moment is retained or partially retained; W_f, W_i, and W_o are the weight matrix of forgetting gate, input gate, and output gate; b_f, b_i, and b_o are the bias terms of the three.

C_t, $\widetilde{C}_t$, and h_t are the memory state, input state and output state after updating at time t, respectively. The calculation formula is shown in (9):

$$\begin{cases} C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \\ \widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ h_t = o_t * \tanh(C_t) \end{cases} \tag{9}$$

In the formula, x_t is the input at time t: tanh is a hyperbolic tangent activation function, which can scale the output state to $(-1, 1)$.

The number of hidden layer neurons, learning rate, regularization coefficient, and training epochs are key parameters that define an LSTM model and directly affect its predictive performance. Therefore, optimizing these parameters should be the primary goal.

3. Prediction Model under the Condition of Missing Data

In order to solve the problem that the random missing values of the wind turbine bearing vibration signal affect the accuracy of RUL prediction, a new strategy is proposed: AS-GAIN-LSTM. Firstly, the missing values in $\widetilde{X}$ are accurately interpolated, and then the optimized LSTM is used for RUL prediction. The overall flow chart is shown in Figure 2.

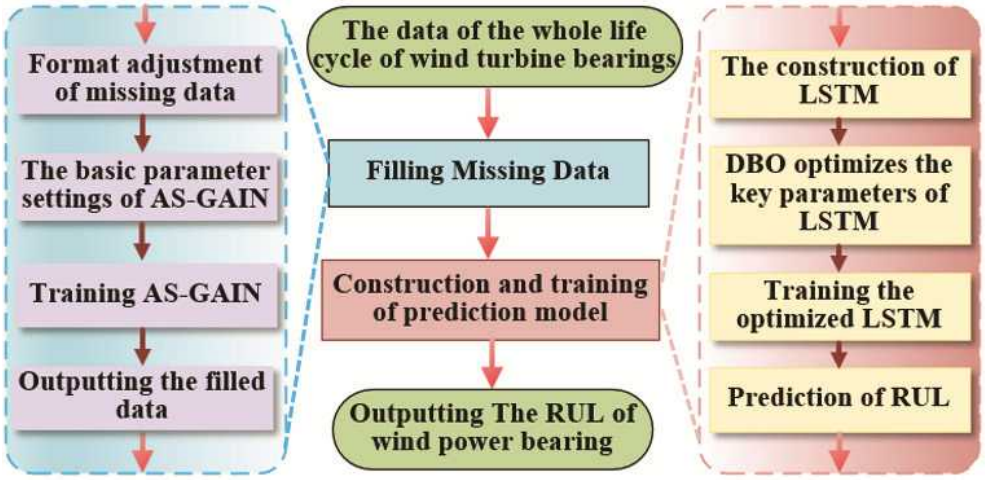


Figure 2. Flowchart for AS-GAIN-LSTM.

The AS-GAIN architecture is shown in Figure 3, and the DBO-LSTM optimization process is shown in Figure 4.

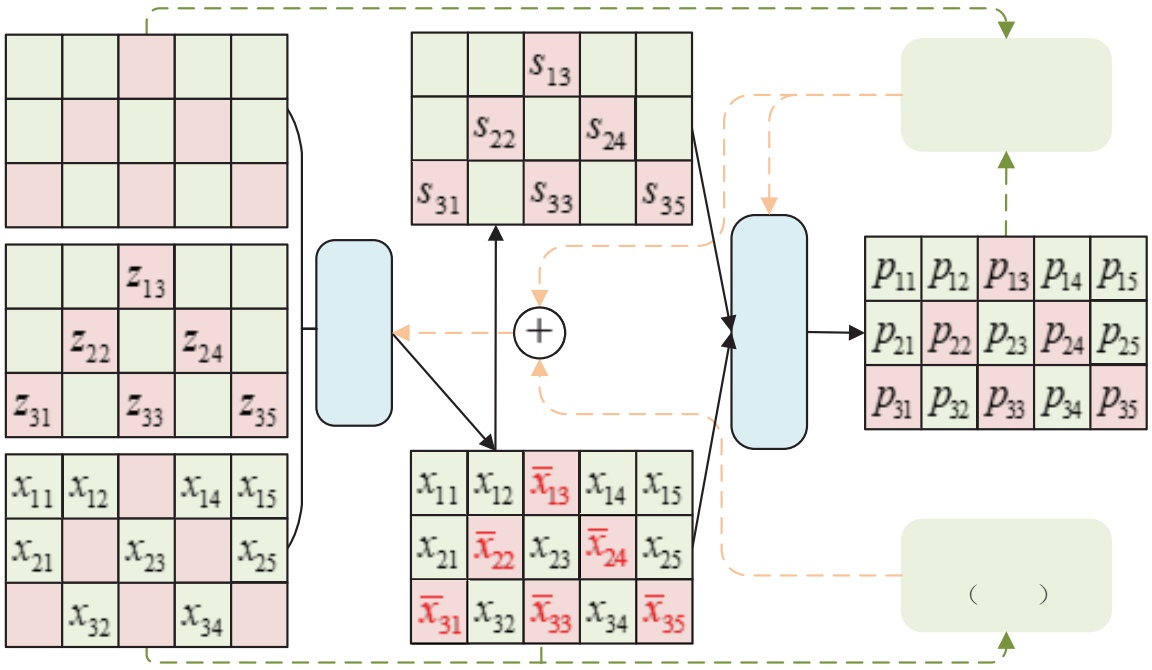


Figure 3. Architecture diagram of AS-GAIN.

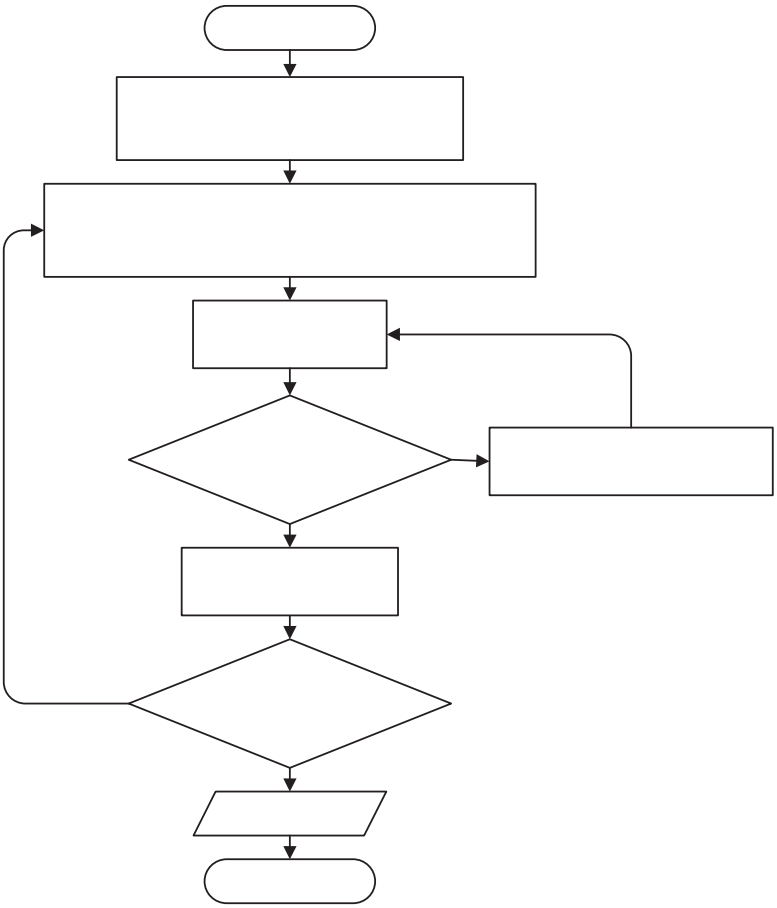


Figure 4. Flowchart for optimizing LSTM with DBO.

3.1. AS-GAIN

In GAIN, the hint vector H enhances the adversarial process between the generator G and the discriminator D. However, it is limited by its definition, providing D with only a limited amount of information. Moreover, the elements of H require manual setting, which increases uncertainty in model performance. To address these issues, this paper introduces average similarity as an augmentation to GAIN's adversarial process, known as AS-GAIN. In the process of model training, the average similarity matrix (AS) between the filling value in the filling matrix $\hat{X}$ and its grouping elements is constructed, and the average similarity matrix S and the filling matrix $\hat{X}$ are used as the two inputs of the discriminator. The discriminator outputs a probability matrix about the filling matrix, and its elements describe the probability that the value of the corresponding filling matrix is the actual value. The $\hat{X}$ refers to the complete data vector after filling, which observes the local information of $\widetilde{X}$, and then replaces the position of the corresponding value 0 in $\widetilde{X}$ with the filling value in $\overline{X}$. The definition of $\hat{X}$ is shown in Formula (10):

$$\hat{X} = M \otimes \widetilde{X} + (1 - M) \otimes \overline{X} \tag{10}$$

The introduction of mean similarity in the discriminator aims to primarily enhance its ability to capture information regarding the filled-in values, thereby overcoming the uncertainty of H and enabling the generator G to obtain more precise results during adversarial training. Average similarity describes the similarity between the imputed values and the actual values in each feature dimension, which indirectly reflects the reliability of the imputed values in an adversarial process. Upon receiving this information, the discriminator immediately provides feedback to the generator, allowing G to generate more realistic values for missing data in different locations. The discriminator function $D(\cdot)$ is

redefined as: $A \otimes S, S \in [0,1]^d$, S is the average similarity matrix with the same dimension as $\hat{X}$. The definition of S is shown in Formula (11):

$$S = [S_1, S_2, \cdots, S_n]^T \tag{11}$$

$$S_i = [s_{i1}, s_{i2}, \cdots, s_{im}], i = 1, 2, \ldots n \tag{12}$$

$$s_{ij} = \begin{cases} 1 \quad , \hat{m}_{ij} = 1 \\ \sum_{j=1}^{m} (1 - \sqrt{(\hat{x}_{\hat{m}_{ij}=0} - \hat{x}_{ij})^2}/\max)/m, \quad \hat{m}_{ij} = 0 \end{cases} \tag{13}$$

In the formula, n is the number of groups of $\hat{X}$, m is the number of elements contained in each group of data, $\hat{m}$ is the label matrix of $\hat{X}$, and the max is the maximum Euclidean distance related to the filling value ($\hat{x}_{\hat{m}_{ij}=0}$).

Assuming the minimum batch size for input of AS-Gain is 1×5, with a single missing value, we can generate a matrix of size 1×5 through generator D. This matrix can be transformed (using Formula (10)) to obtain a filled matrix of size 1×5. The similarity matrix S can then be computed using Formulas (11)–(13), as depicted in Figure 5.

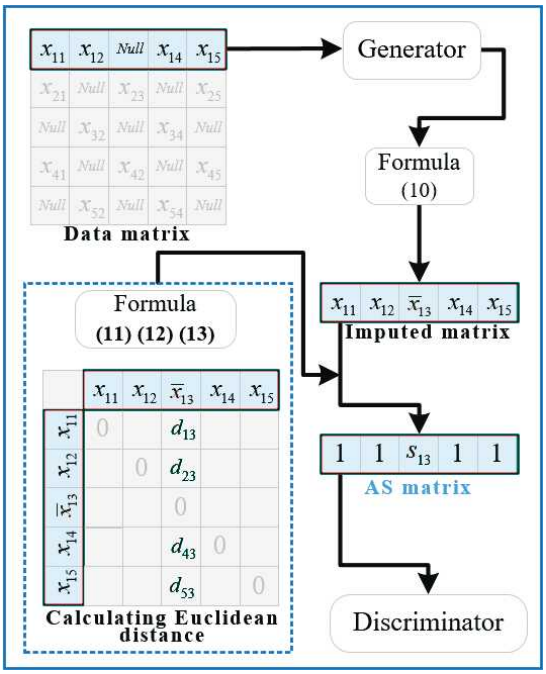


Figure 5. The calculation process diagram of S.

In Figure 5, d represents the Euclidean distance between two elements. AS-GAIN provides an average similarity based on the minimum batch size for each missing value, which reveals the implicit association between the values generated by G in one training round and the ground truth. This association is then fused into D, thereby enhancing its discrimination capabilities. It should be noted that the minimum batch size may vary depending on the type of data and different features within the same type. Suppose the minimum batch size is excessively large. In that case, it can weaken the connection between the imputed value and the actual value, thereby affecting the generation and discrimination capabilities of G and D. Conversely, if the minimum batch size is too small, it can incorrectly describe the relationship between the imputed value and the actual value, thus degrading the performance of G and D. During testing, we found that the smallest batch size that best suits the characteristics of the experimental data employed in Section 5.1 is between 13 and 18.

By improving the discriminator D, AS-GAIN is obtained on the basis of GAIN, and its architecture is shown in Figure 3.

3.2. LSTM Is Optimized by DBO

The LSTM model consists of multiple neurons, connections, and memory units, each incorporating multiple gate mechanisms. Additionally, the model's parameter space is typically vast, encompassing numerous weights and biases. These factors directly impact the model's feature extraction performance, prediction accuracy, and memory overhead. Manual tuning of model parameters requires multiple trials and adjustments, resulting in lower efficiency and unsatisfactory outcomes. To address this issue, a proposal was made to optimize LSTM's key parameters using the Ant Colony Optimization algorithm. The Dung Beetle Optimizer (DBO), operating within the parameter space, can search for the optimal parameter combination that minimizes the loss function or maximizes model performance metrics. Based on this, the design of the LSTM model presented in this paper is shown in Table 1.

Table 1. The setting of parameters for LSTM.

Parameters	Setting
input size	2375×1
Execution environment	CUP
Learn rate drop period	400
Number of layers	3
Learn rate drop factor	0.2
Step length of prediction	8
Activation function	sigmoid
Mini batch size	64
Number of neurons	Optimized by DBO
Initial learning rate	Optimized by DBO
Number of trainings	Optimized by DBO
L2 regularization	Optimized by DBO

The last four primary parameters in Table 1 were optimized through DBO.DBO is a new optimization algorithm [29], inspired by the foraging behavior of the Dung Beetle. It combines the strategies of global search and local development to achieve fast convergence and high accuracy optimization. The population of the dung beetle optimization algorithm is composed of four representative individuals: the ball-rolling dung beetle, the hatching dung beetle, the small dung beetle, and the thief dung beetle. Each individual represents a potential solution. They interact and collaborate to obtain the optimal solution by simulating the behavior and information exchange of dung beetles. The detailed process is showcased in Section 5.3 of this article.

In order to simulate the process of dung beetles pushing the dung ball and using the antenna to identify the direction of the straight line, the target point is allowed to move in the search space in accordance with the specified direction. Considering the sensitivity of dung beetles to light, the position change of dung beetles during the forward process is shown in Formula (14):

$$x_i(t) = x_{\mathrm{i}}(t-1) + \lambda \cdot k \cdot x_i(t-2) + \mu \cdot \Delta x \tag{14}$$

$$\Delta x = |x_i(t-2) - x_w| \tag{15}$$

In the formula, t represents the current number of iterations; $x_i(t)$ describes the position of the t-th iteration of the i-th dung beetle; k is the perturbation factor, and the value range is (0, 0.2); λ and μ are fixed value parameters: Δx describes the intensity of light, and the calculation formula is shown in Formula (15): x_w is the worst local position of the dung beetle.

When the dung beetle encounters obstacles in the process of advancement, it is necessary to re-formulate the route and update the position calculation formula of the dung beetle, as shown in Formula (16):

$$x_i(t) = x_i(t-1) + \tan\theta \cdot |x_i(t-1) - x_i(t)| \tag{16}$$

In the formula, θ is the size of the perturbation, and the value range is $(0, \pi)$.

Reproduction is an important behavior in the dung beetle population. The spawning area of female dung beetles should be set within a safe range. The boundary definition is shown in Formula (17):

$$\begin{cases} {b_L}^* = \max(x_b(1-\rho), b_L) \\ {b_U}^* = \min(x_b(1+\rho), b_U) \end{cases} \tag{17}$$

$$\rho = 1 - t/T_{\max} \tag{18}$$

In the formula, ${b_L}^*$ and ${b_U}^*$ are the lower and upper bounds of the safe spawning area; b_L and b_U are the lower and upper bounds of the optimization problem; x_b is the local optimal position of the dung beetle: ρ is the boundary dynamic adjustment range, and the calculation process is as follows (18): $T_{\max}$ is the maximum number of iterations.

After determining the spawning area of the female dung beetle, the position of the egg ball will change. The formula for calculating the position of the egg ball is updated as shown in Formula (19):

$$B_i(t) = x_b + b_1 \cdot (B_i(t-1) - {b_L}^*) + b_2 \cdot (B_i(t-1) - {b_U}^*) \tag{19}$$

In the formula, $B_i(t)$ describes the position of the t-th iteration of the i-th egg ball; b_1 and b_2 are two independent vectors of $1 \times N$, N is the dimension of the optimization problem.

In order to simulate the foraging behavior of small dung beetles in the population, the foraging area is first defined as shown in Formula (20):

$$\begin{cases} {b_L}^\bullet = \max({x_b}^*(1-\rho), b_L) \\ {b_U}^\bullet = \min({x_b}^*(1-\rho), b_U) \end{cases} \tag{20}$$

In the formula, ${b_L}^\bullet$ and ${b_U}^\bullet$ are the lower and upper bounds of the optimal foraging region, ${x_b}^*$ is the global optimal position of the dung beetle.

Secondly, the position of the small dung beetle foraging is updated, and the calculation formula is shown in Formula (21):

$$C_i(t) = x_b + c_1 \cdot (C_i(t-1) - {b_L}^\bullet) + c_2 \cdot (C_i(t-1) - {b_U}^\bullet) \tag{21}$$

In the formula, $C_i(t)$ describes the position of the t-th iteration of the i-th dung beetle, c_1 is a random number that obeys the normal distribution, and c_2 is a random number in (0, 1).

There are food contenders in the dung beetle population. Assuming that the optimal competition location is, the position update of the i-th thief at the t-th iteration is shown in Formula (22):

$$D_i(t) = x_{sp} + d_1 \cdot d_2 \cdot (\left|D_i(t-1) - x_b\right| + \left|D_i(t-1) - x_{sp}\right|) \tag{22}$$

In the formula, d_1 is a random vector that obeys normal distribution and has a size of $1 \times N$, and d_2 is a fixed constant.

The dung beetle algorithm optimization LSTM flow chart is shown in Figure 4.

3.3. Model-Evaluation Index

The AS-GAIN evaluation index uses root mean square error (*RMSE*) and mean absolute error (*MAE*). The DBO-LSTM evaluation index uses *RMSE*, *MAE*, and mean absolute percentage error (*MAPE*). The calculation methods are shown in Formula (23) [30]:

$$\begin{cases} RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \\ MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \\ MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \end{cases} \tag{23}$$

In the formula, n is the number of samples; y_i is the actual value; $\hat{y}_i$ is the predicted value (filled value by AS-GAIN).

4. Experiment Setting

The bearing life data set provided by the PHM2012 data challenge [31] is used to verify the effectiveness of the proposed method. The vibration data includes the axial and radial signals collected by two vibration sensors. The stop threshold of signal acquisition is 20 g, the sampling frequency is 25.6 kHz, and the sampling time is 0.1 s. The data acquisition device is shown in Figure 6, and the working condition is divided in Table 2.

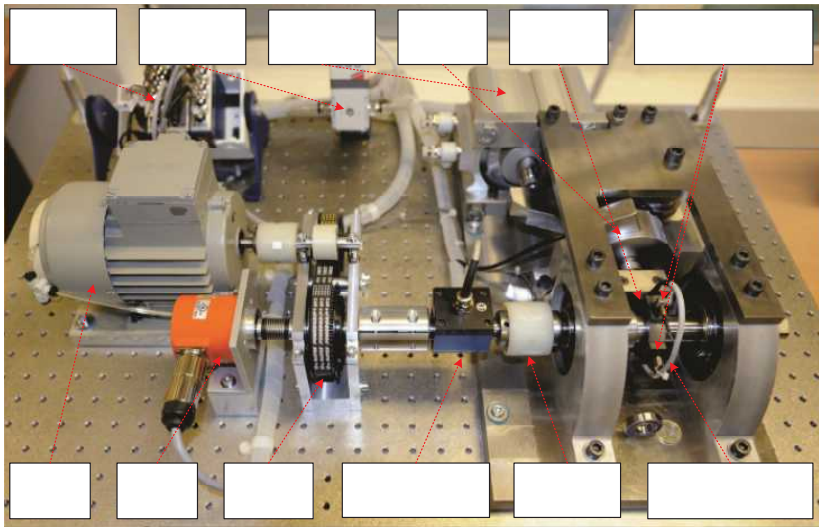


Figure 6. The device for collecting experimental data.

Table 2. Operating condition of the datasets.

Working Condition	Load/N	Speed/(r·min^{-1})
1	4000	1800
2	4200	1650
3	4500	1500

In this study, the ball bearings being investigated share a similar construction principle as those found in other sizeable rotating machinery components. Additionally, monitoring data for wind turbine equipment parts as they approach failure is currently scarce or nonexistent [32]. This accurately predicts the lifespan of wind turbine bearings using publicly available datasets covering the entire life cycle of the bearings. Bearing 1_3 under condition 1 exhibits an evident degradation trend that aligns with the degradation trajectory of wind turbine bearings. Therefore, it serves as experimental data for validating DBO-

LSTM and AS-GAIN-LSTM. Due to the significant differences in data volume and features between bearing 1_3 and bearing 2_3, the latter is used as an experimental subject for validating AS-GAIN. Firstly, bearing 1–3 and bearing 2–3 data are used to simulate the working conditions with missing data to verify the AS-GAIN filling performance. Secondly, the prediction performance of DBO-LSTM is verified by bearing 1–3 degradation stage data. Finally, the degradation stage data after bearing 1–3 filling is used to verify the comprehensive performance of AS-GAIN-LSTM under the condition of missing data.

The life cycle of rolling bearings can be divided into two stages: the health stage and the degradation stage. Due to the large proportion of the healthy stage and the relatively stable signal, selecting this stage to train the model and perform life prediction has a great influence on the prediction accuracy of the model. Therefore, selecting the first point of degradation (FPD) and determining the degradation stage is particularly important. WANG et al. [5] assumed that the degradation amount of the healthy stage obeys the 3 rule proposed by the normal distribution. According to this criterion, the degradation interval of the whole life cycle of the rolling bearing is determined, and the RUL prediction experiment is carried out.

The root mean square (RMS) can be employed to gauge the degradation of bearings [33], thus treating the RMS of vibration signals as experimental objects (with every 2560 original data points constituting a sample point), with a training set-to-test set ratio of 7:3. To validate the performance of DBO-LSTM and AS-GAIN-LSTM, five experimental schemes were designed, as shown in Table 3. The parameters of some parts in GAIN and AS-GAIN are set as shown in Table 4.

Table 3. Design of the experimental plans.

Prediction Plan	Missing Proportion	Imputer	Prediction Model
Plan A	0%		DBO-LSTM
Plan B	0%		LSTM
Plan C	0%		SSA-LSTM
Plan D	0%		RNN
Plan E	10%	AS-GAIN	DBO-LSTM
Plan F	20%	AS-GAIN	DBO-LSTM

Table 4. Set of parameters for GAIN and AS-GAIN.

Parameters	GAIN	AS-GAIN
Mini batch size	16	16
Hyperparameter	15	15
Iteration	1000	1000
Missing rate	0.1~0.8	0.1~0.8
Hint rate	1	
The size of S_i		1×25

5. Experimental Analysis

5.1. Performance Verification of AS-GAIN

To validate the performance of AS-GAIN in filling in missing data, experiments employed the full-life cycle data of bearing 1_3 and bearing 2_3. Figure 7a,b separately simulate the original signal diagrams for bearing 1_3 and bearing 2_3 under missing completely at random (MCAR) with a missing rate of 10%. It should be noted that in the absence of MAR or MNAR patterns, the probability of missing values may be related to the characteristics of other observed values or the nature of the missing value itself. In such cases, the occurrence of missing data can introduce bias into the dataset, which would weaken the influence of the average similarity matrix S in AS-GAIN. How AS-GAIN can adapt to these two missing pattern scenarios remains an open question. Figure 8 depicts the missing value distribution pattern based on the original signal in Figure 7, with each missing value

position corresponding to the RMS of bearing 1_3 and bearing 2_3 (bearing 1_3 and bearing 2_3 refer to the RMS of bearing 1_3 and the RMS of Bearing 2_3, respectively). As depicted in Figure 8, under this MCAR missing pattern, all intervals essentially contain the missing parts; however, there exist some differences in the distribution of missing values for bearing 1_3 and bearing 2_3: severe missing in bearing 1_3 is evenly distributed throughout the entire interval, whereas in bearing 2_3, it mainly occurs between data points 480 and 720. Suppose there is a disproportionate concentration of missing values in a particular region. In that case, the accuracy of the relationship captured by matrix S may be compromised when using AS-GAIN to fill in these values. This could result in suboptimal outcomes not only for AS-GAIN but also for other filling algorithms. In addition, when the missing rate is 10%, the number of missing values for bearing 2_3 is significantly lower than that for bearing 1_3. This is due to the inherent differences in magnitude between the two sets of original data. Each data point carries with it distinct characteristics of its original data, resulting in entirely different traits exhibited by the two sets. This will serve as a test for the stability and reliability of the model.

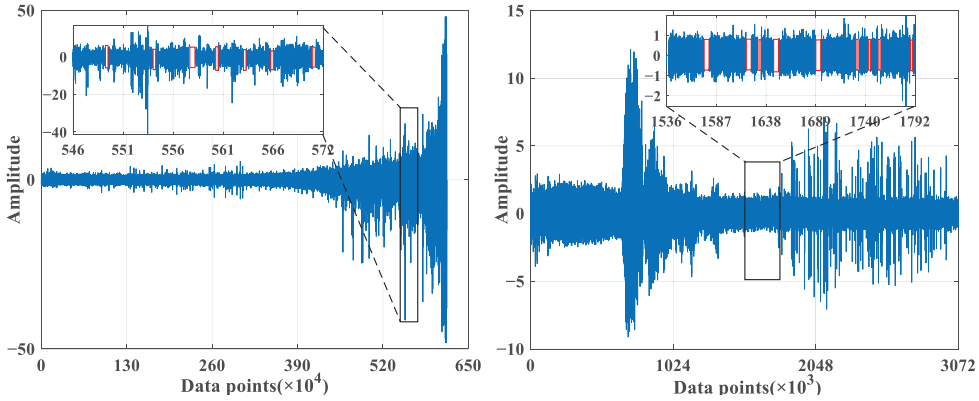


Figure 7. The distribution of missing data in the original signal. (**a**) The original signal for bearing 1_3. (**b**) The original signal for bearing 2_3. (The red flag indicates the missing value).

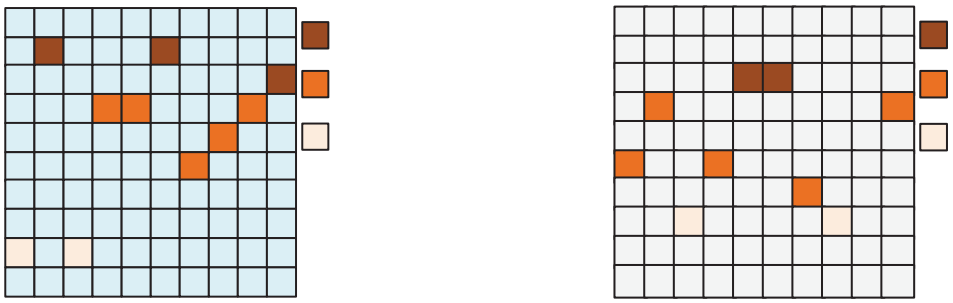


Figure 8. Distribution of different bearings with 10% missing values. (**a**) Bearing 1_3 with 10% missing values. (**b**) Bearing 2_3 with 10% missing values.

For bearing 1_3, Figure 9a,b show the RMSE and MAE of different filling models under different missing rates and evaluate the filling performance of different models through these two indicators. From Figure 7a,b, it can be seen that with the increase in bearing 1_3 missing rate, the performance of each filling model gradually decreases. However, compared with other models, the performance of AS-GAIN is still the best under the same missing rate. Moreover, with the increase in the missing rate, the growth rate of the RMSE

and MAE of AS-GAIN is also the lowest, which indicates that AS-GAIN has better stability. Especially in scenarios with a high missing rate (>80%), AS-GAIN has obvious advantages. When the missing rate is 80%, the RMSE of AS-GAIN is 15%, 35%, 40%, and 60% lower than that of GAIN, KNN, MissForest, and MeanImputer, respectively. Compared with GAIN, KNN, MissForest, and MeanImputer, MAE decreased by more than 20%, 40%, 30%, and 60%, respectively.

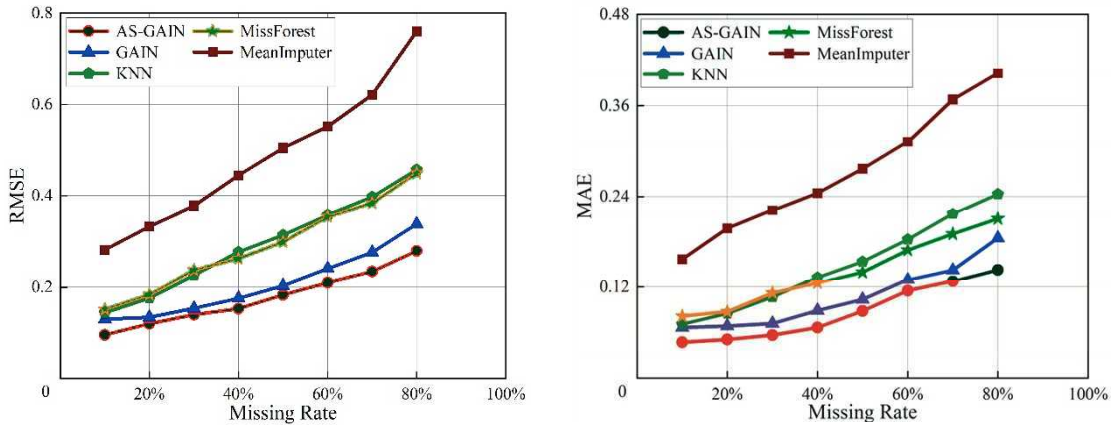


Figure 9. Different evaluation metrics for bearing 1_3 at different missing rates. (**a**) RMSE of bearing1_3 at different missing rates. (**b**) MAE of bearing 1_3 at different missing rates.

For bearing 2_3, Figure 10a,b show the RMSE and MAE of different imputation models under different missing rates. Compared with Figure 9a,b, the differences of each model under different missing rates are smaller, and the values of RMSE and MAE are also smaller. This is because the sample size of bearing 2_3 is smaller and the data itself has a higher similarity. Among the five filling models, the comprehensive index performance of AS-GAIN is still optimal. Especially when the missing rate is 80%, the RMSE of AS-GAIN is better than that of GAIN, KNN, MissForest, and MeanImputer by 8%, 20%, 25%, and 30%, respectively. Compared with GAIN, KNN, MissForest, and MeanImputer, MAE decreased by more than 2%, 15%, 15%, and 35%, respectively. This verifies that AS-GAIN has a good filling effect when dealing with different missing rates and different missing data.

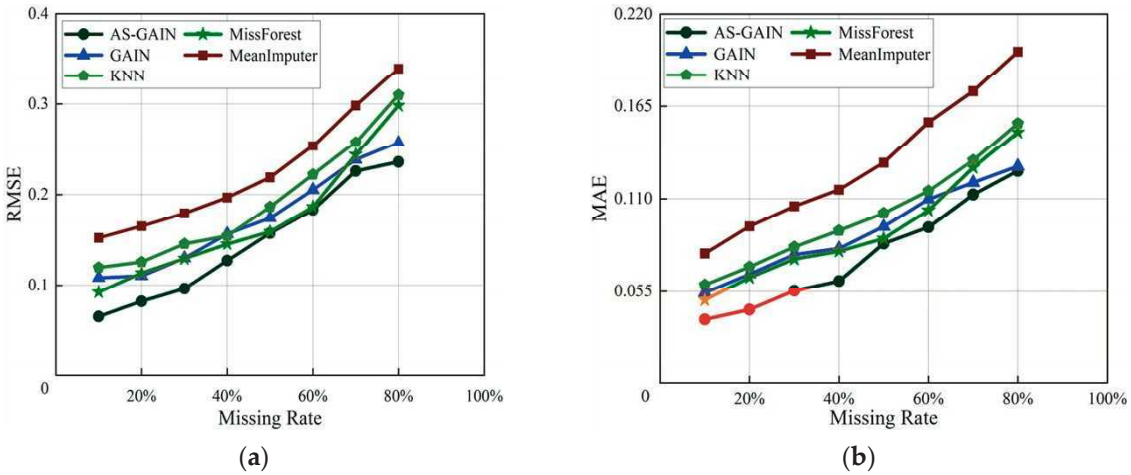


Figure 10. Different evaluation metrics of bearing 2_3 at different missing rates. (**a**) RMSE of bearing 2_3 at different missing rates. (**b**) MAE of bearing 2_3 at different missing rates.

As shown in Figures 9 and 10, as the missing rate gradually increases, imputation errors also increase for different models. This is due to the fact that missing values contain

information about the original data's characteristics, which, when lost, can cause a decrease in the accuracy of prediction models. As the number of missing values increases, the loss of reliable information becomes more significant, resulting in a decline in model prediction accuracy. Based on the trend of the evaluation indicators, it can be observed that the performance decline of AS-GAIN is the most gradual.

5.2. Performance Verification of DBO-LSTM

In order to verify the prediction performance of DBO-LSTM, experiments were carried out using Plans A, B, C, and D. According to the 3 rule, the FPD was determined to be the 1633th data point position, and the degradation interval was [1633, 2375]. Figure 11 describes the health stage and degradation stage of the full-life cycle of bearing 1_3.

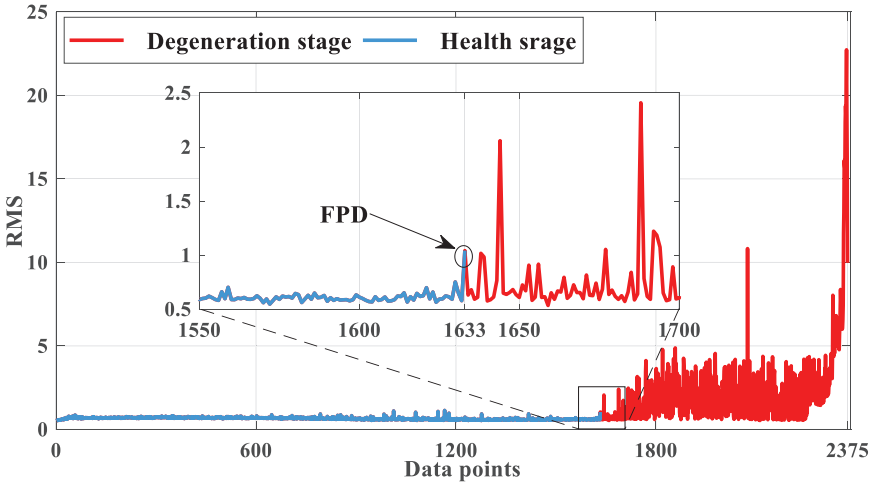
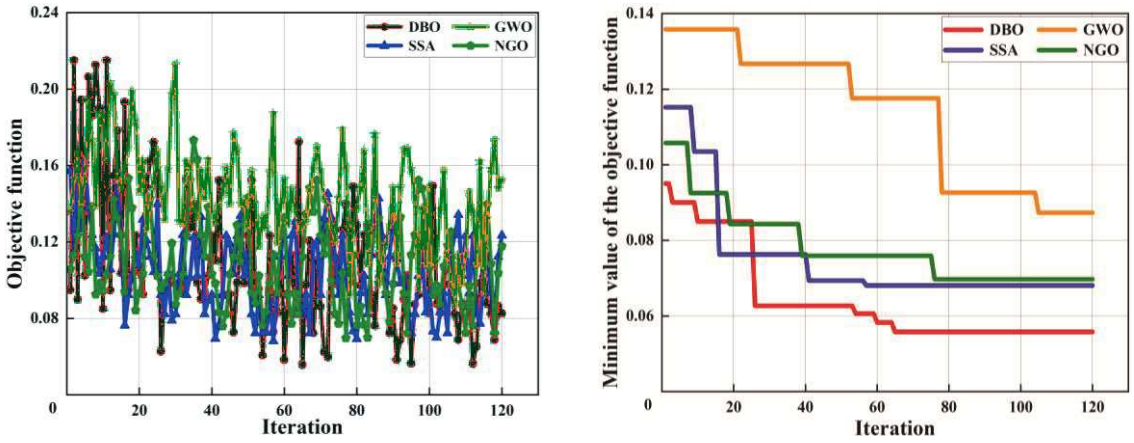


Figure 11. The FPD of bearing 1_3 under the 3 criterion.

In order to improve the generalization ability of the model, MAPE is used as the objective function, Adam0 is used as the optimizer, and the critical parameters of LSTM are optimized based on DBO. In order to give full play to the ability of DBO optimization, the parameters of LSTM are searched 120 times, and the search range of the number of hidden layer neurons is set to 50~150 according to the size of the training set. In order to improve the convergence speed of the prediction model, the initial learning rate search range is set to 0.001~0.01. In order to prevent over-fitting of the prediction model, the training times and regularization coefficients are set to 25~100 and 1×10^{-10}~1×10^{-3}, respectively. The DBO optimization process is shown in Figure 12. It can be seen from Figure 12a that the objective function value tends to be stable and reaches the minimum value after the number of iterations reaches 66. It can be seen from Figure 12b that the minimum value of the objective function is reduced from 0.09503 to 0.0558 in the optimization process. In comparison to other optimization algorithms, it is evident that for the data utilized in this paper, DBO demonstrates a swifter convergence rate and also identifies the optimal solution prior to the cessation of iterations.

The parameter combination corresponding to the minimum value of the optimized objective function is the optimal parameter combination. The DBO optimization results are shown in Table 5. According to Table 5, the optimization parameters of the number of hidden layer neurons, the initial learning rate, the number of training times, and the regularization coefficient are 118, 0.006623, 93, and 1×10^{-10}, respectively. This set of optimization parameters will be used to construct the LSTM model.



(**a**) (**b**)

Figure 12. Optimization process of different algorithms for LSTM. (**a**) Objective function and iteration. (**b**) The minimum value and iteration.

Table 5. The parameters of LSTM optimized by DBO.

Parameters	Optimized Scope	Optimized Results
Number of neurons	50~150	118
Initial learning rate	0.001~0.01	0.006623
Number of trainings	25~100	93
L2 regularization	1×10^{-10}~1×10^{-3}	1×10^{-10}

Figure 13 is the comparison chart between the predicted results and the actual values of Plans A, B, C, and D. As shown in Figure 13, it is evident that Plan D exhibits the most significant deviation from the actual values in terms of prediction results. This indicates that, among the various prediction models, LSTM demonstrates superior predictive ability. Furthermore, the prediction outcomes of Plan A surpass those of Plan B and C, suggesting that employing DBO to optimize LSTM is a feasible approach for enhancing its predictive performance.

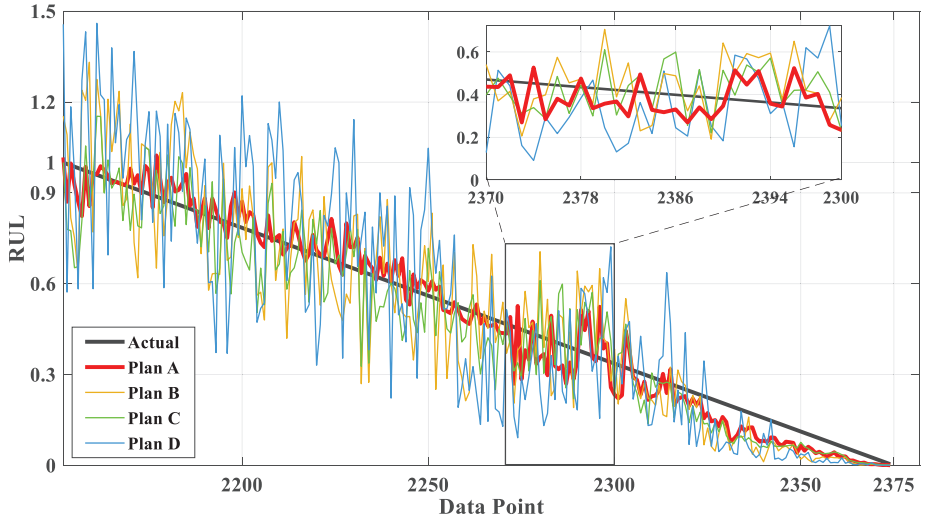


Figure 13. Comparison of the predictions of different plans of bearing 1_3.

Table 6 shows the prediction errors of A, B, and C Plans. It can be seen from Table 5 that the MAE, RMSE, and MAPE of Plan A are 5.78%, 21.06%, and 18.06% lower than those of Plan C, respectively. The MAE, RMSE, and MAPE of Plan A were 44.17%, 40.40%, and

53.85% lower than those of Plan B, respectively. The MAE, RMSE, and MAPE of Plan A were 67.45%, 59.30%, and 71.86% lower than those of Plan D, respectively. When comparing Model B and Model D individually, Model B shows a reduction in the MAE, RMSE, and MAPE of Model D by 41.70%, 31.72%, and 39.03%, respectively. These results sufficiently demonstrate that DBO-LSTM can more accurately describe the degradation trajectory of rolling bearings compared to LSTM and SSA-LSTM. Additionally, LSTM shows better predictive performance compared to RNN.

Table 6. Different plans were used to evaluate the prediction of bearing 1_3.

Plans	MAE	RMSE	MAPE (%)
Plan A	0.4450	1.4794	5.5800
Plan B	0.7971	2.4821	12.090
Plan C	0.4698	1.8742	6.8100
Plan D	1.3673	3.6352	19.831

In order to better emphasize the predictive capability of LSTM, Table 7 presents the runtime of various models under identical conditions (CPU: i5-1240P, RAM: 32 GB, Operating System: Windows 11) and the same experimental data (Data Size: 2375 $\times$ 1). As evidenced by the table, LSTM has the shortest runtime, as it does not factor in the optimization algorithm's runtime. In comparison to the RNN model, LSTM also possesses the ability to complete prediction tasks more quickly. When these models are executed with superior devices or alternative data, shorter runtimes may be attained.

Table 7. Running time of different models.

Models	Running Time (s)
DBO-LSTM	2850 ± 53
LSTM	1500 ± 25
SSA-LSTM	2940 ± 15
RNN	3840 ± 45

5.3. Performance Verification of AS-GAIN-LSTM

To validate the efficacy of the AS-GAIN-LSTM approach, experiments were carried out with Plans A, E, and F. Initially, bearing 1_3's missing rates in the degradation interval (data after FPD in Figure 11) were set at 10% and 20% under the MCAR condition. The missing data simulated from the original signal are illustrated in Figure 14a,b. The corresponding RMS missing distribution of bearing 1_3 is depicted in Figure 15a,b.

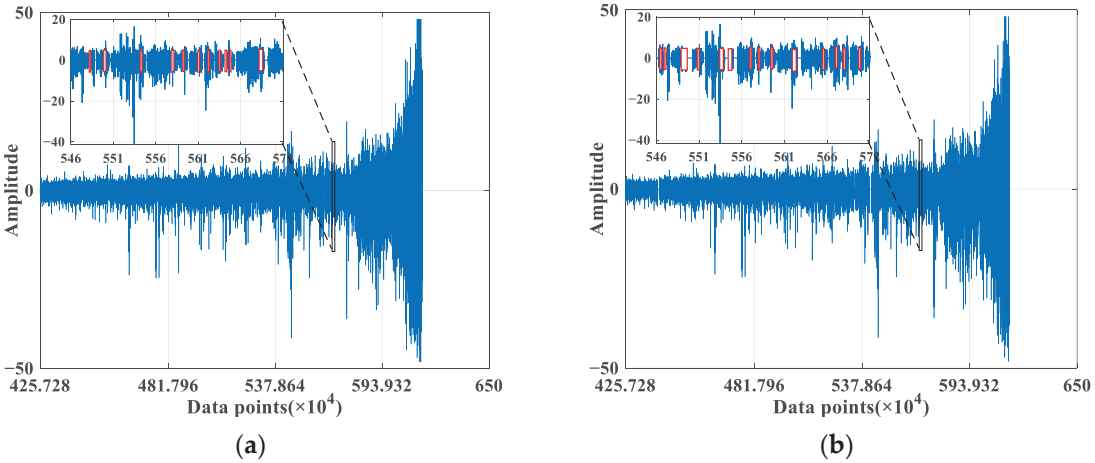
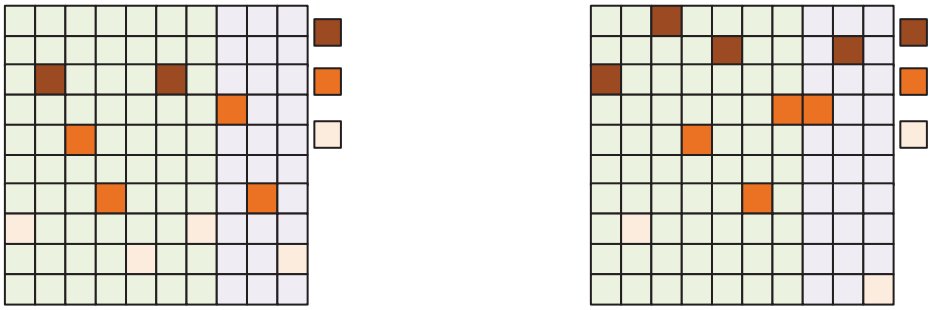


Figure 14. Distribution of different numbers of missing values under bearing 1_3. (**a**) The missing rate is 10%. (**b**) The missing rate is 20%. (The red flag indicates the missing value).



(**a**) (**b**)

Figure 15. Distribution of bearing 1_3 with different missing rates. (**a**) The missing rate is 10%. (**b**) The missing rate is 20%.

When comparing two different scenarios of random missing data, with missing rates of 10% and 20%, the missing values are distributed evenly in both the training and testing sets. The amount of missing data can potentially impact the performance of AS-GAIN-LSTM, and the subsequent sections will provide a more detailed explanation of the results.

Figure 16 is a comparison chart of the actual results of RUL prediction after the proposed prediction model is used to fill the degradation interval data with missing ratios of 0%, 10% and 20%, respectively. The missing value filling method adopts AS-GAIN. The corresponding missing ratios of Plan A, E, and F were 0%, 10%, and 20%, respectively. Based on the depicted graph, it seems that the prediction from Plan A provides the most accurate approximation to the actual value. Initially, both Plan E and Plan F demonstrated more significant discrepancies with the actual values in comparison to later forecasts. This could potentially be attributed to some shared missing values in Figure 15a,b.

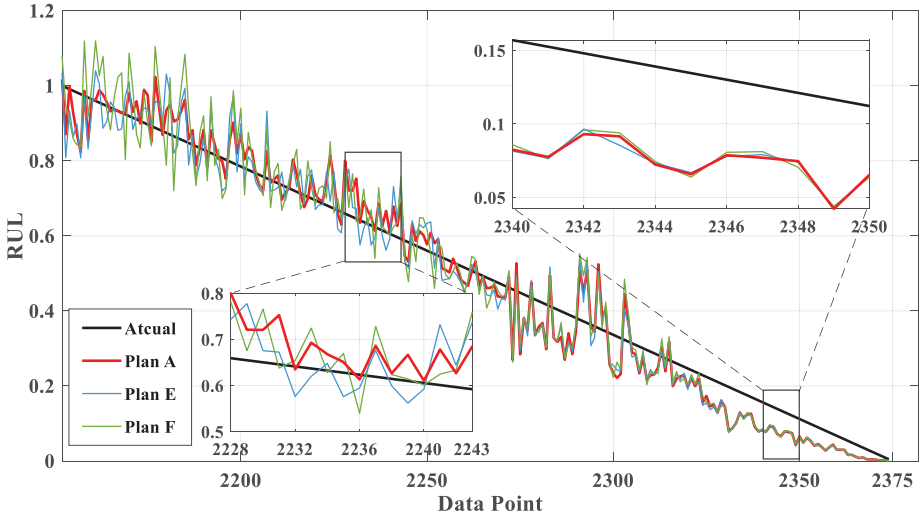


Figure 16. Comparison of the predictions of different plans of bearing 1_3 with missing data conditions.

After AS-GAIN fills the data with three missing ratios (0%, 10%, and 20%), DBO-LSTM is used to predict the RUL of the filled-in complete data. The prediction error is shown in Table 8. It can be seen from the Table that when there are 10% missing values in the data, the MAE, RMSE, and MAPE of using AS-GAIN for data filling and DBO-LSTM for RUL prediction are only 1.75%, 1.89%, and 8.31% higher than those of using DBO-LSTM for life prediction of the original complete data. When there are 20% missing values in the data, the MAE, RMSE, and MAPE of the AS-GAIN-filled and DBO-LSTM-based RUL predictions are only 3.28%, 3.30%, and 12.72% higher than those of the original complete data using

DBO-LSTM for life prediction. It can be seen that when there are missing values in the rolling bearing data, it is feasible to use the AS-GAIN-LSTM strategy for the prediction of RUL.

Table 8. Different plans were used to evaluate the prediction of bearing 1_3 with different missing conditions.

Plans	MAE	RMSE	MAPE (%)
Plan A	0.4450	1.4794	5.5800
Plan E	0.4527	1.5073	6.0435
Plan F	0.4596	1.5282	6.2898

As shown in Figure 16 and Table 8, when inputting missing rate data with varying degrees of incompleteness into DBO-LSTM, the prediction errors corresponding to high-missing-rate data were slightly higher than those associated with low-missing-rate data. According to the principle outlined in Section 3.1, matrix S calculates the Euclidean distance between the filled values of G within the same minimum batch and their actual counterparts, including the Euclidean distance between the filled values of G themselves. However, the G imputation value itself carries an error when compared to the actual value. As more missing values exist in the data, it is inevitable that S will contain more errors, which would lower its discriminatory power for distinguishing D. This, in turn, reduces the reliability of the information contained within the generated data. During LSTM training, the model analyzes the depth features of the data. When the data itself contains errors, it is inevitable that these will negatively impact the model, thereby reducing its predictive capabilities.

6. Conclusions

1. In the data preprocessing stage, aiming at the problem of random missing data in the process of signal acquisition of wind power bearings, AS-GAIN can be used to fill the missing values to obtain more accurate filling values because AS-GAIN can better describe the difference between the filling value and the actual value. In the process of confrontation between the generator and the discriminator, the information about the average similarity of the filled values is integrated into the discriminator, so that the generator pays more attention to the filling effect of the missing values. After many confrontation training sessions, a generator with excellent filling effects can be obtained. The experimental results show that the performance of AS-GAIN in processing the working condition data mentioned in this paper has obvious advantages compared with other models.
2. In the RUL prediction stage, the number of hidden layer neurons, learning rate, regularization coefficient, and training times of LSTM are optimized by using the characteristics of fast convergence and strong optimization ability of the dung beetle algorithm, so that LSTM has better prediction performance in dealing with the prediction problem of long time series. The experimental results show that compared with LSTM and SSA-LSTM, the MAE of DBO-LSTM is reduced by 5.78% and 44.17%, respectively; RMSE is reduced by 21.06% and 40.40% respectively; and MAPE is reduced by 18.06% and 53.85%, respectively.
3. AS-GAIN-LSTM is adopted to solve the problem that the wind power bearing data cannot be predicted by RUL that the accuracy of RUL prediction is low due to the presence of missing values. Firstly, AS-GAIN is used to accurately fill in the missing values, and then DBO-LSTM is used to predict RUL. The experimental results show that when there are missing values in the data, the prediction error using AS-GAIN-LSTM is only slightly improved compared with the prediction error using DBO-LSTM for complete data. AS-GAIN-LSTM gives full play to its filling ability and prediction ability, making it more excellent in dealing with such time series prediction problems with missing values.

Author Contributions: Conceptualization, X.L. (Xuejun Li); Methodology, L.J.; Validation, T.Y.; Writing—original draft, X.L. (Xu Lei); Supervision, Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledgment the support of Guangdong Provincial Basic and Applied Basic Research Fund (2023A1515240083), the Guangdong Provincial Key Construction Discipline Research Capability Improvement Project (2022ZDJS035), and the Guangdong Provincial University Innovation Team Project (2023KCXTD031). Our sincere gratitude goes to the reviewers for their insightful and constructive feedback, which has significantly enhanced the quality of this paper.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stetco, A.; Dinmohammadi, F.; Zhao, X.; Robu, V.; Flynn, D.; Barnes, M.; Keane, J.; Nenadic, G. Machine learning methods for wind turbine condition monitoring. A review. *Renew. Energy* **2019**, *133*, 620–635. [CrossRef]
2. Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mech. Syst. Signal Process.* **2018**, *104*, 799–834. [CrossRef]
3. Djeziri, M.; Benmoussa, S.; Sanchez, R. Hybrid method for remaining useful life prediction in wind turbine systems. *Renew. Energy* **2018**, *116*, 173–187. [CrossRef]
4. Lei, Y.; Li, N.; Gontarz, S.; Lin, J.; Radkowski, S.; Dybala, J. A model-based method for remaining useful life prediction of machinery. *IEEE Trans. Reliab.* **2016**, *65*, 1314–1326. [CrossRef]
5. Wang, Y.; Peng, Y.; Zi, Y.; Jin, X.; Tsui, K.-L. A two-stage data-driven-based prognostic Approach for bearing degradation problem. *IEEE Trans. Ind. Inform.* **2016**, *12*, 924–932. [CrossRef]
6. Su, X.; Wang, S.; Pecht, M.; Zhao, L.; Ye, Z. Interacting multiple model particle filter for prognostics of lithium-ion batteries. *Microelectron. Reliab.* **2017**, *70*, 59–69. [CrossRef]
7. Yang, T.; Li, G.; Li, K.; Li, X.; Han, Q. The LPST-Net: A new deep interval health monitoring and prediction framework for bearing-rotor systems under complex operating conditions. *Adv. Eng. Inform.* **2024**, *62*, 102558. [CrossRef]
8. Yang, B.; Liu, R.; Zio, E. Remaining useful life prediction based on a double-convolutional neural network architecture. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9521–9530. [CrossRef]
9. Bai, R.; Noman, K.; Feng, K.; Peng, Z.; Li, Y. A two-phase-based deep neural network for simultaneous health monitoring and prediction of rolling bearings. *Reliab. Eng. Syst. Saf.* **2023**, *238*, 109428. [CrossRef]
10. Singh, J.; Azamfar, M.; Li, F.; Lee, J. A Systematic Review of Machine Learning Algorithms for Prognostics and Health Management of Rolling Element Bearings: Fundamentals, Concepts and Applications. *Meas. Sci. Technol.* **2021**, *32*, 012001. [CrossRef]
11. Mahmood, F.H.; Kadhim, H.T.; Resen, A.K.; Shaban, A.H. Bearing failure detection of micro wind turbine via power spectral density analysis for stator current signals spectrum. *AIP Conf. Proc.* **2018**, *1968*, 020018.
12. Guo, L.; Li, N.; Jia, F.; Lei, Y.; Lin, J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing* **2017**, *240*, 98–109. [CrossRef]
13. Chen, Y.; Peng, G.; Zhu, Z.; Li, S. A novel deep learning method based on attention mechanism for bearing remaining useful life prediction. *Appl. Soft Comput.* **2020**, *86*, 105919. [CrossRef]
14. Xiang, S.; Li, P.; Huang, Y.; Luo, J.; Qin, Y. Single gated RNN with differential weighted information storage mechanism and its application to machine RUL prediction. *Reliab. Eng. Syst. Saf.* **2023**, *242*, 109741. [CrossRef]
15. Fu, C.; Gao, C.; Zhang, W. RUL Prediction for Piezoelectric Vibration Sensors Based on Digital-Twin and LSTM Network. *Mathematics* **2024**, *12*, 1229. [CrossRef]
16. Li, Y.; Chen, Z.; Hu, C.; Zhao, X. Bearing remaining useful life prediction with an improved CNN-LSTM network using an artificial gorilla troop optimization algorithm. *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.* **2024**. [CrossRef]
17. Liu, J.; Pan, C.; Lei, F.; Hu, D.; Zuo, H. Fault prediction of bearings based on LSTM and statistical process analysis. *Reliab. Eng. Syst. Saf.* **2021**, *214*, 107646. [CrossRef]
18. Magadán, L.; Granda, J.C.; Suárez, F. Robust prediction of remaining useful lifetime of bearings using deep learning. *Eng. Appl. Artif. Intell.* **2024**, *130*, 107690. [CrossRef]
19. Li, G.; Wang, Y.; Xu, C.; Wang, J.; Fang, X.; Xiong, C. BO-STA-LSTM: Building energy prediction based on a Bayesian Optimized Spatial-Temporal Attention enhanced LSTM method. *Dev. Built Environ.* **2024**, *18*, 100465. [CrossRef]
20. Li, Z.; Du, J.; Zhu, W.; Wang, B.; Wang, Q.; Sun, B. Regression predictive modeling of high-speed motorized spindle using POA-LSTM. *Case Stud. Therm. Eng.* **2024**, *54*, 104053. [CrossRef]
21. Jin, X.; Sun, Y.; Que, Z.; Wang, Y.; Chow, T.W. Anomaly Detection and Fault Prognosis for Bearings. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 2046–2054. [CrossRef]
22. Rădulescu, M.; Radulescu, C.Z. Mean-Variance Models with Missing Data. *Stud. Inform. Control.* **2013**, *22*, 299–306. [CrossRef]

23. Yang, K.; Li, J.; Wang, C. *Missing Values Estimation in Microarray Data with Partial Least Squares Regression*; Springer: Berlin, Germany, 2006; pp. 662–669.
24. Han, H.; Sun, M.; Wu, X.; Li, F. Double-cycle weighted imputation method for wastewater treatment process data with multiple missing patterns. *Sci. China Technol. Sci.* **2022**, *65*, 2967–2978. [CrossRef]
25. Liu, W.; Ren, C.; Xu, Y. PV Generation Forecasting With Missing Input Data: A Super-Resolution Perception Approach. *IEEE Trans. Sustain. Energy* **2021**, *12*, 1493–1496. [CrossRef]
26. Hart, E.; Turnbull, A.; Feuchtwang, J.; McMillan, D.; Golysheva, E.; Elliott, R. Wind turbine main-bearing loading and wind field characteristics. *Wind. Energy* **2019**, *22*, 1534–1547. [CrossRef]
27. Yoon, J.; Jordon, J.; Schaar, M. GAIN: Missing Data Imputation using Generative Adversarial Nets. *arXiv* **2018**, arXiv:1806.02920.
28. Landi, F.; Baraldi, L.; Cornia, M.; Cucchiara, R. Working Memory Connections for LSTM. *Neural Netw.* **2021**, *144*, 334–341. [CrossRef]
29. Xue, J.; Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* **2023**, *79*, 7305–7336. [CrossRef]
30. Li, X.; Zhang, W.; Ding, Q. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliab. Eng. Syst. Saf.* **2019**, *182*, 208–218. [CrossRef]
31. Wang, B.; Lei, Y.; Li, N.; Li, N. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Trans. Reliab.* **2020**, *69*, 401–412. [CrossRef]
32. Yang, N.; Stumberg, L.V.; Wang, R.; Cremers, D. D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1278–1289.
33. Camci, F.; Medjaher, K.; Zerhouni, N.; Nectoux, P. Feature evaluation for effective bearing prognostics. *Qual. Reliab. Eng. Int.* **2013**, *29*, 477–486. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Prediction of NOx Emissions in Thermal Power Plants Using a Dynamic Soft Sensor Based on Random Forest and Just-in-Time Learning Methods

Kaixun He * and Haixiao Ding

College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China; dinghx2023@163.com
* Correspondence: kaixunhe@sdust.edu.cn

Abstract: Combustion optimization is an effective way to improve the efficiency of thermal power generation and reduce carbon and NOx emissions. Real-time and precise NOx emission prediction is the basis for combustion optimization control of thermal power plants. To construct an accurate NOx concentration prediction model, a novel just-in-time learning (JITL) method based on random forest (RF) is proposed in the present work. With this method, first, an improved permutation importance algorithm is proposed to extract important variables. In addition, a similarity index that incorporates temporal and spatial measures is defined to select a local training set representative of the process data. Moreover, considering the influence of model parameters on prediction performance under different working conditions, a process monitoring method based on a moving window (MW) is used to monitor the change in working conditions and guide online updating. The experimental results show that the proposed method has excellent prediction accuracy, with a coefficient of determination of 0.9319, a root-mean-square error of 3.6960 mg/m^3, and an average absolute error of 2.7718 mg/m^3 on the test set, making it superior to other traditional methods.

Keywords: just-in-time learning; sample selection; random forest; thermal power plant; soft sensor; NOx

1. Introduction

With the increasing attention paid to environmental protection, determining how to reduce carbon and toxic gas emissions in flue gas while improving boiler efficiency is an important and urgent problem to address in thermal power plants [1–4]. Over recent decades, the reliability of power grid supply has predominantly relied on thermal power generation, with coal-fired thermal power plants contributing approximately 60% of national electricity in China [5]. Furthermore, the integration of intermittent renewable energy sources like wind power into the grid hinges upon the peak-regulating capabilities of thermal power units [6]. Consequently, thermal power generation is poised to maintain its pivotal role over the long term. Coal combustion produces substantial quantities of toxic gases, notably NOx, a primary contributor to atmospheric pollution [7,8]. As stringent environmental regulations continue to be enacted, coal-fired power plants face progressively stringent NOx emission standards for flue gases. Currently, China's laws stipulate that NOx emissions in flue gas should not exceed 50 mg/m^3 [9], which poses a great challenge to the optimal control of power plant boilers and flue gas systems.

Accurate real-time detection of NOx concentration directly affects the amount of ammonia injected into the denitrification device, which is crucial for improving the efficiency of the selective catalytic reduction (SCR) and is the basis for boiler combustion optimization control [10]. The main chemical reaction equations of the denitration process are shown in Equations (1) and (2) [11].

$$NO + NO_2 + 2\,NH_3 \rightarrow 2\,N_2 + 3\,H_2O \quad (1)$$

$$4\,NO + 4\,NH_3 + O_2 \rightarrow 4\,N_2 + 6\,H_2O \quad (2)$$

As illustrated, insufficient ammonia injection can cause ammonia or NOx to escape, whereas excessive injection results in resource wastage. To deal with such problems, CEMS is widely used to track and monitor NOx concentration. Unfortunately, CEMS is greatly affected by environmental disturbances, resulting in inadequate detection accuracy and considerable lag [12]. In view of this challenge, the construction of a high-precision real-time NOx prediction model is a necessary supplement to existing detection methods [13]. Prediction strategies for NOx concentration mainly include model-based and data-based methods. The first method has better interpretation ability but requires a large amount of expert knowledge. The combustion process of thermal power plant boilers is highly complicated, making it difficult to establish accurate models for describing the process [14]. With the rapid development and application of distributed control systems (DCSs) in recent years, large amounts of industrial process data can be collected. As a result, data-driven methods have gradually emerged as a viable alternative. Compared with model-based methods, data-driven strategies bypass the need to solve complex conservation equations, resulting in faster and more reliable model responses; hence, they have received widespread attention. For example, Wang et al. [15] employed deep belief networks to extract data features and constructed networks consisting of an extreme learning machine (ELM), a backpropagation neural network (BPNN), and a radial basis function to predict NOx concentration. Jacob et al. [16] developed a combustion optimization system based on neural networks and particle swarm optimization, which effectively reduced the emission of NOx from power plants. Yang et al. [17] established a NOx prediction model based on a long short-term memory (LSTM) neural network, which exhibited better performance than least-squares support vector machine (LSSVM) and recurrent neural network models. Yuan et al. [18] proposed a NOx emission prediction method using linear regression as the metamodel and adopted BPNN, support vector regression, and decision tree as the basic models. The proposed stacked-generalization ensemble method demonstrated strong robustness and generalization capability. Li et al. [19] proposed a novel model architecture composed of a convolutional neural network and an effective channel attention module, which demonstrated good performance in predicting NOx emissions. Timo Korpela et al. [20] compared the performance of three nonlinear methods—multilayer perceptron, support vector regression, and fuzzy inference system—in predicting NOx concentration in natural gas boilers. Xie et al. [21] applied the sequence-to-sequence structure from the field of natural language processing to the LSTM model, achieving simultaneous prediction of NOx emissions at multiple time points. In addition, to reduce the impact of redundant variables on the performance of the prediction model, many variable selection methods have been incorporated. Xing et al. [22] utilized partial least squares (PLS) for variable selection and established an extreme gradient boosting ensemble model to predict the NOx emission concentration of coal-fired boilers. Tang et al. [23] employed mutual information combined with autoencoders and an ELM to deeply explore the relationship between NOx emission concentration and features. Wang et al. [24] utilized random forest (RF) to calculate the importance of variables and select the important process variables. Based on neural networks, Zhang et al. [25] employed the mean impact value to select variables and achieved multiobjective prediction of boiler thermal efficiency and NOx emission coupling. Tang et al. [12] utilized the LASSO and relief feature selection algorithms to select the important variables. They proposed an error correction strategy and established an ELM model, which can accurately predict NOx concentration at the boiler outlet. LSSVM [26,27], convolutional neural networks [28–32], and ensemble learning methods [33–36] have also been widely used by scholars in NOx concentration prediction.

Although the above studies achieved enhanced results under certain scenarios, they hardly meet the need for timely updates in online applications [3,37]. To address this issue, Lv et al. [38] improved LSSVM and updated the model based on an incremental strategy. Li et al. [14] proposed a variable exponentially weighted MWPLS method, which can be adaptively updated by adjusting the window size. Lara F. A. [39] used the T^2 and Q

statistics of principal component analysis (PCA) to determine whether a model needs to be recalibrated. Lv et al. [40] proposed an adaptive strategy by updating the operating dataset when the model's performance deteriorates. The above methods have achieved good results in addressing specific problems. However, the model parameters are difficult to adjust. Consequently, they are difficult to directly apply to the real-time prediction of NOx concentration in thermal power plants. Recently, just-in-time learning (JITL) [41,42] has seen widespread adoption due to its intrinsic characteristics favoring online implementation. Instead of building a global model, JITL creates a local model based on the similarity between input and output samples in real time. By using the current measured input data, similar samples in the database are collected for modeling. Considering the advantages of JITL, an improved scheme based on JITL for online prediction of NOx emissions is proposed in this work. With the proposed strategy, a supervised similarity distance measurement method is defined to adaptively select important training samples from the original dataset. RF regression is adopted to establish a prediction model, and an active updating strategy is proposed to maintain the model online. Moreover, to establish a robust model, a variable selection method is proposed that enables the robust selection of important variables. The industrial application results show that the presented strategy can provide good prediction accuracy and is suitable for long-term industrial applications.

This paper is organized as follows. Preliminaries about JITL and RF are provided in Section 2. Section 3 describes the target boiler in this work. Section 4 presents the proposed modeling method. Section 5 discusses the experiments and results obtained using real-world data. Finally, the conclusions are provided in Section 6.

2. Preliminaries

The key to predicting NOx concentration is to establish a learner $f(\cdot)$ with historical data. Subsequently, the auxiliary variable X_q is substituted into the formula $\widehat{y} = f(X_q) + \varepsilon_q$ to obtain the prediction value. In this work, JITL and RF are combined to develop a local prediction model. To elucidate the proposed approach, this section outlines the foundational principles of JITL and RF.

2.1. JITL Method

JITL is a dynamic modeling framework in which all historical data are stored in a database, and a model is built in real time by searching the database for the samples most relevant to the query sample X_q by a certain similarity index. After prediction, the established model is discarded. For the regression problem, the most common metric used to measure the similarity between X_q and the historical sample X_i is the Mahalanobis distance, which is defined as follows [43]:

$$d_M(X_i, X_q) = \sqrt{(X_i - X_q)\sum{}^{-1}(X_i - X_q)^{-1}} \tag{3}$$

where $X \in R^{n\times m} = \{X_1, \ldots, X_i, \ldots X_n\}$, $i = 1, 2, \ldots, n$ denotes the sample number in the historical database and $\sum$ is the covariance matrix of X. Based on the Mahalanobis distance values, the similarity between X_q and each sample in the historical data can be calculated using the Gaussian kernel function, as follows:

$$S_{i,q} = \exp(-\frac{d_M^2(X_i, X_q)}{2\sigma^2}) \tag{4}$$

where σ is the kernel width. In accordance with $S_{i,q}$, the first l samples with high similarity are selected from the historical data to construct the local training set. Then, the output prediction of the query sample X_q is given by:

$$\widehat{y}_q = f(X_q, \Theta) \tag{5}$$

where Θ is the hyperparameter of the model $f(\cdot)$.

2.2. RF Regression

RF is an ensemble learning algorithm capable of generating numerous decision trees that serve as regression learners. The final prediction result of RF is derived from the mean value of all trees. The structure of RF is shown in Figure 1. Classification and regression tree (CART) is commonly employed as a decision tree for RF. Given a single output dataset $D = \{X_{train}, Y_{train}\}$, where $X_{train} = (X_1, X_2, \ldots, X_N) \in R^{N \times M}$, $Y_{train} = (y_1, y_2, \ldots, y_N) \in R^{N \times 1}$, and N is the number of samples, along with the decision tree algorithm Γ and the number of base learners T, the steps for constructing a regression model using RF are as follows:

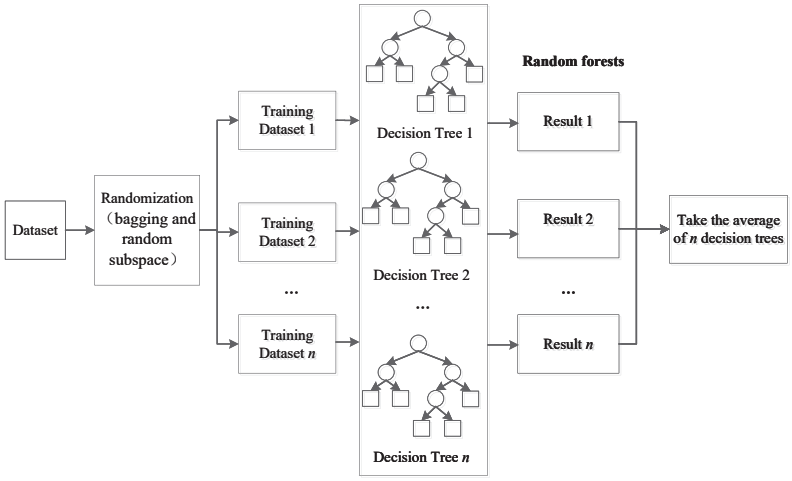


Figure 1. Schematic of RF regression modeling.

Step 1: For each decision tree t, n samples are randomly collected from D to construct a subtraining set D_t using the bagging method, where $t \in [1, T]$.

Step 2: The tth learner $h_t = \Gamma(D_t)$ is trained using D_t. During the training process, for each node, $m(m < M)$ features are selected. Then, the optimal partition is selected from these m features to divide the molecular tree.

Step 3: During the formation of the decision tree, each node should be split in accordance with Step 2. The decision tree is trained with this subset until it is no longer possible to split, and the tree is not pruned.

Step 4: In accordance with Steps 1–3, a series of decision trees are established until T trees are trained.

During the application stage, the test samples are sent to each decision tree for regression prediction, and the prediction results of all decision trees for the same sample are counted. The average value of all results is used as the final predicted value, which can be expressed as follows:

$$H = \frac{1}{T} \sum_{t=1}^{T} h_t(X) \tag{6}$$

The RF algorithm is well suited for modeling complex processes due to its effective utilization of randomness, which reduces inter-tree correlations. Therefore, in this work, RF regression is adopted as the basic modeling method.

3. Proposed Method

Generally, the representativeness of auxiliary variables determines the upper limit of model performance. Selecting important variables helps prevent dimensionality disasters and mitigate overfitting. With RF, important variables are typically identified using permutation importance. The basic idea of permutation importance is as follows: for each variable x^j, $j \in [1, M]$. The sequence of variables x^j in out-of-bag (OOB) data is disrupted, and the corresponding relationship between auxiliary and output variables is broken. Subsequently, CART performs regression predictions on OOB data both before and after shuffling, calcu-

lating the mean square errors (MSEs) for each decision tree, which collectively determine the variable importance index x^j.

Given an RF with T decision trees $H = \{h_1, h_2, \ldots, h_t, \ldots, h_T\}$, where $t \in [1, T]$, the variable importance of x^j is defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2 \tag{7}$$

$$\bar{D}_j = \frac{1}{T}\sum_{t=1}^{T}\left(MSE_t^{OOB} - MSE_{t,v}^{OOB}\right) \tag{8}$$

where MSE_t^{OOB} and $MSE_{t,v}^{OOB}$ are the $MSEs$ of the OOB data on tree t before and after interference, respectively, and $\bar{D}_j$ is the variable importance of x^j. The permutation importance algorithm of RF is presented in Algorithm 1.

Algorithm 1: The permutation importance algorithm of RF

Input: $\{X_{train}, Y_{train}\}$
Output: VI_{RF}^j
for $j = 1 : m$ **do**
  for $t = 1 : T$ **do**
    Calculate the mean square error MSE_t using $MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2$
    For each OOB_t dataset, random interactions are performed on the features in the x^j dimension of the sample;
    Calculate the mean square error MSE_t^j
    Calculate vi_t^j using $vi_t^j = MSE_t^j - MSE_t$
  end
  Calculate the total influence factor VI_{RF}^j using $VI_{RF}^j = \frac{1}{T}\sum_{t=1}^{T} vi_t^j$
end

Based on the above permutation importance algorithm, determining the number of crucial variables to retain is not straightforward. In practice, threshold values are frequently empirically determined to facilitate variable selection, rendering the process notably subjective. To deal with this problem, this work proposes an iterative selection strategy, which determines the variable set to be retained by counting the frequency of selected variables in multiple runs. The specific steps of the improved variable selection method based on permutation importance are as follows:

Step 1: A regression model is established using each decision tree h_t, the value of the corresponding OOB data is predicted, and MSE_t is output.

Step 2: For each OOB_t set, the order of variable $x^j, j \in [1, M]$, is shuffled to obtain a new set; then, MSE_t^j is calculated using the model established in Step 1.

Step 3: The prediction influence factor of x^j on tree h_t is calculated using the following equation:

$$vi_t^j = MSE_t^j - MSE_t \tag{9}$$

Step 4: All decision trees are traversed, and Steps 1–3 are repeated. Afterward, the influence factors of x^j on all decision trees are obtained.

Step 5: The total influence factor of x^j on the RF is calculated as follows:

$$VI_{RF}^j = \frac{1}{T}\sum_{t=1}^{T} vi_t^j \tag{10}$$

Step 6: All variables in $\mathbf{X}_{train}$ are traversed, and Steps 1–5 are repeated to acquire the influence factors VI_{RF} of all variables.

Step 7: The variables are sorted in accordance with the value of VI_{RF}^{j}.

Step 8: Steps 1–7 are repeated ℓ times, and the variables with the highest frequency in the top K are saved.

Step 9: The variable with the lowest frequency of 10% is eliminated each time, and the corresponding OOB error after removing the variables is calculated.

Step 10: The subset of variables corresponding to the minimum OOB error is selected.

3.1. Strategy for Local Training Sample Selection

In JITL, a crucial initial task involves identifying local training samples that closely resemble the query sample. To implement this operation, the distance d_{M_i} between the query sample X_q and the ith historical sample X_i is defined first, and then local training samples are selected based on d_{M_i}. As mentioned above, the Mahalanobis distance is generally used to define the similarity for regression modeling. The Mahalanobis distance considers the spatial characteristics of auxiliary variables but ignores the relationship between dependent variables.

Due to the strong time sequence characteristics inherent in thermal power plant process data, a more representative similarity index can be constructed by introducing the dependent variable information of the adjacent time-domain samples. Accordingly, we propose an improved similarity index by simultaneously considering the temporal and spatial characteristics of the independent and dependent variables.

Using the proposed scheme, the Mahalanobis distance d_{M_i} between X_i and X_q is calculated. The similarity index of all samples in the historical dataset is obtained. The average value of the NOx concentration of the sample in the preceding Mw time window of X_q is calculated. The distance of the dependent variable is calculated as follows:

$$d_y(y_i, \bar{y}_{local}) = \sqrt{(y_i - \bar{y}_{local})^2} \tag{11}$$

where y_i is the concentration of NOx in the historical data and $\bar{y}_{local}$ is the average concentration within the Mw time scale.

In this way, the distance d_M defined by the independent variable and the distance d_y defined by the temporal data and dependent variables are obtained. The final similarity can be calculated through the following equations:

$$d_{sim}(X_i, X_q) = \lambda d_M + (1 - \lambda) d_y \tag{12}$$

$$S_{i,q} = \exp(-\frac{d_{sim}^2(X_i, X_q)}{2\sigma^2}) \tag{13}$$

where $0 \leq \lambda \leq 1$ is a conversion factor; when $\lambda = 1$, the similarity degenerates to a method based solely on the Mahalanobis distance.

In accordance with the above method, L samples with high similarity can be selected from the historical data to construct a local training set D_l. Given that the local training set selected using the above method still has high leverage points, the existence of such samples reduces the performance of the constructed prediction model. At the end of sample selection, we use an F distribution to eliminate sample points with low confidence. The specific operations are described below.

The mean value of the independent variables in D_l is calculated using Equation (14). If $d_M(X_l, u) > F_\alpha(M, L)$, then the sample $\{X_l, Y_l\}$ is deleted from D_l, where α is the quantile; M and L are the number of independent variables and the number of samples in D_l, respectively. For ease of description, the final selected local sample set is still represented by D_l.

$$u = mean(X_{local}) \tag{14}$$

The pseudocode for local training sample selection based on the proposed similarity definition method is shown in Algorithm 2.

Algorithm 2: The algorithm for local training sample selection

Input: $X_q, \{X_{train}, Y_{train}\}$
Output: $\{X_{local}, Y_{local}\}$
if $time < Mw$ **then**
  $\bar{y}_{local} = \sum_{ii=1}^{ii=time} y_{ii}$
else
  $\bar{y}_{local} = \sum_{ii=time-Mw}^{time-1} y_{ii}$
end
for $i = 1 : n$ **do**
  Calculate $d_{M_i}(X_i, X_q)$
  $d_{sim}(X_i, X_q) = \lambda d_M + (1 - \lambda) d_y$
end
Sort $S_{i,q}$ in descending order then
Select the first lth samples $D_l = \{X_{local,} Y_{local}\}$ from D
for $l = 1 : L$ **do**
  Calculate $d_M(X_l, u)$, where $u = mean(X_{local})$
end
if $d_M(X_l, u) > F_\alpha(M, L)$ **then**
  Delete $\{X_l, Y_l\}$
end

3.2. Process Condition Monitoring and Parameter Updating Strategy

The thermal power generation process has typical multicondition characteristics. Under the same working conditions, a high-performance prediction model can be constructed using JITL. However, the local training set constructed according to Equation (12) may drift in its representation of the process conditions after switching conditions. To address this problem, MWPCA is adopted to monitor the production process. Then, λ is updated timely in accordance with the monitoring results. The specific process of the algorithm is described below.

The moving window length is denoted as L, and the corresponding data matrix within the window at time t is denoted as $MatrixI$: $X_t = (x_{t-L+1}, x_{t-L+2}, \ldots, x_{t-1}, x_t)^T \in \mathrm{R}^{L \times M}$. At time $t + 1$, the corresponding data matrix within the window is denoted as $MatrixIII$: $X_{t+1} = (x_{t-L+2}, x_{t-L+3}, \ldots, x_t, x_{t+1})^T \in \mathrm{R}^{L \times M}$, and the data matrix corresponding to the transition window is denoted as $MatrixII$: $\tilde{X}_{t,t+1} = (x_{t-L+2}, x_{t-L+3}, \ldots, x_{t-1}, x_t)^T \in \mathrm{R}^{(L-1) \times M}$. The specific steps are as follows:

Step 1: Transition from $MatrixI$ to $MatrixII$:

$\mu_t \in \mathrm{R}^{M \times 1}$ and $C_t \in \mathrm{R}^{M \times M}$ are set as the mean vector and covariance matrix of $MatrixI$, respectively. The mean vector $\tilde{\mu}_{t,t+1}$ and standard deviation $\delta^i_{t,t+1} (i = 1, 2, \ldots, M)$ of the variables for $MatrixII$ can be determined using Equations (15) and (16), respectively.

$$\tilde{\mu}_{t,t+1} = \frac{L}{L-1}\mu_t - \frac{1}{L-1}x_{t-L+1} \tag{15}$$

$$(\delta^i_{t,t+1})^2 = \frac{L-1}{L-2}(\delta^i_t)^2 - \frac{L-1}{L-2}(\mu^i_t - \tilde{\mu}^i_{t,t+1})^2 - \frac{1}{L-2} \parallel x^i{}_{t-L+1} - \mu^i_t \parallel^2 \tag{16}$$

$$\sum\nolimits_{t,t+1} = diag(\delta^1_{t,t+1}, \delta^2_{t,t+1}, \ldots, \delta^M_{t,t+1}) \tag{17}$$

The covariance matrix $\tilde{C}_{t,t+1} \in R^{M\times M}$ of *MatrixII* can be derived from the above formula and is shown below:

$$\tilde{C}_{t,t+1} = \frac{L-1}{L-2}(C_t - \frac{1}{L-1}x^T{}_{t-L+1}x_{t-L+1} - \sum_t^{-1}(\mu_t - \tilde{\mu}_{t,t+1})(\mu_t - \tilde{\mu}_{t,t+1})^T\sum_t^{-1}) \quad (18)$$

where $\sum_t = diag(\delta_t^1, \delta_t^2, \ldots, \delta_t^M)$ is the diagonal matrix composed of the standard deviations of each variable in *MatrixI*.

Step 2: Transition from *MatrixII* to *MatrixIII*:

Similar to Step 1, here, the mean vector μ_{t+1} of *MatrixIII* and the standard deviation $\delta_{t+1}^i (i = 1, 2, \ldots, M)$ of each variable can be recursively calculated.

$$\mu_{t+1} = \frac{L-1}{L}\tilde{\mu}_{t,t+1} + \frac{1}{L}x_{k+1} \quad (19)$$

$$(\delta_{t+1}^i)^2 = \frac{L-2}{L-1}(\delta_{t,t+1}^i)^2 + (\mu_{t+1}^i - \tilde{\mu}_{t,t+1}^i)^2 - \frac{1}{L-1} \parallel x^i{}_{t+1} - \mu_{t+1}^i\|^2 \quad (20)$$

$$\sum_{t+1} = diag(\delta_{t+1}^1, \delta_{t+1}^2, \ldots, \delta_{t+1}^M) \quad (21)$$

The new samples are standardized as follows:

$$x_{t+1} = \sum_{t+1}^{-1}(x_{t+1} - \mu_{t+1}) \quad (22)$$

The covariance matrix $C_{t+1} \in R^{M\times M}$ of *MatrixIII* can be derived from the above formula.

$$C_{t+1} = \frac{L-2}{L-1}\tilde{C}_{t,t+1} + \frac{1}{L-1}x^T{}_{t+1}x_{t+1} + \sum_{t+1}^{-1}(\mu_{t+1} - \tilde{\mu}_{t,t+1})(\mu_{t+1} - \tilde{\mu}_{t,t+1})^T\sum_{t+1}^{-1} \quad (23)$$

Substituting Equation (18) into Equation (23) yields the expression for transitioning from *MatrixI* to *MatrixIII* as follows:

$$\begin{aligned} C_{t+1} = C_t &- \sum_t^{-1}(\mu_t - \tilde{\mu}_{t,t+1})(\mu_t - \tilde{\mu}_{t,t+1})^T\sum_t^{-1} \\ &- \frac{1}{L-1}x_{t-L+1}x^T{}_{t-L+1} + \frac{1}{L-1}x_{t+1}x_{t+1}^T \\ &+ \sum_{t+1}^{-1}(\mu_{t+1} - \tilde{\mu}_{t,t+1})(\mu_{t+1} - \tilde{\mu}_{t,t+1})^T\sum_{t+1}^{-1} \end{aligned} \quad (24)$$

After the covariance matrix of the data in the new window is obtained, the corresponding principal component model can be obtained through singular value decomposition of the covariance matrix C_{t+1}. The statistic T^2 and the control limit for T^2 with confidence level α can be determined using Equations (25) and (26), respectively.

$$T^2 = x^T P\Lambda^{-1}P^T x \quad (25)$$

$$T_\alpha^2 = \frac{A(L^2-1)}{L(L-A)}F_{A,L-a;\alpha} \quad (26)$$

where $\Lambda \in R^{A\times A}$ is the diagonal matrix composed of the first A eigenvalues of the covariance matrix C_{t+1} and $F_{A,L-a;\alpha}$ is the critical value of the F distribution with A and $L-A$ degrees of freedom and a significance level of α.

When the T^2 corresponding to the query sample is greater than the monitoring threshold, a working condition switch has occurred. If the monitoring statistic of the query sample does not exceed the monitoring threshold, the historical data in the moving window have characterized the current process well. In this case, λ can take a slightly smaller value, and vice versa.

3.3. Overall Flow of the Proposed Modeling Strategy

Following the aforementioned variable selection and local training sample selection methods, the overall flow of the NOx prediction model based on JITL is presented in Figure 2, and the corresponding pseudocode is shown in Algorithm 3.

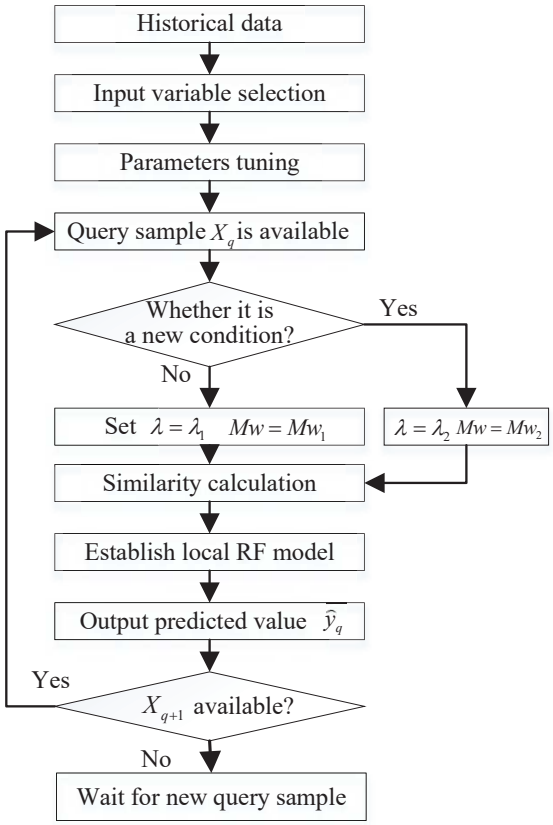


Figure 2. Basic concept of the proposed JIT-RF modeling framework.

Algorithm 3: Just-in-time RF

Input: X_q

Output: $\widehat{y}_q$

Historical data collection and parameter tuning.
Construct training dataset $D = \{X_{train} \in R^{n \times m}, Y_{train} \in R^{n \times 1}\}$
Tune the parameters Mw, λ, $nLeaf$, $nTree$ based on D.
Obtain query sample X_q from the process online.
Monitor the process conditions using MW-PCA;
if *alarm is triggered by MW-PCA* **then**
  $Mw = Mw_2, \lambda = \lambda_2$
else
  $Mw = Mw_1, \lambda = \lambda_1$
end
Similarity calculation
for $i = 1 : n$ **do**
  Calculate the similarity between X_q and $X_i \in X_{train}$
end
Select local training data using Algorithm 2.
RF modeling
The prediction model is established with the collected local training dataset.
Then, $\widehat{y}_q$ is provided.

4. Boiler System and Data Preparation

4.1. Description of the Boiler System

In this work, the direct current boiler of a 1030 MW ultra-supercritical coal-fired unit is adopted as the target object. The object is a π-type boiler, featuring balanced ventilation, ultra-supercritical parameters, primary reheat, a spiral furnace, solid slag discharge, and an open-air arrangement. The pulverizing system employs a medium-speed coal mill with positive-pressure direct cooling primary air. The combustion unit adopts a front and rear wall hedge combustion mode, featuring low-NOx double adjustable air swirl burners and nozzles. The furnace dimensions are 64,500 mm in height, 33,128.7 mm in cross-sectional width, and 16,308.7 mm in depth. A flue gas-regulating baffle device is arranged at the bottom of the flue passage to distribute the flue gas, maintaining the reheat steam outlet temperature within the control load range. The flue gas is collected by the regulating baffle and then introduced into the SCR denitration device through the two tail flues. After denitration, the flue gas enters the air preheater. Figure 3 shows the overall structure of the boiler and SCR denitrification.

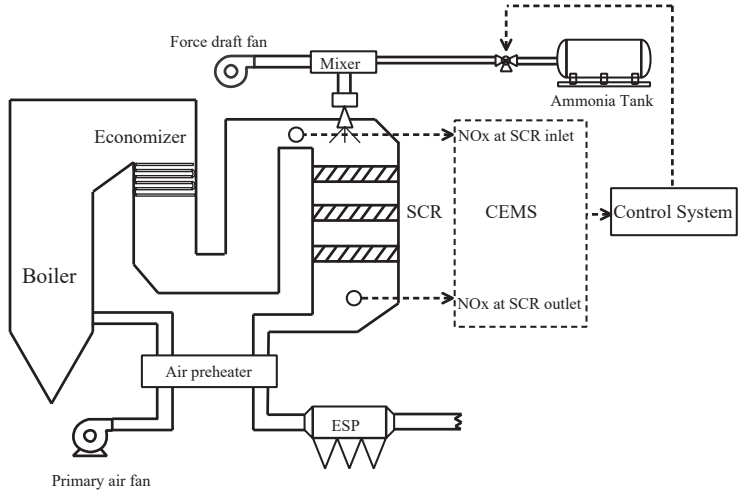


Figure 3. Process flow of SCR denitrification.

4.2. Data Description

The historical data used in this work were obtained from DCS, and a total of 5184 groups of 72 h data were collected at a sampling interval of 50 s. The abnormal samples were eliminated using the 3σ rule (Equation (27)), and 5174 groups of data were retained.

$$\bar{y} - 3\sigma \leq y \leq \bar{y} + 3\sigma \tag{27}$$

where σ and $\bar{y}$ represent the standard deviation and mean value of NOx concentration, respectively. Among these data, 30% were randomly selected to construct the training set, about 60% were used as the test set, and the remaining 10% were utilized to construct the verification set. The original data contained 390 auxiliary variables, which were normalized as follows:

$$x = \frac{x - \bar{x}}{\varphi} \tag{28}$$

where $\bar{x}$ and φ represent the mean and standard deviation values, respectively.

To assess the performance of the mentioned models, three indices—root-mean-square error (RMSE), coefficient of determination (R^2), and mean absolute error (MAE)—were adopted. Generally, the smaller the values of the RMSE and MAE, the higher the accuracy

of the model. R^2 describes the explanatory ability; the closer its value to 1, the greater the explanatory ability of the model.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2} \tag{29}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \tag{30}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right| \tag{31}$$

where n is the number of test samples; y_i and $\widehat{y}_i$ represent the actual and predicted values, respectively; and $\bar{y}$ denotes the mean value.

5. Case Study

5.1. Parameter Tuning

Two parameters should be adjusted when constructing a model using RF: the number of leaf nodes, i.e., $nLeaf$, and the number of decision trees, i.e., $nTree$. In this work, these parameters are adjusted based on the OOB error on the training set. The numbers of leaf nodes and decision trees that minimize the OOB error are regarded as the optimal parameters. As illustrated in Figure 4, when $nLeaf$ is 2, the corresponding MSE curve is the lowest. After the number of decision trees is increased to more than 50, the MSE of the OOB samples hardly decreases. Therefore, in this work, the values of $nLeaf$ and $nTree$ are set to 2 and 50, respectively.

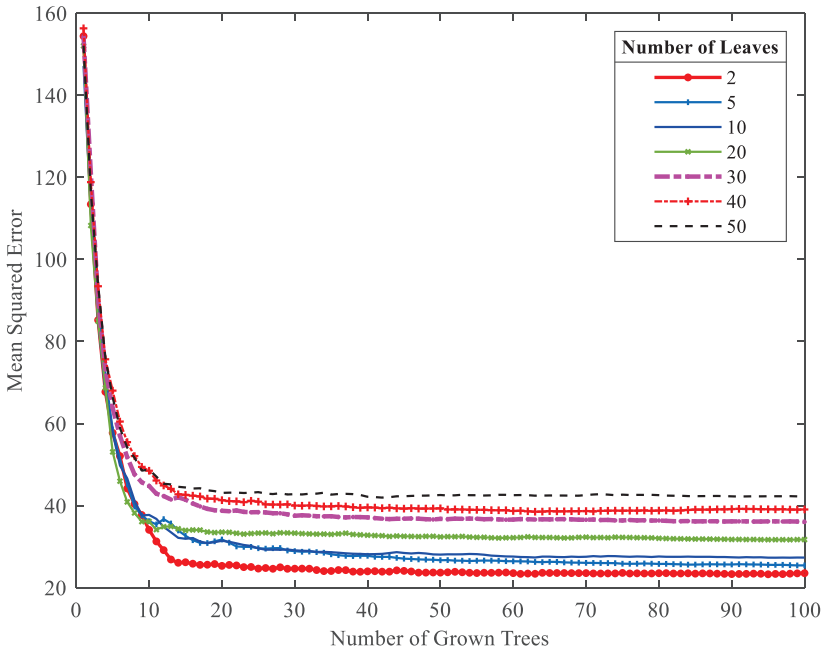


Figure 4. OOB error curves with varying values of $nLeaf$ and $nTree$.

5.2. Results and Discussion

In accordance with the variable selection method mentioned above, with $\ell = 100$ and $K = 50$, the RF model was run 100 times independently. Subsequently, variables that most frequently entered the top 50 based on their importance index were tallied. Finally, a total of 38 sets of variables were selected. Figure 5 shows the selection results.

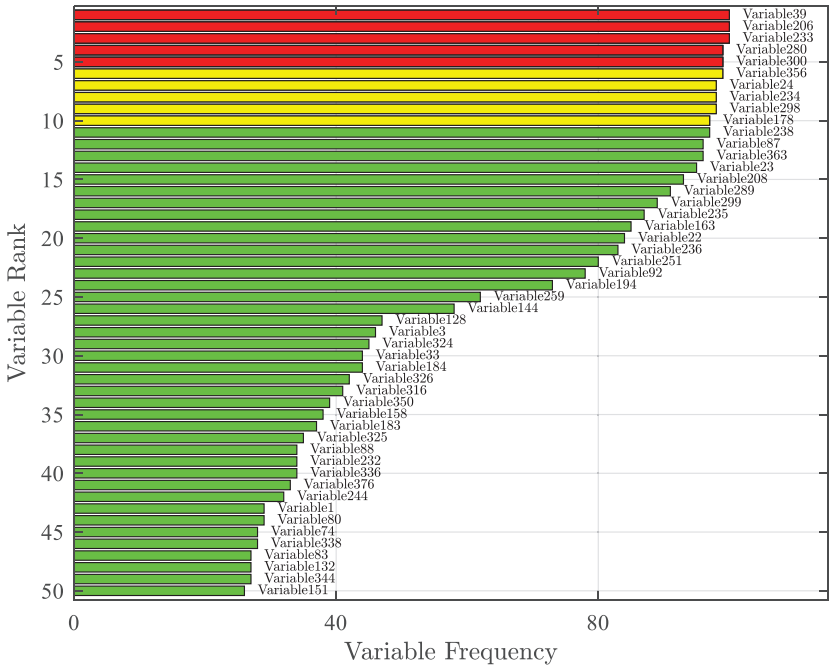


Figure 5. Variable selection results.

To improve adaptability to the multicondition process, MW-PCA was employed to determine whether the parameter λ needed to be updated. Experimentally, the window size was set to Mw = 200 and monitored for 20 consecutive times. When the T^2 control limit was exceeded 15 times or more, an alarm signal was generated. This signal indicated that the process conditions had changed, and the parameter λ had to be updated. Figure 6 illustrates the monitoring results of MWPCA on the test set (the thick red line represents the T^2 control limit for each window, the blue line represents the calculated T^2 and the red circle represents the update time of the model).

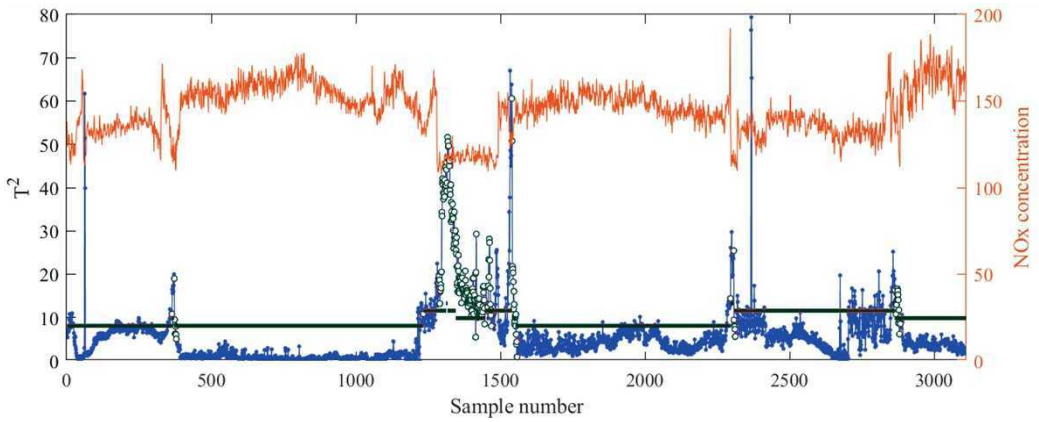


Figure 6. MW-PCA monitoring results.

As illustrated in Figure 6, a total of 228 changes were detected. The alarm occurred frequently— four times. These alarm messages were concentrated in periods of significant fluctuation in NOx concentration. The greater the fluctuation in NOx concentration, the more alarms occurred. These periods corresponded to transitional conditions before and after process changes. When the process condition was switched, $\bar{y}_{local}$ in Equation (11) was insufficient as a representative for the query sample y_q. Therefore, λ should take a slightly larger value to ensure that the similarity calculated using Equation (12) maintains good accuracy.

Figure 7a presents the errors between the measured and predicted values of NOx, predominantly concentrated within the range of ±10 mg/m^3. This indicates that the model demonstrates excellent stability and accuracy. Figure 7b intuitively shows the situation between the predicted and actual values. The model generally tracked NOx concentration variations well, albeit with large errors when the NOx concentration fluctuated significantly. These moments generally corresponded to transitional stages of changes in the operating conditions. This finding is consistent with Figure 6, where the model exhibited significant errors during changes in the operating conditions. Subsequent updates incorporating the latest operational data resulted in improved and sustained model accuracy.

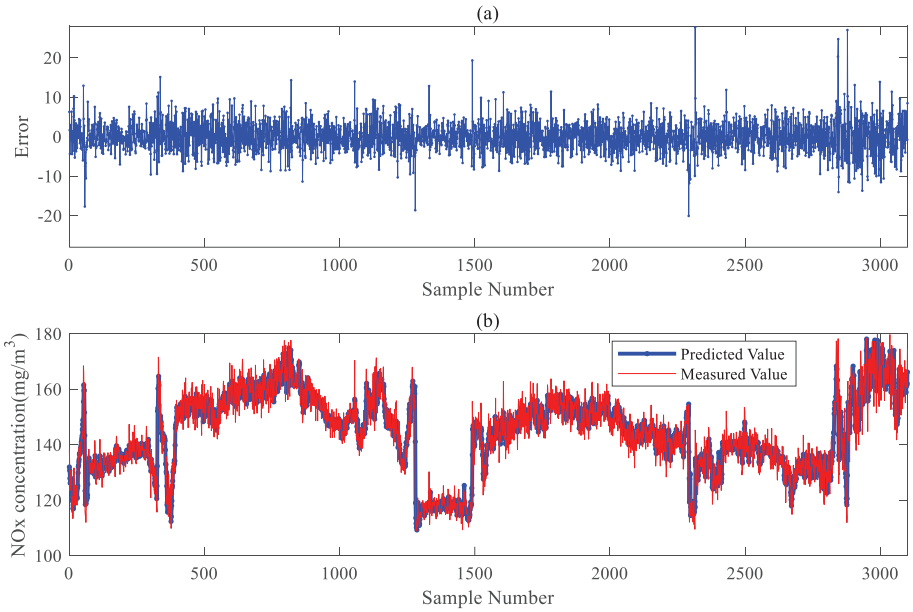


Figure 7. Prediction and error curves of the proposed method. (**a**) The prediction error of the proposed method on the test set. (**b**) The predicted results of the proposed method on test set.

Figure 8a shows the scatter plot of the predictions. The predicted and measured values are closely distributed near the perfect straight line, with small deviation and variance, demonstrating the excellent predictive performance of the model. Figure 8b indicates that the model prediction errors are primarily concentrated within the range of −5 mg/m^3 to 5 mg/m^3, with 99.71% of samples exhibiting an absolute error within 15 mg/m^3.

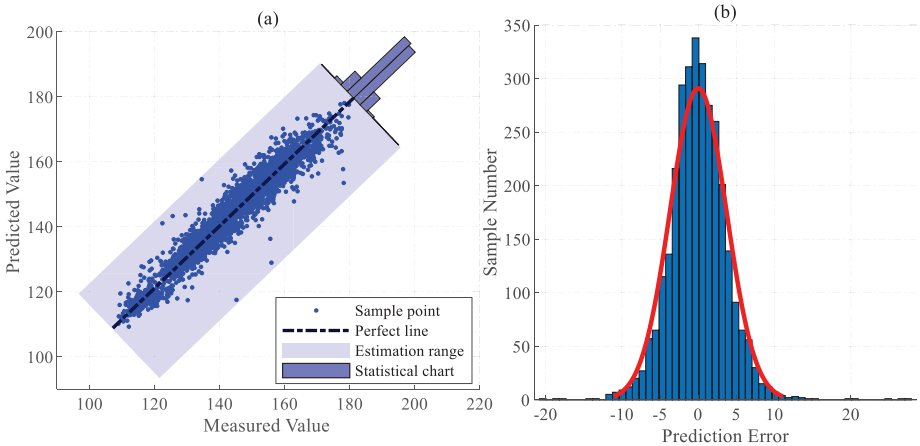


Figure 8. Scatter plot and error distribution diagram of the proposed model. (**a**) The scatter plot of actual measured and predicted values. (**b**) The distribution map of prediction error.

To validate the superiority of the proposed method, the traditional PLS and RF models, the MW-PLS and MW-RF models incorporating the MW strategy, and the JIT-PLS and JIT-

RF models with the JITL strategy were adopted as comparison methods. Table 1 presents the results of the three evaluation metrics for the six models and the proportion of samples with an absolute error within 15 mg/m^3.

Table 1. Prediction results of different models.

Method	R^2	RMSE	MAE	Proportion with Absolute Error Less than 15 mg/m^3
JIT-PLS	0.8473	5.5345	3.7936	97.29%
PLS	0.7879	6.5226	4.5792	96.23%
MW-PLS	0.6222	8.7057	6.1037	92.48%
RF	0.9197	4.0138	2.9939	99.58%
MW-RF	0.7095	7.6345	5.1912	93.94%
JIT-RF	0.9252	3.8727	2.9018	99.68%
Proposed method	0.9319	3.6960	2.7718	99.71%

The R^2, RMSE, and MAE values of the proposed model on the test set were 0.9319, 3.6960 mg/m^3, and 2.7718 mg/m^3, respectively, surpassing those of the other models in both prediction accuracy and error distribution. The predictive performance of the RF model was superior to that of the PLS model, which indicates that the RF model holds a greater advantage in predicting NOx concentration. The predictive performance of the PLS and RF models with the addition of the MW strategy was inferior to that of the traditional PLS and RF models, possibly due to information loss caused by the fixed length of MW. Specifically, the R^2, RMSE, and MAE of the PLS model on the test set were 0.7879, 6.5226 mg/m^3, and 4.5792 mg/m^3, respectively. The R^2, RMSE, and MAE of the JIT-PLS model on the test set were 0.8473, 5.5345 mg/m^3, and 3.7936 mg/m^3, respectively. A comparison of the performance of the PLS and JIT-PLS models indicates that the PLS model incorporating the JITL strategy exhibited better predictive performance compared to the traditional PLS model, which proves the effectiveness and applicability of the JITL strategy. The performance results of the JIT-RF model on the test set were 0.9252, 3.8727 mg/m^3, and 2.9018 mg/m^3. Incorporating the model update strategy into the JIT-RF model leads to further improvements in prediction performance.

Figures 9 and 10 show the error distributions of the six comparative algorithms on the test set, illustrating that the RF model exhibited more stable prediction performance compared to the PLS model, with the JIT-RF model demonstrating the highest prediction accuracy. The incorporation of the JITL strategy into both the PLS and RF models resulted in a significant reduction in prediction errors. The range of error fluctuation also significantly improved, indicating the effectiveness of the JITL strategy. Conversely, the inclusion of the MW strategy led to a notable increase in prediction errors and the fluctuation range of errors. Thus, the addition of the MW strategy results in decreased accuracy due to the loss of input information and increased instability in prediction.

Figures 11–13 present the error statistics and scatter plots. In Figure 12b, the NOx concentration predicted by the JIT-RF model and the actual measured values approximately follow the diagonal line. In contrast, Figure 13a demonstrates that the sample points predicted by the MW-PLS model deviate significantly from the perfect line. MW-PLS exhibited the worst and most unstable predictive performance, whereas the JIT-RF model had better predictive accuracy than the other models. After the addition of the MW strategy, the predictive performance of the MW-PLS and MW-RF models decreased (Figure 13), possibly due to the incomplete information in the training set caused by the MW strategy. After the JITL strategy was incorporated into the PLS and RF models, the sample points became closer to the perfect line (Figures 11b and 12b), indicating that the prediction performance of the model improved and demonstrating the advantages of the JITL strategy.

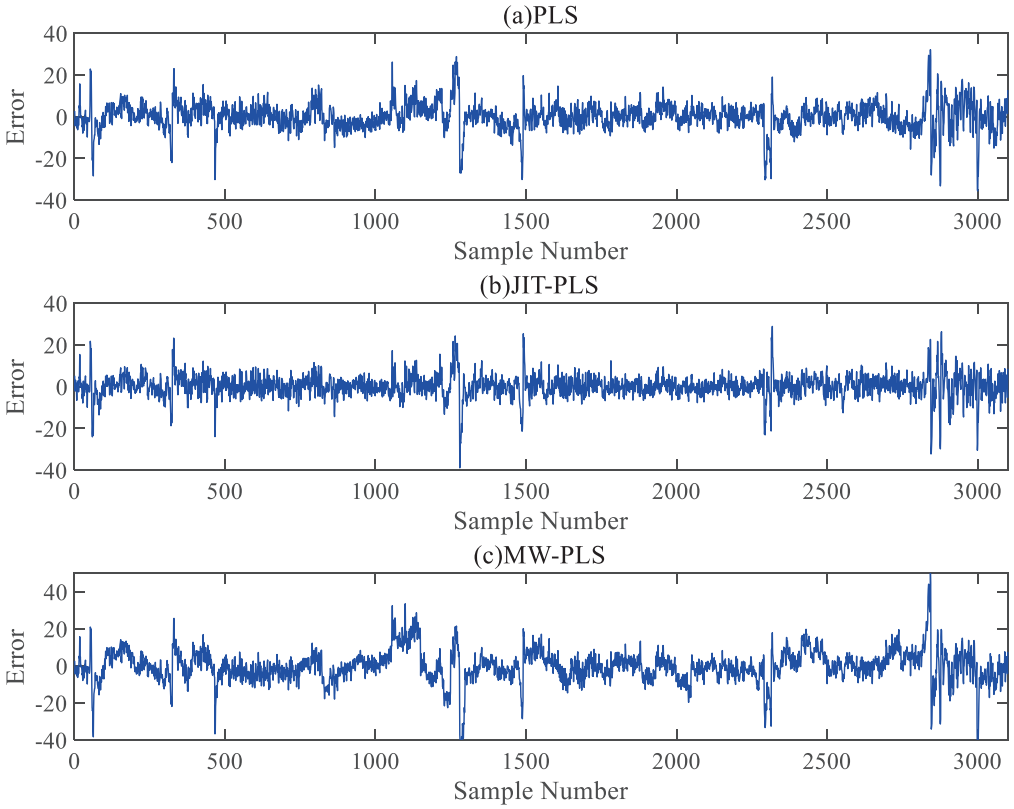


Figure 9. Error curves of the PLS algorithm.

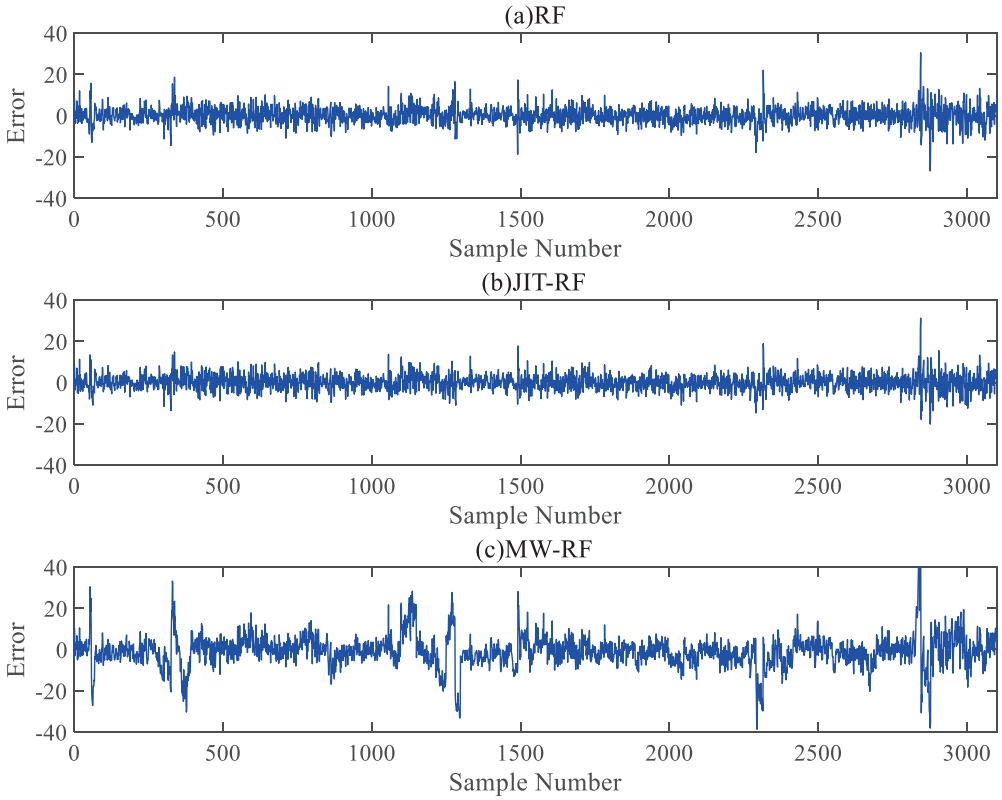


Figure 10. Error curves of the RF algorithm.

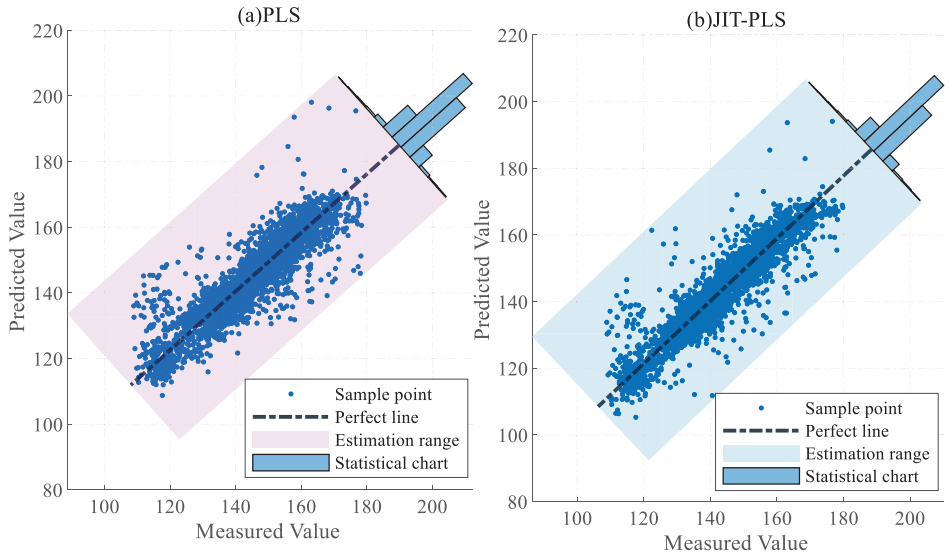


Figure 11. Predictive scatter plots for PLS and JIT-PLS models.

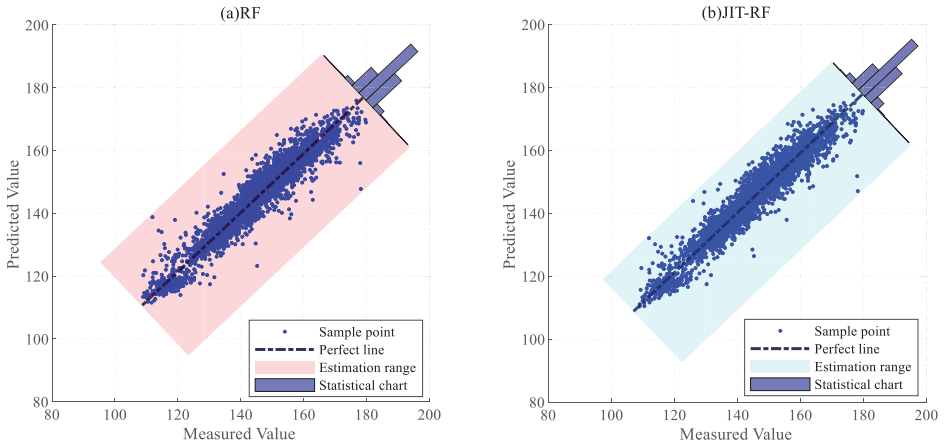


Figure 12. Predictive scatter plots for RF and JIT-RF models.

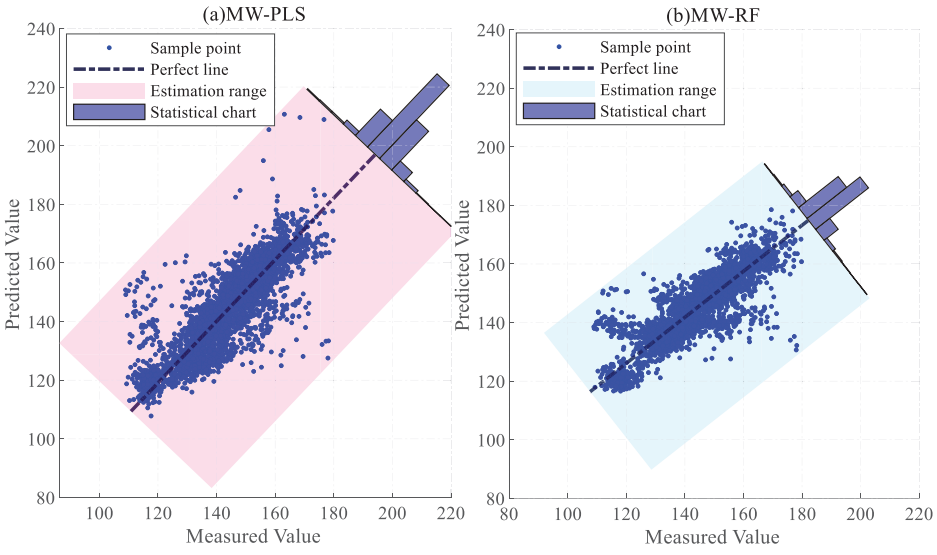


Figure 13. Predictive scatter plots for MW-PLS and MW-RF models.

The above analysis indicates that the proposed method outperforms traditional approaches and is particularly suited for the online prediction of the concentration of NOx emissions.

6. Conclusions

In this work, an improved JITL-based prediction method is proposed to predict the concentration of NOx emissions in a coal-fired power plant. A supervised similarity distance measurement method is defined, and local training samples can be effectively selected. To establish a robust model, a variable selection method is also proposed that enables the robust selection of important variables. Several comparative experiments on a real-world industrial dataset are presented. The experiments show that the proposed method has good accuracy. Therefore, it is suitable for the long-term prediction of NOx emissions. With the development of industrial information technology, the acquisition of multisource process data has become increasingly convenient. Based on multisource data, such as audio and images, building a large prediction model will be beneficial to improving the accuracy and real-time performance of NOx emission prediction, which will be our future research focus.

Author Contributions: K.H.: conceptualization, methodology, and writing—original draft preparation; H.D.: software, data curation, visualization, and investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under grant numbers 62273214 and 62233012, and in part by the Research Fund for the Taishan Scholar Project of Shandong Province of China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data generated in this study are presented in this article. For any clarifications, please contact the corresponding author.

Acknowledgments: The authors would like to thank the editors and anonymous referees for their invaluable comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, K.; Su, H. Exploring the current situation of air pollution emissions and flue gas desulfurization and denitration technologies in thermal power plants. *China Plant Eng.* **2022**, *1*, 211–212.
2. Wang, B. Analysis of integrated air pollution control technology in thermal power industry. *Energy Conserv. Environ. Prot.* **2021**, *10*, 64–66.
3. Cao, W. Soft Sensor and Its Application of Flue Gas NOx Concentration in Thermal Power Plant Based on Improved Autoencoder. Master's Thesis, Zhejiang University, Zhejiang, China, 2022.
4. Zhang, L.; Lin, D.; Wang, Y.; Ji, G.; Ma, S.; Cao, X.; Liu, Z.; Ma, R.; Wang, B. Review of applications of machine learning in nitrogen oxides reduction in thermal power plants. *Therm. Power Gener.* **2022**, *52*, 7–17.
5. Cai, S. Challenges and prospects for the trends of power structure adjustment under the goal of carbon peak and neutrality. *S. Energy Constr.* **2021**, *8*, 8–17.
6. Qin, S. Analysis of Wind Power Grid Connection Technology and Power Quality Control. *Electronic Technology.* **2022**, *51*, 110–111.
7. Pang, S.; Sun, P. Application analysis of flue gas desulfurization and denitration technology in air pollution control of thermal power plant. *Leather Manuf. Environ. Technol.* **2022**, *3*, 82–84.
8. He, L.; Wei, H.; Cui, Y.; He, J. Research on prevention and control technology of boiler air pollution in thermal power plant. *Chem. Eng. Manag.* **2021**, *24*, 45–46.
9. Qian, H.; Zhang, C.; Chai, T. Research on outlet NOx concentration prediction of SCR denitration system based on random forest algorithm. *J. Eng. Therm. Energy Power* **2021**, *36*, 122–129+137.
10. Pan, Y. Intelligent Control of Power Plant SCR Flue Gas Denitration System Based on Mechanism Modeling. Ph.D. Thesis, North China Electric Power University, Beijing, China, 2019.
11. Li, K. Research on NOx Emission Prediction of Power Plant Boiler Based on Ensemble Learning Method. Ph.D. Thesis, Northeast Electric Power University, Jilin, China, 2021.

12. Tang, Z.; Zhu, D.; Li, Y. Data driven based dynamic correction prediction model for NOx emission of coal fired boiler. *Proc. CSEE* **2022**, *42*, 5182–5194.
13. Zhuo, J.; Jiao, W.; Song, G.; Xiong, S.; Yao, Q.; Pan, T. A review on nitrogen oxides prediction model in combustion optimization of boilers. *J. Combust. Sci. Technol.* **2016**, *22*, 531–540.
14. Li, Z.; Lee, Y.; Chen, J.; Qian, Y. Developing variable moving window PLS models: Using case of NOx emission prediction of coal-fired power plants. *Fuel* **2021**, *296*, 120441. [CrossRef]
15. Wang, F.; Ma, S.; Wang, H.; Li, Y.; Zhang, J. Prediction of NOx emission for coal-fired boilers based on deep belief network. *Control. Eng. Pract.* **2018**, *80*, 26–35. [CrossRef]
16. Tuttle, J.; Vesel, R.; Alagarsamy, S.; Blackburn, L.; Powell, K. Sustainable NOx emission reduction at a coal-fired power station through the use of online neural network modeling and particle swarm optimization. *Control. Eng. Pract.* **2019**, *93*, 104167. [CrossRef]
17. Yang, G.; Wang, Y.; Li, X. Prediction of the NOx emissions from thermal power plant using long-short term memory neural network. *Energy* **2020**, *192*, 116597. [CrossRef]
18. Yuan, Z.; Meng, L.; Gu, X.; Bai, Y.; Cui, H.; Jiang, C. Prediction of NOx emissions for coal-fired power plants with stacked-generalization ensemble method. *Fuel* **2021**, *289*, 119748. [CrossRef]
19. Li, N.; Lv, Y.; Hu, Y. Prediction of NOx emissions from a coal-fired boiler based on convolutional neural networks with a channel attention mechanism. *Energies* **2022**, *16*, 76. [CrossRef]
20. Korpela, T.; Kumpulainen, P.; Majanne, Y.; Häyrinen, A.; Lautala, P. Indirect NOx emission monitoring in natural gas fired boilers. *Control. Eng. Pract.* **2017**, *65*, 11–25. [CrossRef]
21. Xie, P.; Gao, M.; Zhang, H.; Niu, Y.; Wang, X. Dynamic modeling for NOx emission sequence prediction of SCR system outlet based on sequence to sequence long short-term memory network. *Energy* **2020**, *190*, 116482. [CrossRef]
22. Xing, H.; Guo, J.; Zhang, Y.; Liu, S.; Liu, B.; Chang, Z. NOx emission prediction based on variable selection and XGBoost combined model. *Autom. Instrum.* **2021**, *7*, 21–25.
23. Tang, Z.; Wang, S.; Chai, X.; Chai, S.; Ouyang, T.; Li, Y. Auto-encoder-extreme learning machine model for boiler NOx emission concentration prediction. *Energy* **2022**, *256*, 124552. [CrossRef]
24. Wang, Z.; Peng, X.; Cao, S.; Zhou, H.; Fan, S.; Li, K.; Huang, W. NOx emission prediction using a lightweight convolutional neural network for cleaner production in a down-fired boiler. *J. Clean. Prod.* **2023**, *389*, 136060. [CrossRef]
25. Zhang, J.; Gu, C.; Li, R.; Jin, J. Design of Coupling Prediction Model for Boiler Energy Efficiency and NOx Emission. *Energy Conserv. Technol.* **2020**, *38*, 407–411+481.
26. Yang, T.; Ma, K.; Lv, Y.; Bai, Y. Real-time dynamic prediction model of NOx emission of coal-fired boilers under variable load conditions. *Fuel* **2020**, *274*, 117811. [CrossRef]
27. Lv, Y.; Liu, J.; Yang, T.; Zeng, D. A novel least squares support vector machine ensemble model for NOx emission prediction of a coal-fired boiler. *Energy* **2013**, *55*, 319–329. [CrossRef]
28. Yu, Y.; Han, Z.; Xu, C. NOx concentration prediction based on deep convolution neural network and support vector machine. *Proc. CSEE* **2022**, *42*, 238–248.
29. Wu, S.; Ma, Y. Prediction of NOx emission from power plant boiler based on mixed deep learning network. *China Meas. Test* **2022**, *48*, 166–174.
30. Wen, X.; Li, K.; Wang, J. NOx emission predicting for coal-fired boilers based on ensemble learning methods and optimized base learners. *Energy* **2023**, *264*, 126171. [CrossRef]
31. Liu, H.; Wang, Y.; Li, X.; Yang, G. Prediction of NOx emissions of coal-fired power plants based on mutual information-graph convolutional neural network. *Proc. CSEE* **2021**, *42*, 1052–1060.
32. Huang, Z. Prediction of NOx emissions from power plants based on EEMD and convolutional neural network. *J. Eng. Therm. Energy Power* **2022**, *37*, 96–103.
33. Ma, L.; Zhang, L.; Gu, Y.; Wu, W.; Tang, Y. Prediction of NOx concentration at the entrance of denitration system based on stacking generalization ensemble method. *Electr. Power Technol. Environ. Prot.* **2022**, *38*, 517–524.
34. Wang, H.; Zhang, G.; Huang, Y.; Zhang, Y. NOx emission prediction of boiler based on stacking ensemble learning. *Ind. Control. Comput.* **2021**, *34*, 92–93+96.
35. Li, Y.; Huang, W.; Xi, J. NOx Emission Forecasting based on Stacking Ensemble Model. *J. Eng. Therm. Energy Power* **2021**, *36*, 73–81.
36. Wang, W.; Fan, H.; Liang, C.; Zhao, Z.; Shao, Y.; Tan, C.; Zheng, C. Predictive modeling of NOx outlet of hedged boiler based on random forest. *Therm. Power Gener.* **2022**, *51*, 91–104.
37. Wang, Y. Research on Nox Emission Prediction and Control of Coal-Fired Boiler. Ph.D. Dissertation, North China Electric Power University, Beijing, China, 2021.
38. Lv, Y.; Yang, T.; Liu, J. An adaptive least squares support vector machine model with a novel update for NOx emission prediction. *Chemom. Intell. Lab. Syst.* **2015**, *145*, 103–113. [CrossRef]
39. Napier, L.; Aldrich, C. An IsaMill™ soft sensor based on random forests and principal component analysis. *IFAC-PapersOnLine* **2017**, *50*, 1175–1180. [CrossRef]
40. Lv, Y.; Lv, X.; Fang, F.; Yang, T.; Romero, C. Adaptive selective catalytic reduction model development using typical operating data in coal-fired power plants. *Energy* **2020**, *192*, 116589. [CrossRef]

41. Yang, X.; Zhou, C. Research progress on the application of just-in-time learning in process industry. *Comput. Appl. Chem.* **2018**, *35*, 746–758.
42. Joshi, T.; Goyal, V.; Kodamana, H. A novel dynamic just-in-time learning framework for modeling of batch processes. *Ind. Eng. Chem. Res.* **2020**, *59*, 19334–19344. [CrossRef]
43. Ma, X.; Wang, T.; Zhou, H. Fuzzy multiple kernel support vector mechine based on weighted mahalanobis distance. *Comput. Sci.* **2022**, *49* (Suppl. S2), 302–306.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Multi-Scale Feature and Multi-Channel Selection toward Parkinson's Disease Diagnosis with EEG

Haoyu Wu [1], Jun Qi [1,*], Erick Purwanto [1], Xiaohui Zhu [1], Po Yang [2] and Jianjun Chen [1,*]

[1] Department of Computing, Xi'an Jiaotong-Liverpool Univeristy, Suzhou 215000, China; haoyu.wu18@student.xjtlu.edu.cn (H.W.); erick.purwanto@xjtlu.edu.cn (E.P.); xiaohui.zhu@xjtlu.edu.cn (X.Z.)
[2] Department of Computer Science, The University of Sheffield, Sheffield S10 2TN, UK; po.yang@sheffield.ac.uk
* Correspondence: jun.qi@xjtlu.edu.cn (J.Q.); jianjun.chen@xjtlu.edu.cn (J.C.)

Abstract: Objective: Motivated by Health Care 4.0, this study aims to reducing the dimensionality of traditional EEG features based on manual extracted features, including statistical features in the time and frequency domains. Methods: A total of 22 multi-scale features were extracted from the UNM and Iowa datasets using a 4th order Butterworth filter and wavelet packet transform. Based on single-channel validation, 29 channels with the highest R2 scores were selected from a pool of 59 common channels. The proposed channel selection scheme was validated on the UNM dataset and tested on the Iowa dataset to compare its generalizability against models trained without channel selection. Results: The experimental results demonstrate that the proposed model achieves an optimal classification accuracy of 100%. Additionally, the generalization capability of the channel selection method is validated through out-of-sample testing based on the Iowa dataset Conclusions: Using single-channel validation, we proposed a channel selection scheme based on traditional statistical features, resulting in a selection of 29 channels. This scheme significantly reduced the dimensionality of EEG feature vectors related to Parkinson's disease by 50%. Remarkably, this approach demonstrated considerable classification performance on both the UNM and Iowa datasets. For the closed-eye state, the highest classification accuracy achieved was 100%, while for the open-eye state, the highest accuracy reached 93.75%.

Keywords: machine learning; Parkinson's disease; EEG; channel selection; wavelet packet transform

1. Introduction

Parkinson's disease (PD) is a chronic and progressive neurodegenerative disorder that affects millions of people worldwide [1]. It is characterized by various motor and non-motor symptoms, thus making it a complex condition to diagnose and manage. The cardinal motor symptoms of Parkinson's disease include bradykinesia (slowness of movement), rigidity (stiffness of muscles), tremors, and postural instability. However, PD is not solely a motor disorder. Patients may also experience a wide range of non-motor symptoms, such as depression, anxiety, sleep disturbances, cognitive impairment, and autonomic dysfunction, as well as some non-motor symptoms, such as sleep disturbances, may appear in the early stage of PD [1,2]. To date, the diagnosis of PD is mainly based on clinical assessment and the presence of specific motor symptoms. There is no definitive diagnostic test for PD, so healthcare professionals must carefully evaluate the patient's history, perform a thorough physical examination, and rule out other conditions that may mimic PD symptoms. Motivated by Health Care 4.0, which is a new era of health care propelled by the advent of Industry 4.0 [3], numerous researchers have explored the potential of utilizing wearable sensors signal [4] for the computer-based diagnosis of Parkinson's disease, such as using an accelerometer [5,6] and EEG. EEG testing, a non-invasive and cost-effective technique widely available in medical centers that captures the electrical and magnetic field signals generated by neuronal activity on the scalp at specific

frequencies [7], has demonstrated effectiveness in diagnosing and predicting various neurological disorders, including epileptic seizures, the development of biomarkers for Alzheimer's disease, and the detection of abnormalities in schizophrenia [8–10].

Nevertheless, EEG signals present two main challenges for analysis: their stochastic nature and low signal-to-noise ratio (SNR) [11]. The low SNR indicates a high noise level within EEG signals, thus complicating pre-processing efforts, as current algorithms partially filter out the signal while attempting to remove noise. This adversely impacts classification model performance. Moreover, the inherent randomness of EEG signals demands the use of sophisticated nonlinear and dynamic models for effective feature extraction and classification, leading to considerable computational time and resource requirements. To address the low SNR issue, several advanced pre-processing algorithms have been developed. For example, the research by [12] provided evidence of the effectiveness of their Artifact Subspace Reconstruction (ASR) as an automate method for removing artifacts. This study suggested that a cutoff parameter between 20 and 30 is preferable, as opposed to the previously suggested and default values of 5 to 7, which resulted in excessive removal of brain activities. And it also revealed that ASR enhances the quality of ICA decomposition, as indicated by an increased number of dipolar independent components. Gu et al. [13] have also proposed an automatic ocular artifact removal (AOAR) method for EEG signals, which outperformed the other methods in terms of the root mean square error, SNR, and correlation coefficient, particularly in cases with lower SNR levels. Islam et al. [14] made a good review of those studies in this field.

Considering that the classification models remain complex due to the high dimensionality of the feature vector, three approaches are commonly employed to reduce dimensionality: feature selection, frequency band analysis, and channel selection. Feature selection primarily relies on methods such as single-factor analysis of variance or chi-square analysis, thereby often employing *p*-value calculations [15,16]. Alternatively, deep learning techniques can be employed to fuse the feature matrix [17]. Frequency band analysis focuses on examining different frequency rhythms to identify more informative patterns. Similar to the aforementioned methods, channel selection aims to identify the most informative EEG channels to reduce the dimensionality of the feature vector at the channel level.

To reduce the dimensionality of traditional EEG features based on manual extraction (such as statistical features in the time and frequency domains), we propose a channel selection approach using single-channel classification, thus building upon previous research [18]. We selected a common set of 29 channels for our analysis based on the montage utilized in two publicly available datasets used in the referenced study. The specific contributions are outlined as follows:

- A Parkinson's disease EEG classification model depending on the public UNM and Iowa datasets has been proposed, which uses multi-scale features: a time domain, frequency domain features extracted by band-pass filter and wavelet packet transform, and an entropy feature (Section 3);
- We introduced a channel selection method consisting of 29 channels, which achieved a remarkable recognition accuracy of 100% on the UNM training dataset. Furthermore, the model successfully passed the out-of-sample testing on the Iowa dataset, thus demonstrating its generalizability. (Sections 4 and 5);

This paper follows the below structure: In Section 2, several state-of-the-art studies relevant to the domain of this paper will be discussed. In Section 3, we will discuss the detailed implementation processes and methods for the dataset, data pre-processing, and feature engineering. In Section 4, we will delve into the channel selection based on single-channel classification. In Sections 5 and 6, we will present the classification results, including the validation of the proposed channel selection method on the training dataset and the out-of-sample testing, as well as a comparison of the classification performance before and after channel selection.

2. Related Works

Gulay et al. [19] introduced a hybrid feature extraction method called EEMD VAR for EEG signals, thus integrating the EEMD and VAR techniques. Empirical comparisons were made between the EEMD VAR method, Hjorth parameters, and the AR model. The proposed algorithm achieved a maximum classification accuracy of 100%, thereby significantly outperforming the maximum accuracy of 72% attained by the AR method and Hjorth parameters. This highlights the potential of the EEMD VAR method to provide robust classification performance. Furthermore, the results show superior overall diagnostic accuracy, sensitivity, specificity, and AUC values for subjects with Parkinson's disease (PD). Among the classification algorithms evaluated, the ANN stands out as the most prominent based on all performance evaluation metrics considered in this study. One limitation of this study lies in the lack of discussion or exploration of channel selection for EEG signals. The researchers did not explain the reason for the channel selection and running of the out-of-sampling test, which raises concerns about the reliability of their channel selection approach. The use of the ANN significantly increases the computational costs associated with feature extraction and model training as well.

The study of Coelho et al. [20] examined the effectiveness of Hjorth features extracted from electroencephalographic (EEG) signals as potential biomarkers for Parkinson's disease (PD). Biomarkers are measurable indicators of some biological state or condition. For Parkinson's disease, typical examples include α-synuclein, uric acid levels, mutations in relevant genes such as LRRK2 or GBA, and brain MRI, which have been widely applied in Parkinson's disease risk assessment and symptom diagnosis [21]. Currently, the research on AI-based Parkinson's disease diagnosis using EEG is still exploring the use of machine learning features as potential biomarkers. The focus has primarily narrowed to EEG frequency bands associated with Parkinson's disease symptoms, such as alpha, theta, and beta bands [22–24]. Notably, the prominent role of the beta band in diagnosing resting tremors suggests that features related to the beta band could serve as new PD biomarkers. Their analysis utilized EEG data from PD patients exposed to auditory stimuli, which were obtained from the publicly available database known as The Patient Repository for EEG Data + Computational Tools (PRED + CT). The investigation revealed notable differences in the proposed biomarkers across various brain lobes, including parietal, frontal, central, and occipital regions, as well as between healthy individuals and those with PD. Support Vector Machine (SVM), Random Forest, and K-Nearest Neighbors (KNN) algorithms were employed for the classification task, along with a five-fold cross-validation process. The proposed method achieved a high accuracy of 89.56% in distinguishing PD patients from healthy individuals using an SVM classifier. However, the study lacks research on channel selection, because although it identified some channels that exhibited outstanding performance on different features, it did not further conduct training and validation for channel selection.

Smrdel [25] employed both common spatial patterns (CSPs) and a Laplacian mask to facilitate robust selection and feature extraction in their study. They utilized the CSP whitening matrix to identify the channels that exhibited the greatest potential for distinguishing between EEG signals from healthy individuals and those with PD. By leveraging the selected features obtained through CSPs, they achieved a classification accuracy of 85% when categorizing EEG records into groups of healthy controls and PD patients. Furthermore, using features extracted via the Laplacian operator, they achieved a classification accuracy of 90%; however, their research lacks validation of the channel selection approach. Although they identified some channels with the highest information content, they failed to demonstrate whether this channel selection approach can prevent overfitting when applied to datasets beyond their own.

Dar et al. [26] introduced a novel architecture, 1D-CRNN-ELM, which merges an Extreme Learning Machine (ELM) with a one-dimensional Convolutional Recurrent Neural Network (1D-CRNN) for emotion detection in patients with Parkinson's disease (PD). This hybrid approach leverages the strengths of both neural networks to enhance the feature

extraction and classification of EEG data. This framework is designed to handle various emotions and experimental conditions, thus enabling cross-dataset learning. After pre-processing the EEG data, the CRNN serves as a feature extractor, while the ELM functions as the classifier. The CRNN can be fine-tuned with additional datasets to learn new emotion sets. The study utilizes cross-dataset learning by training on PD patient data and fine-tuning with publicly available datasets and vice versa. Using an 80:20 train–test data split, the model achieved high accuracy values. Using the SEED-IV dataset to fine-tune the architecture resulted in an accuracy of 92.5%. Leave-one-out cross-validation yielded high mean accuracy values for all datasets in the experiment. The study also demonstrated that high-performance emotion detection is feasible using only 1-second EEG segments from 14 channels. However, the CRNN's computational costs remain high, and despite extensive validation within their dataset, cross-dataset validation is still lacking. Additionally, the reliability of the ELM is questionable.

Numerous studies have been conducted on the publicly available New Mexico and Iowa datasets, some of which serve as the foundation for this research. A new LEAPD index was created by Anjum et al. [27], which effectively and quickly distinguishes individuals with Parkinson's disease (PD) from control subjects. The LEAPD method encodes the power spectral density (PSD) using a limited number of parameters and holds promise for generating PD diagnostics or enhancing control algorithms for real-time applications. Moreover, it has the potential to improve the accuracy of predicting motor progression and refining classifications of PD-related sub-types. The utilization of the LEAPD index can contribute to the development of cost-effective diagnostic tools and real-time control signals for Parkinson's disease and other neurodegenerative conditions. Levodopa is a medication commonly used for the treatment of Parkinson's disease [4,28]. However, it remains unclear whether the channel selection method based on the LEAPD feature would be effective when applied to traditional multi-scale features. Furthermore, the study did not explore the channel selection scheme based on these traditional features within the dataset. In our study, we employed single-channel evaluation for channel selection. Another study focusing on the detection of abnormal EEG signals has provided insights into the extraction of sub-band wavelet coefficients and the calculation of statistical characteristics using the Discrete Wavelet Transform (DWT) [29]. This study extracted six statistical features from the wavelet coefficients of each sub-band. Additionally, this study has addressed the issue of feature redundancy in multi-scale feature extraction through the utilization of ensemble learning techniques.

3. Machine Learning-Based Classification Model for Parkinson's Disease Patients' EEG Signal

The classification model was developed based on the UNM dataset and tested on the UNM dataset and Iowa dataset. The implementation structure of this model is illustrated in Figure 1. The UNM dataset was divided into a training set and a validation set, while the Iowa dataset was utilized as an external validation data source. After undergoing pre-processing, all data were subjected to two feature extraction approaches, thus resulting in the generation of 13 and 9 distinct features, respectively. These 22 features constituted the feature vector used to train six commonly used machine learning models. The model's classification performance and the reliability of the channel selection approach were evaluated based on the obtained results.

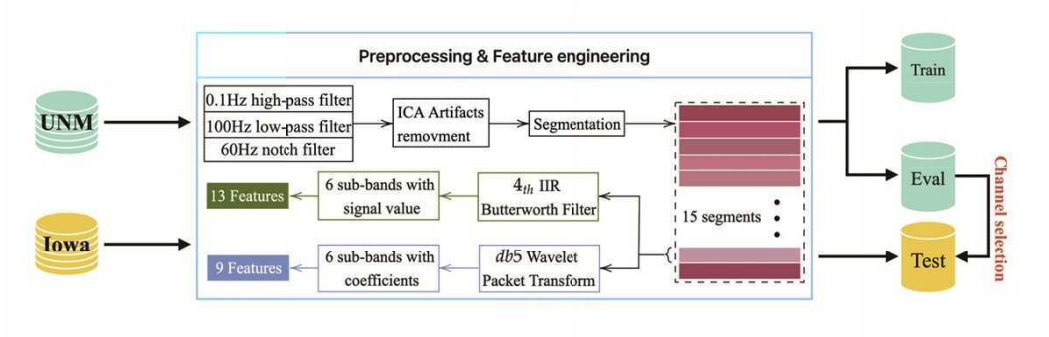


Figure 1. The implementation structure of the proposed EEG classification model.

3.1. Datasets

We utilized electroencephalogram (EEG) recordings obtained from two separate studies conducted at the University of New Mexico (UNM) in Albuquerque, New Mexico, and the University of Iowa (UI) in Iowa City, Iowa [22,27,30]. The UNM dataset consisted of EEG recordings from 27 patients diagnosed with Parkinson's disease (PD) and 27 control participants (a control record was excluded from the current study due to its insufficient duration), while the UI dataset included EEG recordings from 14 PD patients and 14 control participants. For the PD patients from the UNM dataset, EEG recordings were obtained during the OFF medication sessions, specifically in the defined OFF levodopa period, which occurred 12 h after the last dose of dopaminergic medication.

As discussed by Anjum et al. [27], the control participants were carefully matched with the PD patients in terms of age and sex, and there were no significant differences in education or pre-morbid intelligence measurements between the two groups. Parkinson's disease patients underwent neuropsychological and questionnaire assessments in their ON state, with a neurologist administering the United Parkinson's Disease Rating Scale (UPDRS) motor scores. Resting state EEG recordings were acquired from UNM subjects in both eyes-open and eyes-closed conditions, while EEG data from the 28 Iowa subjects were recorded solely in the eyes-open condition. The details are shown in Table 1.

Table 1. PD and Control Participant Demographics.

Dataset	**UNM**		**Iowa**	
Condition	**PD**	**Control**	**PD**	**Control**
Sex	17M 10F	17M 10F	6M 8F	6M 8F
Age	69.5 ± 8.7	69.5 ± 9.3	70.5 ± 8.7	70.5 ± 8.7
MMSE	28.7 ± 1	28.8 ± 1	-	-
MOCA	-	-	25.9 ± 2.7	27.2 ± 1.7
UPDRS	22.2 ± 10.3	-	13.4 ± 6.6	-
Year since Dx	5.7 ± 4.2	-	5.6 ± 3.2	-
EEG recording (min)	3.59 ± 1	3.63 ± 1.8	3.11 ± 1.2	3.17 ± 0.9
BDI	7.6 ± 5.3	4.8 ± 4.8	-	-
Year of Ed	17.3 ± 3.3	16.6 ± 3.1	16.6 ± 3.7	16.6 ± 2.8
Year of Ed (Parents)	12.5 ± 3.8	12.5 ± 3.1	-	-
LED (mg)	707.4 ± 448.6	-	796 ± 409	-
NAART	45.2 ± 10.3	47.1 ± 7.5	-	-

3.2. Pre-Processing

The EEG signals underwent a filtering process ranging from 0.1 Hz to 100 Hz, with a sampling rate of 500 Hz, using two different 64-channel montages. In the UNM dataset, the online reference was set to channel CPz, while in the Iowa dataset, it was set to channel Pz. Consequently, after removing channels that exhibited differences across those two montages, 59 channels were utilized for subsequent feature extraction and classification. To address artifacts caused by eye blinks, independent component analysis (ICA) was

employed, thus resulting in the extraction of 56 independent components [31,32]. Additionally, a 60 Hz notch filter was applied to eliminate interference from the power line.

For the UNM dataset, each EEG recording comprised two distinct sessions: one with eyes closed and another with eyes open. Following the feature extraction process, the features corresponding to these two events were identified and segregated into two separate datasets.

The two events, namely eyes closed and eyes open, had a duration of 60 s each, thus resulting in 30,000 samples for each event within a single recording. To facilitate further analysis, both events were divided into 15 segments, with each segment containing 2000 samples. Within each segment, we extracted six sub-bands to enable additional feature engineering. These sub-bands included delta (1–4 Hz), theta (4–8 Hz), alpha1 (8–10 Hz), alpha2 (10–13 Hz), alpha (8–13 Hz), and beta (13–30 Hz). The boundary frequencies were treated as separate sub-bands. To separate these sub-bands, we employed two methods: a 4th order Butterworth IIR filter and Wavelet Package Transform. From each sub-band, we extracted multi-scale features, thus capturing diverse aspects of the EEG signal.

3.3. Feature Extraction

In order to extract multi-scale features from the aforementioned six sub-bands, both IIR filtering and Wavelet Packet Transform were applied to extract the corresponding frequency ranges from the original signal, as shown in Figure 2. We employed a 4th Butterworth IIR filter as a band-pass filter and a notch filter to generate sub-bands and extract 13 features. The purpose of the notch filter was to eliminate noise from the power frequency signals at 60 Hz and 180 Hz. Next, we extracted the six sub-bands mentioned earlier from each pre-processed segment by utilizing the IIR filter as six corresponding bandpass filters. From each sub-band, we extracted a total of 13 features, thus consisting of 9 time domain features, three frequency-domain features, and one entropy feature, as shown in Table 2, where MAD represents for the mean value of the signal's absolute deviation, which is expressed as below:

$$MAD = \frac{1}{m}\sum_{i}^{m}|x_i - u| \qquad i = 1, 2, \cdots, m \tag{1}$$

where x_i represents the ith data point in the sequence, while u represents the arithmetic mean value of the sequence. RMAV represents for the ratio of the absolute mean value between two sub-bands. To derive the remaining nine features, we employed Wavelet Packet Transform using the $db5$ wavelet. For each segment, we utilized the pre-processed signal without any resampling. In order to generate the 6 sub-bands, we set the decomposition level to six, thus taking into consideration the Sampling Theorem. Specifically, we selected six coefficients: AAAAAA6, AAAAAD6, AAAADA6, AAAADD6, AAAAD5, and AAAD4, as illustrated in Figure. From these coefficients, we calculated nine features, as outlined in Table 3.

As was mentioned before, there are two kinds of eye state in UNM dataset: eyes open and eyes closed. Considering the significant differences in data collected based on these two eye conditions and the non-parallel nature of the data collection process, we applied our feature engineering separately to these two types of data, thus resulting in corresponding feature sets, as shown in Table 4. For each participant with one eye state and one feature extraction method, the dimensionality of the feature vector came out to $59 \times 15 \times 6 \times 9$ or 13. Here, 59 represents the common number of channels, 15 denotes the number of segments, 6 indicates the number of sub-bands extracted, and 9 or 13 signifies the number of extracted features depending on different feature extraction methods.

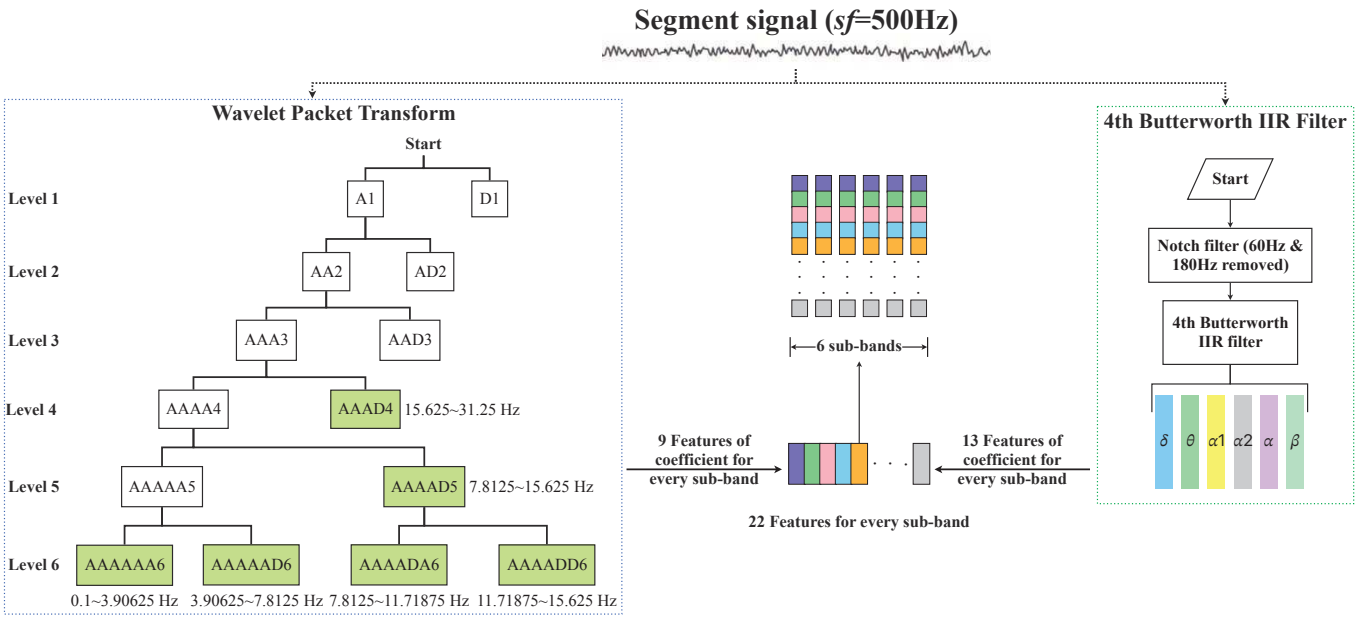


Figure 2. Feature engineering structure for one segment signal, where features obtained from different methods are stored separately.

Table 2. Features extracted from sub-bands generated by band-pass filter.

Scales	Feature Types
Time domain	Mean, Absolute Mean, Power Mean, Standard Deviation, Kurtosis, Skewness, MAD, Interquartile Range for every Sub-Band and RMAV for Adjacent Sub-Bands.
Frequency domain	Freqeuncy Center, Mean Value of Frequency and Root Mean Square of the Frequency for Every Sub-Bands.
Entropy	Sample Entropy for every Sub-Band.

Table 3. Features extracted from sub-bands generated using Wavelet Packet Transform.

Scales	Feature Types
Coefficient	Mean, Absolute Mean, Power Mean, Standard Deviation, Kurtosis, Skewness, MAD, Interquartile Range of the Coefficients for every Sub-Band and RMAV for Adjacent Sub-Bands.

Table 4. Dimensionality of the feature vectors in different feature datasets for one participant.

Eye States	Feature Extraction Methods	Dimensionality
Eyes Closed	IIR	69,030
	Wavelet	47,790
Eyes Open	IIR	69,030
	Wavelet	47,790

4. Channel Selection

Channel selection is one of the main contributions of this study. Our channel selection was based on the UNM dataset, where the training set was used to train a simple SVM model, and the validation set was used to evaluate the classification performance of each channel. Figures 3 and 4 demonstrate the classification performance of each channel after feature extraction using Wavelet Packet Decomposition. It is worth noting that some channels shown in the images do not exist in the Iowa dataset and were ignored in

subsequent testing. The evaluation metric for assessing the classification performance of individual channels is the R2 score, and we set the threshold for channel selection at 0.7, thus meaning that channels with an R2 score below 0.7 were not accepted. Following this criterion, a total of 29 channels that met the requirement were selected based on the two feature extraction methods. For the feature set derived from the 4th Butterworth IIR filter, 11 channels were selected: 'Oz', 'F4', 'P8', 'CP2', 'Cz', 'Fp2', 'P2', 'FC5', 'T7', 'O1', and 'FC6'. For the feature set derived from Wavelet Packet Decomposition, 22 channels were selected: 'TP7', 'TP9', 'TP10', 'FC3', 'FC4', 'FC5', 'PO7', 'PO8', 'CP3', 'CP5', 'P1', 'P3', 'P4', 'P5', 'P6', 'P8', 'AF3', 'C4', 'F5', 'Oz', 'O1', and 'O2'. That is, 29 channels were selected in total. Figure 5 shows an example of the relationship between 20 selected channels' feature values and model outputs. In order to visualize this relationship into a 2D figure, we simply took the average value of all feature values for each channel, as the shown feature value. SHAP (SHapley Additive exPlanations), which is a method rooted in game theory that elucidates the results of machine learning models, was applied by using a SHAP Python tool to describe the impact on model output for every feature value in this figure [33]. Therefore, positive correlation in this diagram can be observed as high feature values accompanied by positive SHAP values, and all these 20 channels showed a positive correlation with the model output.

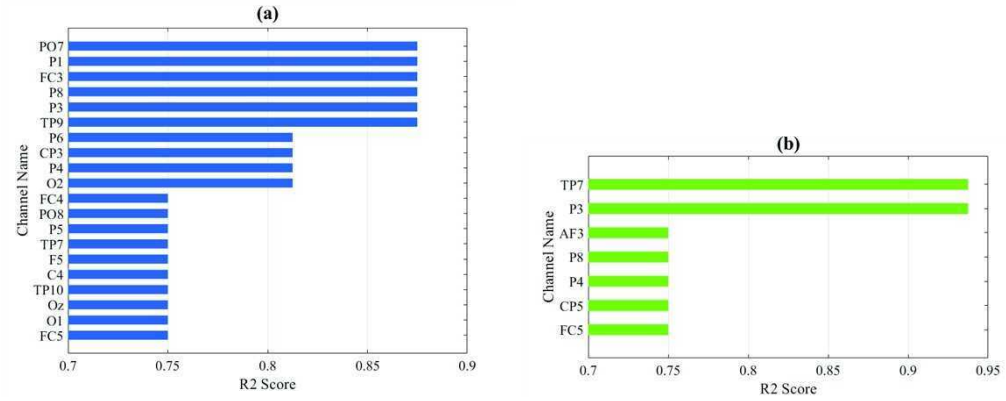


Figure 3. Single-channel evaluation R2 score result for eyes-closed (**a**) and eyes-open (**b**) feature dataset extracted by Wavelet Packet Transform.

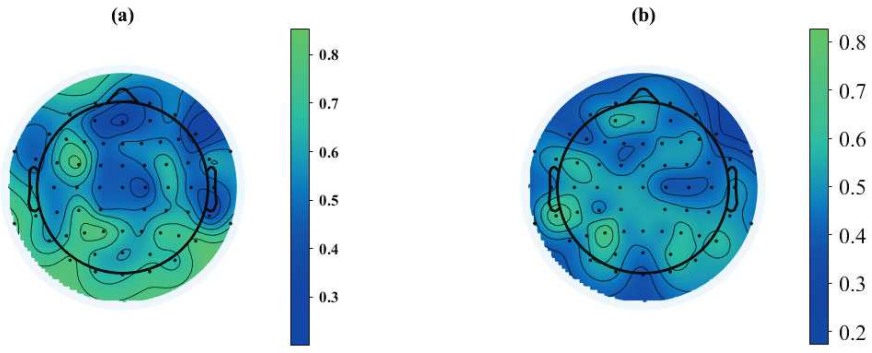


Figure 4. Single-channel evaluation brain maps for eyes-closed (**a**) and eyes-open (**b**) feature dataset extracted by Wavelet Packet Transform. The color bar refers to the R2 score, and high R2 score means high single-channel classification performance.

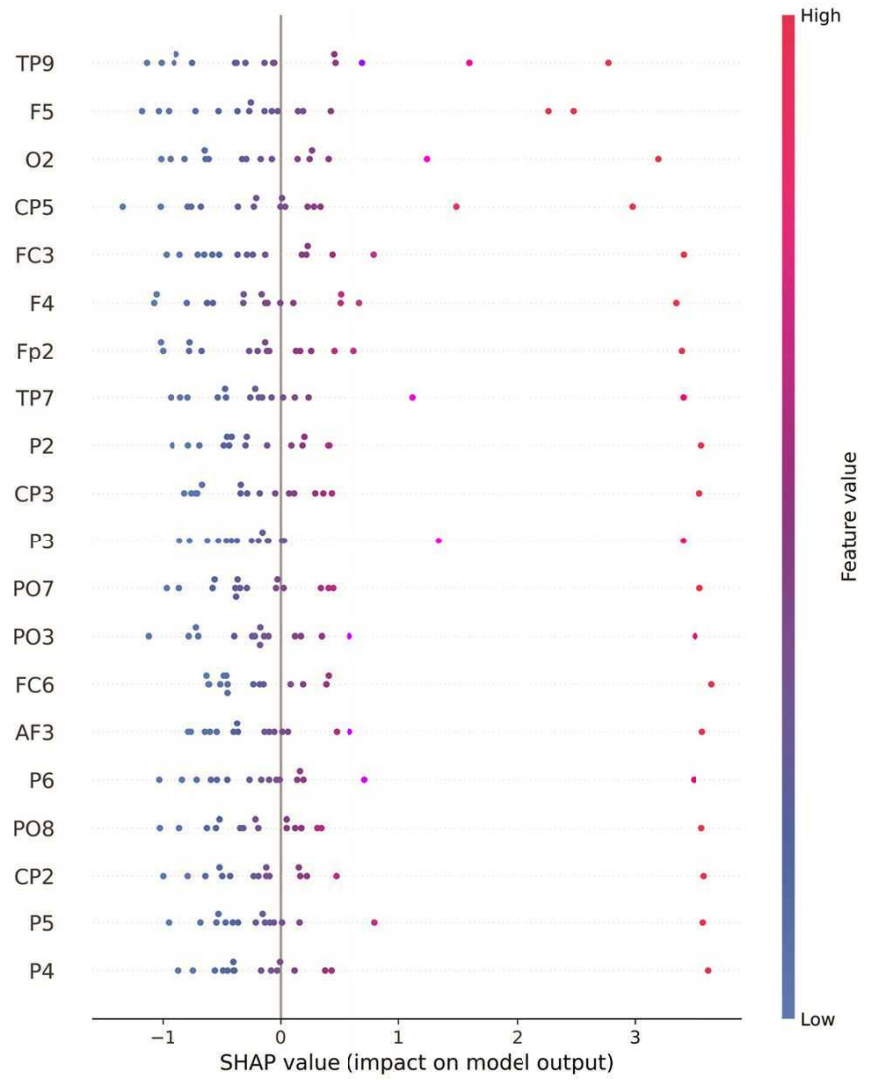


Figure 5. Relationship between channel feature value and the model output.

5. Results

In the multi-channel classification task, six machine learning models were employed: Logistic Regression (LR), KNN, SVM, AdaBoost, XGBoost, and Random Forest. Logistic Regression had a maximum iterate number of 5000, while KNN used seven neighbors. The linear kernel was applied to SVM, just like in the single-channel evaluation task. The learning rates for AdaBoost and XGBoost 1.5.1 were both set to 0.01. AdaBoost terminated boosting at a maximum of 50 estimators, while XGBoost had a gamma value of 0.05. The Random Forest model had a default setting of 100 trees. Considering the objectives of this study, as well as the complexity and limited interpretability inherent in deep learning, we have not devised a comprehensive deep learning EEG classification framework. Instead, a simple neural network with two fully connected (FC) layers with layer sizes of 512 and 1 was employed as a classifier. The dropout rate was set to 0.2 after the first FC layer, and we chose binary cross entropy as the loss function; the optimizer was Adam, as usual. All the parameters were selected by applying a grid search. For training and testing, data with features from all channels and the best channels were used separately. The results were presented and compared in our previous study [18]. In this session, we will briefly give the overall evaluation results on the UNM dataset, which are shown in Table 5 and mainly focus on the channel selection results and the out-of-sample testing.

Table 5. Overall accuracy results with all 22 features on two eyes states with different numbers of channels.

State		Models							
		LR	KNN	SVM	AdaBoost	XGBoost		RF	ANN
						Tree	Linear		
Eyes closed	29 best channels	93.75%	93.75%	**100%**	75%	87.5%	93.75%	87.5%	93.75%
	All 59 channels	93.75%	87.5%	93.75%	81.25%	87.5%	81.25%	87.5%	62.5%
Eyes open	29 best channels	87.5%	62.5%	81.25%	75%	75%	81.25%	81.25%	**93.75%**
	All 59 channels	81.25%	62.5%	81.25%	81.25%	75%	75%	75%	62.5%

We conducted testing on the evaluation set of the UNM dataset and calculated the R2 score, which is shown in Figure 6. The possibility of selecting fewer channels to significantly reduce the dimensionality of the features was also evaluated by comparing the classification performance of two feature extraction methods, along with their respective channel selection schemes and the intersection of these schemes. The experimental results, presented in Figure 7, provide insights into the assessment of selecting fewer channels to achieve a greater reduction in feature dimensionality.

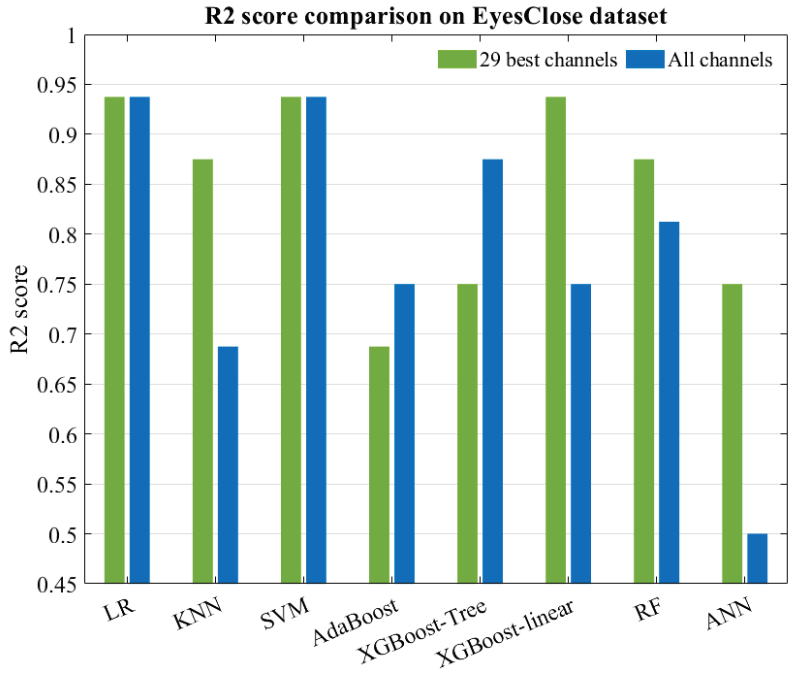


Figure 6. R2 score comparison on eyes-closed evaluation set.

Due to the availability of only open-eye state data in the Iowa dataset, we designed two training schemes for out-of-sample testing: one using only open-eye state data from the UNM dataset as the training set and the other using both eye states from the UNM dataset. Both schemes utilized the Iowa dataset as the test set, and the results are presented in Table 6. When using the UNM (open-eye) dataset as the training set and the Iowa (open-eye) dataset as the test set, the LR model performed the best, thus correctly classifying 24 out of 28 samples, followed by the SVM and other models, with 23 out of 28 samples correctly classified. All models showed better performance compared to the results obtained from testing within the UNM dataset, with an average increase in accuracy of approximately 0.03 and a negligible average decrease in the R2 score of less than 0.01.

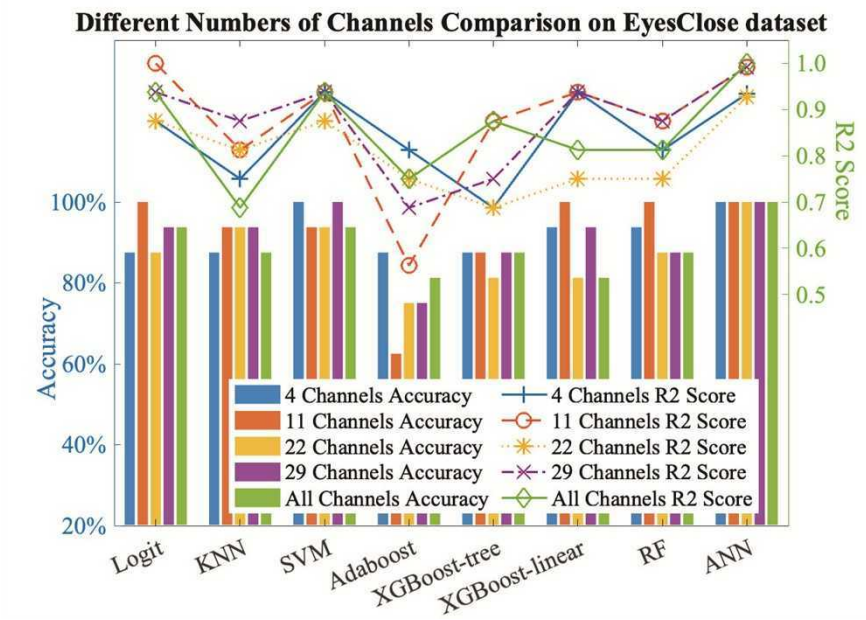


Figure 7. Comparison between different channel selection schemes on eyes-closed dataset.

Table 6. Out-of-sample Test Results.

Score Type	Eye State	LR	KNN	SVM	AdaBoost	XGBoost-Tree	XGBoost-Linear	RF	ANN	Mean
Accuracy	eyes open	0.857	0.714	0.821	0.821	0.821	0.821	0.786	0.821	0.808
	eyes open & closed	0.857	0.786	0.857	0.821	0.821	0.786	0.786	0.929	0.829
Sensitivity	eyes open	0.786	0.714	0.714	0.929	1.0	0.714	0.643	0.786	0.786
	eyes open & closed	0.714	0.623	0.714	0.929	0.929	0.642	0.571	0.929	0.754
Specificity	eyes open	0.929	0.714	0.929	0.714	0.643	0.929	0.929	0.857	0.830
	eyes open & closed	**1.0**	**0.929**	**1.0**	**0.714**	**0.714**	**0.929**	**1.0**	**0.929**	0.901
AUC_ROC	eyes open	0.841	0.783	0.832	0.796	0.827	0.821	0.849	0.821	0.821
	eyes open & closed	0.827	0.855	0.852	0.816	0.841	0.786	0.867	0.929	0.845
R2 Score	eyes open	0.714	0.678	0.786	0.714	0.75	0.75	0.679	0.286	0.676
	eyes open & closed	0.786	0.786	0.75	0.786	0.714	0.679	0.714	0.714	0.742

Furthermore, the results obtained by combining open-eye and closed-eye samples from the UNM dataset for training showed improvements compared to using only open-eye data. The average increase in accuracy was approximately 0.01, while the average decrease in sensitivity was around 0.05. On the other hand, the average increase in specificity was approximately 0.07, and the average increase in the area under the curve (AUC) was around 0.01. Additionally, the average increase in the R2 score was approximately 0.02.

6. Discussion

Our findings, as shown in Figure 6, indicate that the dataset with channel selection achieved a higher average R2 score compared to the dataset with all channels for the seven classifiers examined. Specifically, the KNN, XGBoost-linear, and Random Forest models showed improvements, while the AdaBoost and XGBoost-Tree models exhibited noticeable decreases in performance. However, when considering the overall picture, the classification models without the application of channel selection demonstrated a relatively poorer fit on the test set but showed better fit on the training set. This suggests that the utilization of all channels during training led to the occurrence of overfitting, thus resulting in inferior performance on the test set. These findings also demonstrate the successful implementation of our channel selection approach, which effectively balances the need for excellent classification performance (accuracy, sensitivity, precision, etc.) while mitigating the overfitting associated with involving all channels in the classification process.

Considering further selection of the chosen channels, we took the intersection (4 channels) and union (29 channels) of the channel selection results based on traditional multi-scale features (11 channels) and statistical features of the wavelet coefficients (22 channels). We then performed cross-validation on the training set to obtain the average R2 score and compared it with the R2 score on the test set. According to Figures 7 and 8, it was observed that using the 11 channels effectively reduced the R2 score on the training set while maintaining a better accuracy and R2 score on the test set, whereas the 22 channels did not exhibit the same behavior. Subsequently, testing the intersection (four channels) revealed a significant decrease in the R2 score on the training set, but the accuracy, sensitivity, and R2 score on the test set exhibited excellent performance, thus averaging above 0.9. This demonstrates that among the chosen 29 channels, the four channels 'Oz', 'P8', 'FC5', and 'O1' primarily contributed to the observed effects (although the 11-channel scheme exhibited better classification accuracy on the test set). The accuracy and R2 score reflect the classification performance and fitting degree on the respective test sets, and higher values indicate better model performance. According to Figure 8, it is evident that the 29-channel selection scheme remains the most reliable, with the majority of results concentrated above an R2 score of 0.7 and an accuracy of 0.8. Moreover, the R2 score exhibited minimal fluctuations in the high accuracy (0.9) region, thus consistently hovering around 0.8 and 0.9 and indicating good generalization ability of this scheme. Further testing can be conducted on out-of-sample data. In contrast, the four-channel selection scheme clearly exhibited severe overfitting, and therefore, this scheme was not considered for subsequent out-of-sample validation. It is worth noting that the 11-channel scheme showed similar performance, with an R2 score above 0.6 and an accuracy above 0.7. Although it is not as outstanding as the 29-channel scheme, it can be considered as an alternative depending on the specific application. However, it showed a larger fluctuation range in the high accuracy (0.9) region, thus indicating poorer generalization ability.

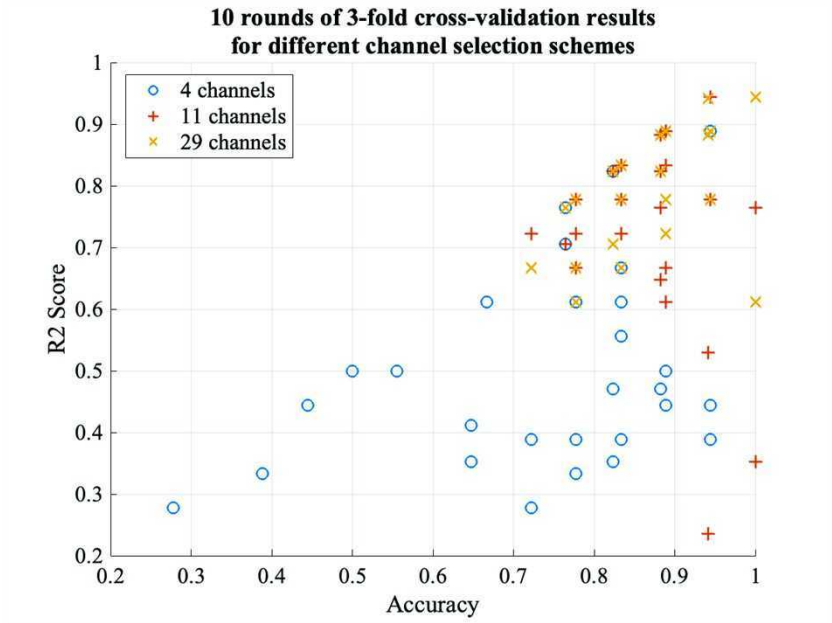


Figure 8. Comparison of 10 rounds of three-fold cross-validation results between different channel selection schemes on eyes-closed dataset.

From our out-of-sample testing results, it is evident that the proposed channel selection scheme has passed the external testing phase and exhibited comparable or even better performance than the original internal testing conducted on the dataset. Moreover, we observed that incorporating both open-eye and closed-eye samples in the training set was beneficial for classifying EEG signals under the open-eye state. We speculate that this improvement can be attributed to the fact that EEG signals recorded during the closed-eye

state are purer, with less noise compared to signals obtained during the open-eye state. Participants also experience less interference during the closed-eye state, thus making the signals closer to the theoretically pathological original signals and more valuable as references. In other words, training the machine learning model with a combination of closed-eye and open-eye datasets helps the model consider the features of both scenarios comprehensively and adjust the confidence intervals to approach the real range of Parkinson's disease onset. This, in turn, improves the classification of signals under the eyes-open condition.

7. Conclusions

EEG has proven to assist in the clinical diagnosis of Parkinson's disease. However, the computational costs associated with feature engineering and machine learning training on EEG signals are substantial, which hinders the monitoring and assisted diagnosis of Parkinson's disease in uncontrolled environments such as homes. Numerous studies have been conducted in this direction. In this paper, we conducted a study utilizing the publicly available UNM dataset and proposed a Parkinson's disease EEG classification model based on machine learning and traditional feature extraction techniques. We effectively reduced the feature dimensionality by employing a single-channel selection approach, thus theoretically reducing it by more than half. The proposed channel selection scheme not only achieved a maximum classification accuracy of 100% within the UNM dataset but also demonstrated its generalization capability through out-of-sample testing using the Iowa dataset. This verifies the effectiveness and generalizability of our channel selection method. In the future, we aim to develop an automated EEG channel selection model utilizing deep learning techniques based on our current methods. This model is designed to provide tailored channel selection schemes for individual Parkinson's patients. Additionally, we anticipate extending this approach to other EEG-related tasks, such as epilepsy diagnosis and emotion recognition, thereby enhancing its applicability and utility in various clinical and research settings. We believe that this channel selection approach can provide valuable insights for future research in related fields.

Author Contributions: Conceptualization, H.W.; Data curation, J.Q.; Funding acquisition, J.Q., E.P., X.Z., P.Y. and J.C.; Investigation, H.W.; Methodology, H.W.; Project administration, J.Q.; Resources, J.Q. and J.C.; Software, H.W.; Supervision, J.Q. and J.C.; Validation, H.W.; Visualization, H.W.; Writing—original draft, H.W.; Writing—review and editing, H.W., J.Q. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: The authors wish to acknowledge the support from the National Natural Science Foundation of China (No.62301452) and the Xi'an JiaoTong-Liverpool University Research Development Fund (RDF-21-01-080).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in Narayanan Lab, specifically in section 22: Linear predictive coding distinguishes spectral EEG features of Parkinson's disease, at https://narayanan.lab.uiowa.edu/article/datasets (accessed on 13 May 2024).

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Kouli, A.; Torsney, K.M.; Kuan, W.L. *Parkinson's Disease: Etiology, Neuropathology, and Pathogenesis*; Exon Publications: Brisbane, Australia, 2018; pp. 3–26. [CrossRef]
2. Kalia, L.V.; Lang, A.E. Parkinson's disease. *Lancet* **2015**, *386*, 896–912. [CrossRef] [PubMed]
3. Li, J.; Carayon, P. Health Care 4.0: A Vision for Smart and Connected Health Care. *IISE Trans. Healthc. Syst. Eng.* **2021**, *11*, 171–180. [CrossRef] [PubMed]
4. Qi, J.; Yang, P.; Newcombe, L.; Peng, X.; Yang, Y.; Zhao, Z. An overview of data fusion techniques for Internet of Things enabled physical activity recognition and measure. *Inf. Fusion* **2020**, *55*, 269–280. [CrossRef]

5. Yue, P.; Wang, X.; Yang, Y.; Qi, J.; Yang, P. Up-Sampling Active Learning: An Activity Recognition Method for Parkinson's Disease Patients. In Proceedings of the Pervasive Computing Technologies for Healthcare, Thessaloniki, Greece, 12–14 December 2023; pp. 229–246. [CrossRef]
6. Tao, L.; Wang, X.; Peng, X.; Yang, P.; Qi, J.; Yang, Y. Activity Selection to Distinguish Healthy People from Parkinson's Disease Patients Using I-DA. In Proceedings of the 2021 17th International Conference on Mobility, Sensing and Networking (MSN), Exeter, UK, 13–15 December 2021; pp. 66–73. [CrossRef]
7. da Silva, F.L. EEG: Origin and Measurement. In *EEG—fMRI: Physiological Basis, Technique, and Applications*; Mulert, C., Lemieux, L., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 23–48. [CrossRef]
8. Gandal, M.J.; Edgar, J.C.; Klook, K.; Siegel, S.J. Gamma synchrony: Towards a translational biomarker for the treatment-resistant symptoms of schizophrenia. *Neuropharmacology* **2012**, *62*, 1504–1518. [CrossRef] [PubMed]
9. Smailovic, U.; Jelic, V. Neurophysiological Markers of Alzheimer's Disease: Quantitative EEG Approach. *Neurol. Ther.* **2019**, *8*, 37–55. [CrossRef] [PubMed]
10. Kannathal, N.; Choo, M.L.; Acharya, U.R.; Sadasivan, P.K. Entropies for detection of epilepsy in EEG. *Comput. Methods Programs Biomed.* **2005**, *80*, 187–194. [CrossRef] [PubMed]
11. Maitin, A.M.; Romero Muñoz, J.P.; García-Tejedor, Á.J. Survey of Machine Learning Techniques in the Analysis of EEG Signals for Parkinson's Disease: A Systematic Review. *Appl. Sci.* **2022**, *12*, 6967. [CrossRef]
12. Chang, C.Y.; Hsu, S.H.; Pion-Tonachini, L.; Jung, T.P. Evaluation of Artifact Subspace Reconstruction for Automatic Artifact Components Removal in Multi-Channel EEG Recordings. *IEEE Trans. Biomed. Eng.* **2020**, *67*, 1114–1121. [CrossRef]
13. Gu, Y.; Li, X.; Chen, S.; Li, X. AOAR: An automatic ocular artifact removal approach for multi-channel electroencephalogram data based on non-negative matrix factorization and empirical mode decomposition. *J. Neural Eng.* **2021**, *18*, 056012. [CrossRef]
14. Islam, M.K.; Rastegarnia, A.; Yang, Z. Methods for artifact detection and removal from scalp EEG: A review. *Neurophysiol. Clin. Neurophysiol.* **2016**, *46*, 287–305. [CrossRef]
15. Jain, R.; Ganesan, R.A. Reliable sleep staging of unseen subjects with fusion of multiple EEG features and RUSBoost. *Biomed. Signal Process. Control* **2021**, *70*, 103061. [CrossRef]
16. Lee, S.; Hussein, R.; Ward, R.; Jane Wang, Z.; McKeown, M.J. A convolutional-recurrent neural network approach to resting-state EEG classification in Parkinson's disease. *J. Neurosci. Methods* **2021**, *361*, 109282. [CrossRef] [PubMed]
17. Ren, Y.; Yang, J.; Zhang, Q.; Guo, Z. Multi-Feature Fusion with Convolutional Neural Network for Ship Classification in Optical Images. *Appl. Sci.* **2019**, *9*, 4209. [CrossRef]
18. Wu, H.; Qi, J.; Yue, Y. Machine Learning-based EEG Signal Classification of Parkinson's Disease. In Proceedings of the 2023 IEEE 10th International Conference on Cyber Security and Cloud Computing (CSCloud)/2023 IEEE 9th International Conference on Edge Computing and Scalable Cloud (EdgeCom), Xiangtan, China, 1–3 July 2023; pp. 423–428. [CrossRef]
19. Gulay, B.K.; Demirel, N.; Vahaplar, A.; Guducu, C. A novel feature extraction method using chemosensory EEG for Parkinson's disease classification. *Biomed. Signal Process. Control* **2023**, *79*, 104147. [CrossRef]
20. Coelho, B.F.O.; Massaranduba, A.B.R.; Souza, C.A.d.S.; Viana, G.G.; Brys, I.; Ramos, R.P. Parkinson's disease effective biomarkers based on Hjorth features improved by machine learning. *Expert Syst. Appl.* **2023**, *212*, 118772. [CrossRef]
21. Espay, A.J.; Kalia, L.V.; Gan-Or, Z.; Williams-Gray, C.H.; Bedard, P.L.; Rowe, S.M.; Morgante, F.; Fasano, A.; Stecher, B.; Kauffman, M.A.; et al. Disease modification and biomarker development in Parkinson disease. *Neurology* **2020**, *94*, 481–494. [CrossRef]
22. Singh, A.; Cole, R.C.; Espinoza, A.I.; Brown, D.; Cavanagh, J.F.; Narayanan, N.S. Frontal theta and beta oscillations during lower-limb movement in Parkinson's disease. *Clin. Neurophysiol.* **2020**, *131*, 694–702. [CrossRef] [PubMed]
23. George, J.S.; Strunk, J.; Mak-McCully, R.; Houser, M.; Poizner, H.; Aron, A.R. Dopaminergic therapy in Parkinson's disease decreases cortical beta band coherence in the resting state and increases cortical beta band power during executive control. *Neuroimage Clin.* **2013**, *3*, 261–270. [CrossRef] [PubMed]
24. Khoshnevis, S.A.; Sankar, R. Diagnosis of Parkinson's disease using higher order statistical analysis of alpha and beta rhythms. *Biomed. Signal Process. Control* **2022**, *77*, 103743. [CrossRef]
25. Smrdel, A. Use of common spatial patterns for early detection of Parkinson's disease. *Sci. Rep.* **2022**, *12*, 18793. [CrossRef]
26. Dar, M.N.; Akram, M.U.; Yuvaraj, R.; Gul Khawaja, S.; Murugappan, M. EEG-based emotion charting for Parkinson's disease patients using Convolutional Recurrent Neural Networks and cross dataset learning. *Comput. Biol. Med.* **2022**, *144*, 105327. [CrossRef] [PubMed]
27. Anjum, M.F.; Dasgupta, S.; Mudumbai, R.; Singh, A.; Cavanagh, J.F.; Narayanan, N.S. Linear predictive coding distinguishes spectral EEG features of Parkinson's disease. *Park. Relat. Disord.* **2020**, *79*, 79–85. [CrossRef] [PubMed]
28. Kostrzewa, R.M.; Nowak, P.; Kostrzewa, J.P.; Kostrzewa, R.A.; Brus, R. Peculiarities of L-DOPA treatment of Parkinson's disease. *Amino Acids* **2005**, *28*, 157–164. [CrossRef] [PubMed]
29. Tao, W.; Kong, X.; Zhong, Y.; Chen, L. Automatic detection of abnormal EEG signals using multiscale features with ensemble learning—ProQuest. *Front. Hum. Neurosci.* **2022**, *16*, 943258. [CrossRef]
30. Singh, A.; Richardson, S.P.; Narayanan, N.; Cavanagh, J.F. Mid-frontal theta activity is diminished during cognitive control in Parkinson's disease. *Neuropsychologia* **2018**, *117*, 113–122. [CrossRef] [PubMed]
31. Rashmi, C.R.; Shantala, C. EEG artifacts detection and removal techniques for brain computer interface applications: A systematic review. *Int. J. Adv. Technol. Eng. Explor.* **2022**, *9*, 354–383. [CrossRef]

32. Pion-Tonachini, L.; Kreutz-Delgado, K.; Makeig, S. ICLabel: An automated electroencephalographic independent component classifier, dataset, and website. *NeuroImage* **2019**, *198*, 181–197. [CrossRef]
33. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Research on Automated Modeling Technology of Sluice Gate Project Based on BIM Technology

Jiedeerbieke Madiniyeti [1,2], Qiujing Zhou [2,*], Huijun Qi [1,3], Yang Chao [3] and Yu Zhang [3]

[1] College of Hydraulic and Civil Engineering, Xinjiang Agricultural University, Urumqi 830052, China; jedel@xjau.edu.cn (J.M.); qihuijun@hhu.edu.cn (H.Q.)
[2] Key Laboratory of River Basin Digital Twinning of Ministry of Water Resources, Beijing 100048, China
[3] College of Water Conservancy and Hydropower Engineering, Hohai University, Nanjing 210098, China; chaoyangccc@hhu.edu.cn (Y.C.); hhuhszy@outlook.com (Y.Z.)
* Correspondence: zhouqj@iwhr.com

Abstract: In order to improve the efficiency and accuracy of the modeling and design work of the sluice gate project, this paper proposes an automatic generation template of the sluice gate project with customized semantics and project layout scheme, aiming at realizing the rapid assembling of all kinds of components of the sluice gate project. In the construction process, this paper first starts from basic parametric modeling and proposes constraints as the basis of modeling. Subsequently, a template library framework is developed based on the constraints to ensure that the generated templates have a high degree of standardization and consistency. Finally, an efficient and flexible template library is successfully constructed by using the customized classes and functions of Revit API, which provides powerful technical support for the modeling and design work of sluice gate engineering. This achievement helps to promote the informationization and intelligent development of the water conservancy engineering industry, and its versatility and scalability also make it have a wide range of application prospects in other water conservancy engineering fields.

Keywords: sluice gate engineering; Revit API; parameterized family; template library

1. Introduction

With the rapid development of information technology and the increasingly complicated construction of water conservancy projects, the traditional design, construction, and management of water conservancy projects have been difficult to meet the actual needs of modern water conservancy projects. In this context, BIM (building information modeling) technology came into being and has shown great application potential in the field of hydraulic engineering. The core advantage of BIM technology is that it integrates a three-dimensional digital model of multi-stage information such as design, construction, and operation and maintenance. This model not only accurately expresses the physical form of hydraulic engineering, but also deeply integrates its functional characteristics, material properties, construction processes, and even operation and maintenance management and other all-round information. For water conservancy projects, BIM of information integration shows great potential and value, which promotes the improvement in design efficiency, the optimization of construction management, and the improvement in cooperation ability [1–3]. Although in water conservancy projects, BIM technology has shown some potential for application, it also faces a series of challenges and limitations, such as information exchange and sharing between multiple specialties, which cannot ensure the consistency and accuracy of data. Complex topography and dynamic water flow conditions increase the difficulty and accuracy of model construction. In addition, the non-uniformity of data interaction standards also hinders the smooth flow and utilization of information [4,5]. Due to the complexity and specificity of water conservancy projects, it often takes a lot of time and effort to optimize and integrate the model building process.

Individual parametric models are relatively simple to build, but the real challenge lies in the later integration process. The connection between each independent model is intricate and complex, and it is necessary to accurately match the connection relationship between terrain and structure. In addition, software operation, cutting, merging, and other fine adjustments not only require superior skills but also bring a heavy workload, as each step needs to be meticulous to ensure that the overall model exhibits seamless and efficient operation. This paper focuses on considering the problems and solutions encountered in the parametric modeling and integration process of complex buildings, the core reason for which is that the application of BIM technology or the practice of the digital twin concept cannot be separated from the most basic model parameterization and accuracy. Parametric modeling is the core of BIM technology which defines components and elements in a parametric way and establishes the relationship between parameters and models [6–8]. The fusion of parametric models and parameter rationality, which together ensure the efficient and accurate combination of components to form a complete and consistent model, is also an important aspect in the application of BIM technology.

The sluice gate project is a class of low-head hydraulic structures built on rivers, lakes, channels, seawalls, or reservoir banks [9,10]. Multiple studies have shown that it is now possible to create parametric drive models of buildings related to sluice gate engineering in Revit 2020 software and effectively perform complete stitching of resources and models [11–16]. The earliest study by Ren et al. [11] proposed a system of BIM integrated into a design solution by taking a small pump gate project as an example, applying BIM technology to realize the modules of modeling, collaboration, collision, drawings, and reports in the design stage. Jiang et al. [12] performed parametric modeling of a sluice gate, ad extraction of the model's quantities, and preparation of project estimates by means of secondary development, which further confirmed the practicality and applicability of BIM technology. Jiang et al. [13] showed that combined with the characteristics of different cross-section types of farmland channels, the Revit API interface is used to carry out the secondary development of Revit software, and intelligent modeling and three-dimensional reinforcement methods for farmland channels are proposed. Yuan et al. [14] analyzed the deficiencies in the construction safety process using BIM technology and a safety knowledge base for several hydraulic buildings. In addition, there are studies on similar buildings, such as Zhang et al. [17]. A new geological model building method was proposed by adopting a topological geometry modeling strategy and using parametric modeling. Geng et al. [18] established a water conservancy project construction progress management system based on building information modeling and GIS, which provides a three-dimensional dynamic visualization environment for water conservancy projects, and fine control of a water conservancy project's volume and progress plan management. Zhang et al. [19], with the combination of BIM technology and WebGIS, proposed an extended IFC export method for hydropower projects and a data exchange method between extended IFC and batch 3D Tiles, and the validity of the method was verified through a case of design modification of a crushed concrete gravity dam. Gong et al. [20] developed an automated assembly program that can be used for tunnel tube sheets, which also involves different levels of parametric framework construction and application. From most of the literature, it can be seen that at present, the fully parametric sluice model is basically solved, and even the accuracy of the model has reached a high degree, but the research on the accurate construction and automated assembly for the complex constraint relationship between models is still insufficient; complex buildings still take a lot of time to assemble, and some of the models may lose the comparative advantage of efficiency compared to two-dimensional modeling due to collision problems. In particular, how to efficiently and accurately define the physical, geometric, and functional constraints between different model components in sluice gate engineering, as well as how to realize the intelligent and automated assembly and integration of these models, is still a challenge to be solved. Therefore, it is necessary to study the efficient integration method of the model and explore new ways to effectively improve modeling efficiency and quality. Taking a sluice gate

project as a case study, this paper aims to reveal the parametric modeling and efficient integration of the sluice gate model, and with the help of Revit's secondary development means, it further proposes the parameter semantics of each component and an automation arrangement scheme to simplify the modeling process and improve the efficiency of modeling, so as to meet the urgent needs of the water conservancy engineering field for efficient and accurate modeling.

2. Revit Secondary Development Technology

2.1. Revit API Toolkit

Revit software is a series of software designed for building information modeling (BIM) developed by Autodesk, through which Revit can effectively solve the problem of model-driven design and improve the efficiency and quality of design [7,21]. In order to meet specific project requirements, the functionality of Revit software needs to be extended and customized, usually by using Revit API for in-depth development or writing Revit built-in scripts for automation. Also, to simplify repetitive tasks, the Revit macro editor is utilized. The majority of Revit external applications in use today are installed on the platform through a variety of secondary development routes that provide a wide range of customization options and enhancements [22,23]. Among them, Revit API, as the most direct and powerful development tool, provides developers with the ability to interact with Revit's core functionality in depth and is the preferred method to achieve customized functionality extensions [24,25]. Revit API contains a rich set of namespaces, classes, and their functions and attributes, which are resources that provide strong support for secondary development and greatly reduce the difficulty of development, enabling developers to extend and customize the functionality of Revit software more efficiently. In the development of the template library, a variety of Revit API built-in functions and classes are used, such as using the methods in the Document class to create, delete, or modify elements, and using the Transaction class to encapsulate changes to ensure that they are executed in a single transaction. For view manipulation, the View class and its subclasses are used to obtain, create, or modify views. While built-in functions can address the implementation of basic functionality, they are not designed to be "plug and play" solutions for specific applications or template libraries. Therefore, when developing template libraries for automated family assembly, custom classes or functions are often needed to meet specific functionality requirements to support automated family alignment and accurate model placement. Table 1 shows the main Revit API custom functions or classes used to build template libraries.

Table 1. Main functions or classes of template library.

Name	Clarification
TemplateArrangeView	Template layout view
TemplateArrangeVm	Template layout view model
DrawingCanvas	Customized drawing boards
TemplateConfigView	Template setting view
WindowPlaceBottomPlate	Placing the baseboard window
PlaceBottomPlateVm	Placement of the base plate view model
IPlaceLockChamberBottomPlate	Docking board interface
WindowPlaceWingWall	Placement of wing wall windows
PlaceWingWallVm	Placement of wing wall view model

Revit API provides two ways to extend its functionality: one is the IExternal Command interface, which is used to define external commands, in which the user clicks on the added command button to launch the corresponding command generated by the secondary

development; and the other way, the IExternal Application interface, is used to create an external application, i.e., to add menus or toolbars, which is automatically executed when Revit starts and shuts down. Since IExternal Application provides a higher level of extensibility, it is common to use IExternal Application to overload OnStartup( ) and OnShutdown( ) functions to realize the development of functions.

2.2. Interface Development

WPF (Windows Presentation Foundation) is a Microsoft graphical presentation system based on the .NET Framework that provides a rich set of controls and layout options that enable developers to easily create beautiful and easy-to-use user interfaces [26,27]. WPF introduces data-driven core concepts such as DataDriven, DataBinding, and the XAML language which significantly separate the user interface (UI) design from the underlying logic. At the same time, WPF utilizes the DirectX underlying interface, its powerful graphics vector-rendering engine compared to the previous generation of CDI/GDI + programming model, to achieve a qualitative leap. Therefore, WPF in the field of interface development has occupied a significant position. The MVVM design pattern is a software architecture pattern for developing GUI applications that separates the user interface (View) from the business logic (Model) and introduces an intermediate layer, called ViewModel, to coordinate the interactions between View and Model [13,28]. The MVVM pattern makes full use of the XAML in WPF and the DataBinding features, layered to achieve functional modularity and reduce the coupling between individual modules, so that the program's architecture is clear and the functionality is clear, while facilitating collaborative team development. When developing a Revit API-based automatic family template library in MVVM mode, the ViewModel layer calls Revit API to realize the interaction with BIM, performs the reading, modification, and updating of BIM model, and ensures that the application logic is seamlessly integrated with the Revit environment. The View layer is responsible for the display and interaction of the user interface through rich interactive elements and collects user input feedback to be processed by the ViewModel layer so as to realize a clear separation of user interactions and business logic. The Model layer manages business data and logic, organizes and manages data, stores core information such as template configuration and parameter settings, and provides solid data support for the template library. The three work together to realize the functional requirements and ensure the effective separation of interface design and business logic. The use of MVVM mode needs to be fully understood and prepared for the development environment and the need to call the Revit API classes or functions to ensure that the development process can efficiently use the API to achieve the operational requirements of BIM. The specific development process is shown in Figure 1.

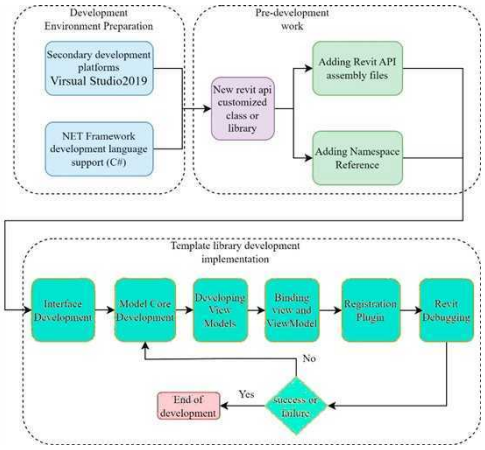


Figure 1. Revit software customization development process based on MVVM pattern.

3. Sluice Engineering Template Library Creation

3.1. Revit Parametric Design Routine Flow

Parametric design, which is based on a set of predefined parameters to create, modify, and control a design model, represents a more intelligent and efficient way of thinking about design [29,30]. The set parameters can be geometric dimensions, material properties, mechanical properties, etc., which together define the characteristics and behavior of the model. By adjusting the parameters, the model can be rapidly iterated and optimized to meet different design requirements and constraints. In parametric design, the dimensional parameter-driven process is a core concept implying precise control by tuning specific dimensional parameters [31,32]. Figure 2 shows the conceptual flowchart of conventional parametric design, which shows that the first step is to preset the basic parameter framework of the model and then achieve the flexible modification of model dimensions and attributes by continuously adjusting these parameters. This process is repeated until the model fully meets the requirements, which also reflects the advantages of efficient iteration and precise control of parametric design. For the modeling of Revit-driven models of conventional hydraulic buildings, one can refer to a study written by the authors [16]. In this study, the core of parametric modeling focuses mainly on the 3D semantics of sluice gate engineering components and the definition of their constraints.

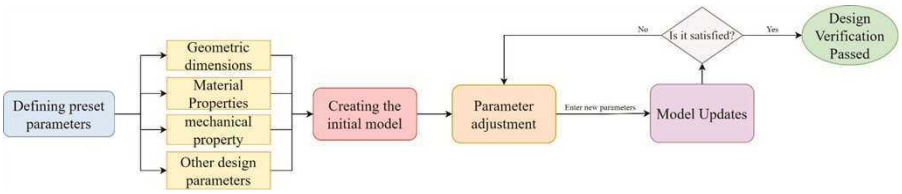


Figure 2. Conceptual flowchart of parametric design.

3.2. Definition of Constraints

In the process of parametric modeling of the sluice gate project using Revit, given that the overall model is pieced together by multiple families of components, it is necessary to determine the constraints between the components before building to ensure precise articulation between the components. By determining the constraint relationships, the modeling sequence of the components can be further judged and optimized. Subsequently, models or auxiliary graphics such as geometries, reference planes, and dimensional annotations are created. Next, the key step of model parameterization is carried out by associating the reference plane with the model boundary and closely associating the dimension annotation with the parameters. For example, for the gate pier model, according to the spatial location of the relationship between the components and the need for structural stability, the bottom of the gate pier must be strictly constrained in the elevation of the base plate and the use of fixed constraints must be fixed. Therefore, when building the gate pier model, parameter adjustment of the height of the gate pier should be limited to the upward extension, and its bottom elevation should be kept constant, as any change may affect the accurate position of the base plate. Through the above analysis of the gate pier, it can be argued that the components need to be set between the rigorous articulation relationship in order to ensure the accuracy of model construction while achieving the flexibility of the adjustment of parameterization. Figure 3 is a schematic diagram of the main constraints between the components of the sluice gate project.

Special attention should be paid to the fact that, according to the constraint relationship of each component, in order to facilitate the overall layout and avoid constraint conflicts, it is necessary to build the model from the most constrained component first. The order of building components should not only consider the articulation between components, but also consider the stability and integrity of the overall structure. As shown in Figure 3, the gate pier, as the most constrained component, should be built first. On this basis, the bottom plate and gate can be further parameterized through modeling to ensure that they are smoothly connected with the gate pier. After completing the modeling of the

bottom plate and gate, the dissipative pool, cover, wing wall, and other parts are gradually modeled to ensure that the entire model construction process is orderly and efficient.

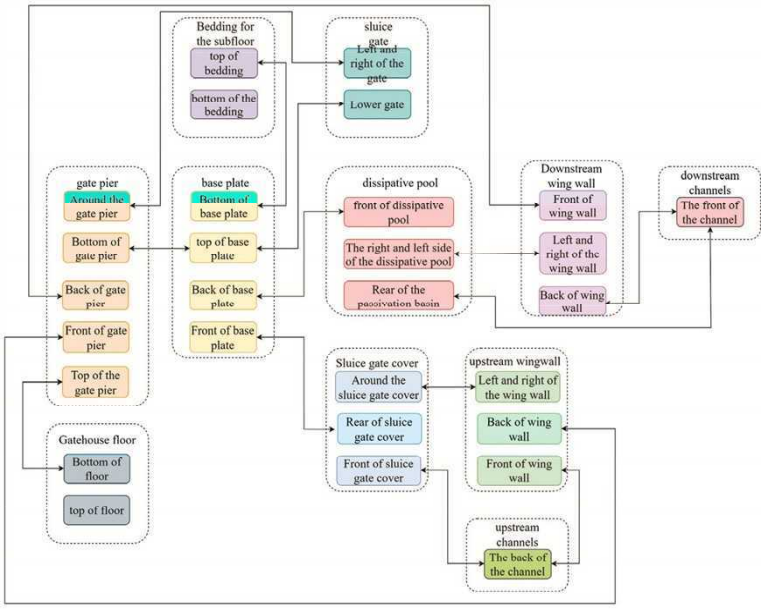


Figure 3. Schematic diagram of main constraints of each member of sluice project.

3.3. Custom Semantics and Project Layout Scheme Definition

Before the development of the template library, the original abstract project layout is constructed by humans, using the data storage technology to achieve its concretization, and then the "custom semantics and project layout" is created. Therefore, the essence of the sluice gate project template library is the "custom semantics and project layout plan" for automated modification, saving, and transfer. "Customized semantics and project layout" are based on JSON format, which is used to record component information and layout. In this project, each individual component of the lock project is abstracted into the class TemplateArrangeData, which contains six generic pieces of information, namely name, location, size, whether it is slit or not, load family, and load family type. For specific components, such as base plates, in addition to the generalized data, several pieces of information are included in the dimensions, such as length, thickness, width, bedding thickness, toothed wall thickness, toothed wall angle, etc., with different types of dimensions defined for each component. The parameters of the main setup semantics are shown in Table 2. The semantic representation of specific constructs is as follows:

```
public class TemplateArrangeData : BaseViewModel
{//Name
public BuiltInCType Name { get; set; }
//Location
public Point Location { get; set; } = new Size(152.36);
//Size
public Size Size { get; set; } = new Size(4000,500,3200,100,300,30,100,300,30);
//HasParting
public bool HasParting { get; set; }
//Family Name
public string Family { get; set; }
FamilySymbol Name.
public string FamilySymbol { get; set; }}.
```

In the preliminary parametric modeling process, the constraints and articulation conditions between the components were studied in detail, and a set of construction sequences were developed, which are, in order, gate piers, bottom plates, gates, dissipative

pools, decks, wing walls, etc. The components were then serialized into JSON files with the file extension "proj". After these components were set up through the human–computer interface, they were serialized into JSON files with the suffix "proj". Each project layout plan contains a Name field, which is named by the user when saving. At the same time, the "Data" field records the TemplateArrangeData array and records the content of the project layout.

Table 2. Setting of main parameters of each component.

	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6
gate pier	length	height	height	depth of Tank 1	length of Tank 1	distance of Tank 1 from upstream
base plate	height	length	height	thickness	bedding thickness	angle of tooth wall 1
sluice gate	height	height	thickness	border width	rib width	rib thickness
dissipative pool	length	height	thickness	plane diffusion angle	downslope Height	height of tooth wall No 1
wing wall	front width of base plate	thickness of base plate	back width of base plate	wall width	wall height	wall chamfer width

3.4. Module Development

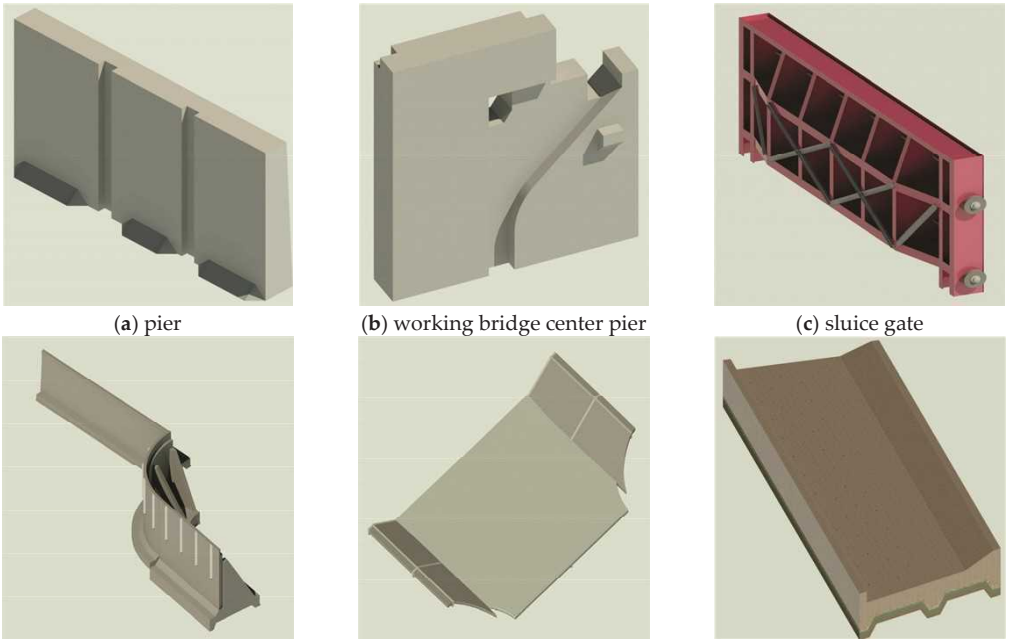
Based on the "custom semantics and project layout scheme", the process of realizing the rapid assembly of the template library can be divided into two steps. In the first step, on the basis of the three-dimensional model of the sluice gate, the three-dimensional semantic model and the plane pre-arrangement design plan are imported to generate the project layout plan file (*.proj), and the second step is to instantiate the model in the project space according to the file and assemble the model of the whole project. The instantiation operation is generally a step-by-step operation, according to the modeler's modeling habits of step-by-step instantiation, and each step can modify the parameters of the components according to the needs of the gradual assembly of the model. The template library is fully compliant with the "custom semantics and project layout scheme" file, and instantiation is performed silently. In this form, the system automatically reads the semantic information and completes model instantiation according to the preset rules and parameters. The entire process requires no manual intervention from the user, which greatly improves work efficiency. The generated "custom semantics and project layout scheme" file will be saved in the data folder of the platform root directory, which can be modified and reassembled after engineering optimization. Therefore, the function of the template library is mainly centered on four core aspects, which are as follows: (1) Support for floor plan layout pre-design. After defining the data semantics and business logic in the Model layer, it is in the View layer so that the user can carry out preliminary layout planning. (2) Support step-by-step assembly. Define the adjustment parameter mechanism in the Model layer to allow users to modify the parameters of building blocks in each step to meet specific needs. (3) Support silent assembly. Define instantiated interaction logic in the Model layer to provide the automated assembling process. (4) JSON-based persistence of the project layout scheme. Support importing/exporting JSON configuration files to load/save the project layout scheme, which facilitates the sharing and reusing of the design results.

4. Project Case

4.1. Project Overview

The Xingou River Extension and Dredging Project is a project approved by the State Council for the comprehensive improvement in the water environment in the Taihu Lake Basin, and is also one of the key projects for the implementation of the flooding of the basin to the north of the Yangtze River in the Taihu Lake Basin Flood Control Plan. The object

of this study is the construction of a sectional gate with two holes of a net width of 12 m and a gate bottom plate with a top elevation of −0.12 m, using a reinforced concrete dock structure, with a rising horizontal plane steel gate. Figure 4 shows the main structural diagram of the sluice gate, which shows the bottom plate and mat structure with toothed walls, and the dissipation pond is also set with multiple toothed walls and is indented on the left and right sides. The upstream wing wall structure has a curved shape, while the upper part of the gate pier is skillfully designed with construction details that are closely connected with the working bridge and the gate chamber; so, the overall structure of the sluice gate is relatively more complicated. The main building level of the gate is level 3, the secondary building is level 4, the peak acceleration of ground vibration is 0.1 g, and the basic intensity of any corresponding earthquake is degree V.



(**a**) pier (**b**) working bridge center pier (**c**) sluice gate

(**d**) wing wall (**e**) berms (**f**) dissipation pool

Figure 4. Partially parameterized family models.

4.2. Parametric Modeling

The sluice gate consists of several components, and in order to ensure that the model can be efficiently spliced and maintain the integrity of the whole structure, the main model is divided into three partitions, namely the gate chamber section, the upstream connecting section, and the downstream connecting section, to create the model. Through the constraint framework system set up in the previous stage, the geometry, reference planes, and parameters were created in detail, and the association between parameters and annotations, reference planes, and models was realized. A total of 42 parameterized custom families were created without counting the system families, and each family was carefully designed and optimized to meet the complex needs of the sluice model, and the specific families created are shown in Table 3. Among them, the construction of the handrail arc wing wall family is relatively complex, as each wing wall family is a combination of the wing wall as well as the main body, handrail array model, triangular angle, trapezoidal tooth chop, wave protection wall, bedding and other ancillary components. The wing wall families use adaptive conventional model templates, and adaptive points can be created by modifying the reference points. The geometry drawn by capturing the adaptive points will produce adaptive components. During the construction process, for model flexibility, the wing wall body is divided into two parts: the base plate and the vertical wall. The base plate is further divided into the front toe plate and the back toe plate, but there is no dividing line in the middle, and the height and width of the divided model are used as the

reference variables. The handrail array is a model made up of a single handrail arrayed on a segmentation path and controlled by parameters such as upper and lower bottom widths, width, and height. The other families of models are usually relatively simple to model, and the parameter settings and controls are more intuitive and easier, so they are not described in detail. Figure 4 shows a partially created parametric family model.

Table 3. A parameterized family breakdown of the Shiyan sluice gate.

Triangular shear model	Triangular stiffening rib based on circular arc (right)	Linear cavities	Center pier
Pedestrian bridge	Triangular stiffening rib based on circular arc (left)	Circular toothed curb	Center pier trim
Gate chamber base plate	Triangular stiffener rib based on straight line (right)	Concrete berm (upstream)	Working bridge center pier
Arc handrail	Triangular stiffening rib based on straight line (left)	Concrete berm (downstream)	Side piers
Waveguard	De-stressing pool	Elevation gate	Side pier trim
Circular wing wall	Cow leg	Wing wall shear model	Working bridge side piers
Circular wing wall bedding	Straight line bedding toothed cantilever	Footbridge	Upstream
Rounded corners	Straight line handrail	Side pedestrian bridge	Downstream
Straight corners	Linear waveguard	Drainage holes (surface-based, cross-alignment)	Rounded steel parapet (right)
Straight wing walls	Chamfering of linear waveguards	Straight steel parapet	Rounded steel parapet (left)
Straight wing wall bedding	Concrete bottom protection		

4.3. Application of the Template Library

The template library mainly realizes the extension of the functional area through the external application (IExternal Application) and invokes the external command to display the interactive window for each model parameter input. Therefore, on the basis of the "custom semantics and project layout scheme", relying on the Visual Studio platform, combined with the NET Framework, using C# programming language, and combined with the above mentioned several custom classes and functions, we can build and extend the functionality of the template library to achieve richer and more efficient family splicing and management. Here, we take two representative complex components, the wing wall and base plate, as our main focus to elaborate on the specific implementation of the template library. TemplateArrangeView is an XAML file of a template layout page which is rendered directly into a template layout view. Based on the MVVM programming model, the view has a unique corresponding view model TemplateArrangeVm, and also contains a custom control DrawingCanvas. The TemplateArrangeVm view model covers the template layout view of the save, import, view, list, delete, add, edit, etc., operations. Therefore, through the cooperative work of TemplateArrangeView and TemplateArrangeVm, it can realize the efficient display of the template layout interface and the user interaction functions. DrawingCanvas, as a custom control in TemplateArrangeView, rewrites the Canvas control in the WPF standard, adds mouse operation functions, and is a special element in View. DrawingCanvas provides important APIs such as ClearCanvas, DeleteVisual, AddVisual, ReDraw, CenterVisuals, and so on, which are available to TemplateArrangeView, and provides powerful dynamic management functions for the TemplateConfigView, which is also a special element in View that can change the river direction and the number of sluice holes, and pre-read the "*.proj" custom project layout file to quickly load and configure the existing template layout. This saves time and effort in setting up the project from scratch. On the basis of the template layout and operation functions, WindowPlaceBottomPlate is used to place window templates, make further changes to labels and view models, and provide an interface for inputting calculation parameters. For bottom plate members, the properties of width and length can be regulated through the PlaceBottomPlateVm

view model. PlaceBottomPlateVm is a component of the ViewModel layer. During the instantiation process, IPlaceLockChamberBottomPlate placement logic is used to create elevations (LevelSet) for the bottom plate model, read model parameters entered by the user, locate operations, call the view window, place the model, and perform a series of operations. IPlaceLockChamberBottomPlate and the processing logic of the other classes is similar. For the wing wall component, WindowPlaceWingWall is the xaml file for placing the window of the wing wall. Since the wing wall is constructed by the self-adaptive conventional model, the placing window is a little bit more complicated, but it also provides a human–machine interface for inputting the parameters of the calculation. PlaceWingWallVm is the view model for placing the window of the wing wall. DrawWingWallVmParameter contains the path information (straight line, circular arc) and style information (cantilever, handrail, empty box) of the wing wall, and WingWallModelCurveCopyOptions indicates the copying method of the wing wall. Through the effective realization of the above functions, a complete and accurate overall model of the sluice gate can finally be flexibly pieced together, thus greatly improving the efficiency and accuracy of water conservancy project design. Figure 5 shows the specific assembly process diagram of the sluice gate.

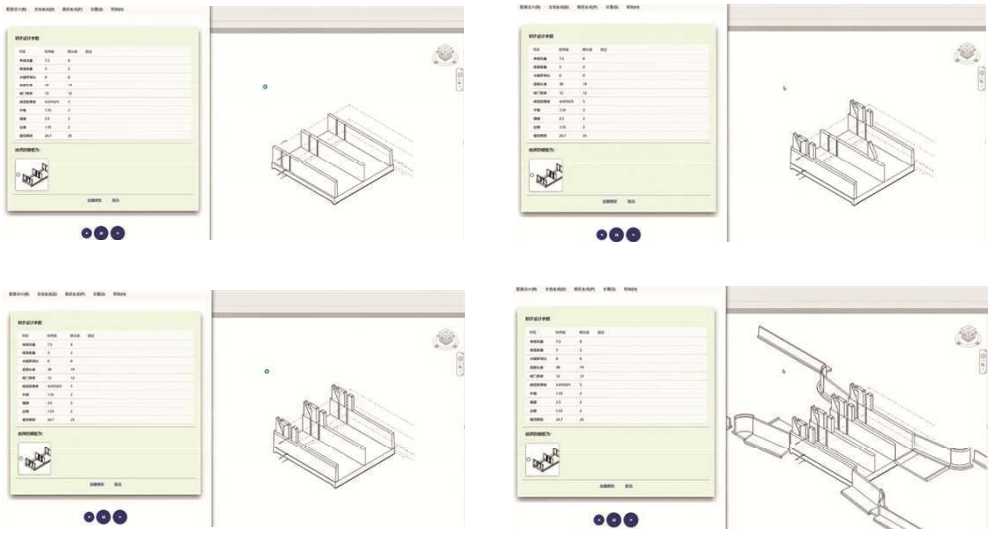


Figure 5. A diagram of the assembling process of the sluice gate.

The custom-developed template library is a set of procedures developed to solve the problems induced by model assembly, mainly through the "custom semantics and project layout scheme", achieving the intelligent assembly of sluice gate projects so as to solve the complexity and uncertainty of the interface between models. Therefore, it can be considered that on the basis of parametric family modeling, making full use of the Revit API function and combining this with the MVVM development mode, the automatic assembly process of sluice gate project models can be realized quickly.

5. Conclusions

The complex process of the parametric modeling and integration of sluice gate engineering is explored in depth, and an efficient and reliable solution for sluice gate engineering design is provided through the introduction of inter-model constraints and an automatic assembly template library based on Revit API. The outcome is specifically summarized in the following two aspects:

(1) Parametric modeling steps that consider constraints between components. The building sequence is determined based on the constraints, which ensures the reasonableness of the model assembly, and moreover, it improves the model's precision at the source and enhances the accuracy of the model's construction.

(2) The development of a template library for sluice modeling for automated assembly. Based on the development of Revit API, a template library for the automatic assembly of each component of the sluice project was created, which greatly simplifies the process of model construction and improves work efficiency. The template library for automated modeling not only solves the cumbersome procedures and model errors induced by model assembly, but also provides a strong tool support for future water conservancy project design. The template library is not only a revolutionary simplification of the traditional modeling process, but also a far-reaching layout for future water conservancy engineering design practice.

Author Contributions: Conceptualization, methodology, simulation, conceptualization, and writing original draft preparation, J.M. and H.Q.; supervision, project administration, and funding acquisition, Q.Z.; data collection, methodology, and writing review and editing, Y.C. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financially supported by the Open Research Fund, the Key Laboratory of River Basin Digital Twinning of Ministry of Water Resources (No. Z0202042022); the National Key Research and Development Plan of China (No. 2022YFC3005403); the National Nature Science Foundation of China (No. 52309151).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank the experts and engineers who helped during the modeling of this example and the development of Revit API.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jing, P. The Application of BIM Technology in Simulation Modeling of Concrete Dam Construction. *Agro Food Ind. Hi-Tech* **2017**, *28*, 1757–1761.
2. Zhou, Y.; Bao, T.; Shu, X.; Li, Y.; Li, Y. BIM and ontology-based knowledge management for dam safety monitoring. *Autom. Constr.* **2023**, *145*, 104649. [CrossRef]
3. Barazzetti, L.; Banfi, F.; Brumana, R.; Previtali, M.; Roncoroni, F. BIM from Laser Scans. . .Not Just for Buildings: NURBS-Based Parametric Modeling of a Medieval Bridge. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *III-5*, 51–56. [CrossRef]
4. Tan, Y.; Chen, W.; Lin, E.; Lin, P.; Zhou, T.; Zhou, M.; Liu, C.; Pei, L.; Liang, C.; Shang, C.; et al. Research and practice of multidimensional information model for super high arch dam construction. *J. Tsinghua Univ. Sci. Technol.* **2022**, *62*, 1884–1895.
5. Liu, D.; Chen, J.; Hu, D.; Zhang, Z. Dynamic BIM-augmented UAV safety inspection for water diversion project. *Comput. Ind.* **2019**, *108*, 163–177. [CrossRef]
6. Piras, G.; Muzi, F. Energy Transition: Semi-Automatic BIM Tool Approach for Elevating Sustainability in the Maputo Natural History Museum. *Energies* **2024**, *17*, 775. [CrossRef]
7. Sun, S.N.; Zhang, H.J. Research on the Application of BIM Technology in Water Conservancy Project. *J. Eng. Manag.* **2016**, *30*, 103–108.
8. Hariri-Ardebili, M.A.; Mahdavi, G.; Nuss, L.K.; Lall, U. The role of artificial intelligence and digital technologies in dam engineering: Narrative review and outlook. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106813. [CrossRef]
9. Si, H.Y.; Chen, M. Three Dimensional Structural Analysis of Complex Sluice Chamber Structures. *Appl. Mech. Mater.* **2012**, *170–173*, 1971–1976. [CrossRef]
10. Fu, Z.; Yan, Z. Stability analysis on a new type of floating sluice. *J. Hydraul. Eng.* **2005**, *36*, 1014–1018.
11. Ren, H. BIM technology-based 3-D collaborative integrated design scheme of pump-sluice project. *Water Resour. Hydropower Eng.* **2017**, *48*, 67–73.
12. Jiang, N.; Liu, Y. Application of sluice model information based on BIM and VB. *J. Water Resour. Water Eng.* **2018**, *29*, 145–150.
13. Jiang, G.; Tang, S.; Ma, T.; Zhang, Y.; Yang, B.; Huang, X.; Liu, Y. 3D Intelligent Modeling and Reinforcement Method of Farmland Canal Based on Revit. *J. Shenyang Agric. Univ.* **2023**, *54*, 579–585.
14. Yuan, Z.; Zhou, L.; Xu, C. Risk identification and application of pumping station construction safety based on BIM. *J. Drain. Irrig. Mach. Eng.* **2020**, *38*, 157–162.
15. Gan, L.; Liu, H.; Yan, Y.; Chen, A. Bridge bottom crack detection and modeling based on faster R-CNN and BIM. *IET Image Process.* **2024**, *18*, 664–677. [CrossRef]

16. Jiedeerbieke, M.; Niu, Z.; Li, P.; Zhang, Y.; Kuai, P.; Li, J. Dam safety visual monitoring system based on BIM technology and B/S architecture. *J. Drain. Irrig. Mach. Eng.* **2020**, *38*, 583–588.
17. Zhang, Y.; Zhong, D.; Wu, B.; Guan, T.; Yue, P.; Wu, H. 3D Parametric Modeling of Complex Geological Structures for Geotechnical Engineering of Dam Foundation Based on T-Splines. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 545–570. [CrossRef]
18. Geng, Z.; Li, D.; Liu, S.; Yu, H.; Ouyang, L. Implementation method and realization of BIM + GIS in construction progress dynamic scheduling of water conservancy projects. *J. Water Resour. Water Eng.* **2020**, *31*, 138–142.
19. Zhang, S.R.; Jiang, P.Q. Implementation of BIM plus WebGIS Based on Extended IFC and Batched 3D Tiles Data: An Application in RCC Gravity Dam for Republication of Design Change Model. *KSCE J. Civ. Eng.* **2021**, *25*, 4045–4064. [CrossRef]
20. Gong, J.; Bao, T.F.; Zhu, Z.; Yu, H.; Li, Y.T. BIM-based framework of automatic tunnel segment assembly and deviation control. *Undergr. Space* **2024**, *16*, 59–78. [CrossRef]
21. Abanda, F.H.; Byers, L. An investigation of the impact of building orientation on energy consumption in a domestic building using emerging BIM (Building Information Modelling). *Energy* **2016**, *97*, 517–527. [CrossRef]
22. Kensek, K.M. Integration of Environmental Sensors with BIM: Case studies using Arduino, Dynamo, and the Revit API. *Inf. Constr.* **2014**, *66*, e044. [CrossRef]
23. Chen, D.Y.; Fan, T.W. *Study of Applying API on Evacuation Route Assessing in BIM Modeling*; DEStech Publications, Inc.: Lancaster, PA, USA, 2015; pp. 81–84.
24. Zhang, Q.; Su, Q.; Yan, Y. Research on Calculating Quantity of Utility Tunnel with Revit Secondary Development. In *Advances in Transportation Geotechnics IV, Proceedings of the 4th International Conference on Transportation Geotechnics, Chicago, IL, USA, 23–26 May 2021*; Tutumluer, E., Nazarian, S., Al-Qadi, I., Qamhia, I.I., Eds.; Lecture Notes in Civil Engineering; Springer: Cham, Switzerland, 2022; pp. 63–73.
25. Su, G.; Kensek, K. Fault-detection through integrating real-time sensor data into BIM. *Inf. Constr.* **2021**, *73*, e416. [CrossRef]
26. Zhu, B.; Zhang, Y.; Zhao, Y.; Zhang, J.; Liu, Y. BIM-Based Intelligent Parametric Modeling Technology in Bridge Engineering. *Bridge Constr.* **2022**, *52*, 18–23.
27. Zhang, F.; Shen, Y.; Tan, L.; Jin, R.; Han, Y. Research on Self-adaptive Reinforcement Plug-in of Prefabricated Concrete Component Based on BIM. *KSCE J. Civ. Eng.* **2024**, *28*, 2588–2599. [CrossRef]
28. Zhou, M. MM Electrical Tool: A Tool for Generating Electrical Single Line Diagrams in BIM. Master's Thesis, University of Southern California, Los Angeles, CA, USA, 2019.
29. Asl, M.R.; Zarrinmehr, S.; Yan, W. Towards BIM-Based Parametric Building Energy Performance Optimization. In Proceedings of the 33rd Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), Cambridge, ON, Canada, 24–26 October 2013; pp. 101–108.
30. Wu, T.Y.; Jeng, T.S. Reforming Design Studios Experiments in integrating BIM, parametric design, digital fabrication, and interactive technology. In Proceedings of the 30th International Conference on Education and Research in Computer Aided Architectural Design in Europe, eCAADe 2012, Prague, Czech Republic, 12–14 September 2012; pp. 49–54.
31. Yan, W. WP-BIM: Web-based Parametric BIM towards Online Collaborative Design and Optimization. In Proceedings of the 35th eCAADe Conference, Rome, Italy, 20–22 September 2017; pp. 527–534.
32. Girardet, A.; Boton, C. A parametric BIM approach to foster bridge project design and analysis. *Autom. Constr.* **2021**, *126*, 103679. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

mathematics

MDPI

Article

Task-Importance-Oriented Task Selection and Allocation Scheme for Mobile Crowdsensing

Sha Chang *, Yahui Wu *, Su Deng, Wubin Ma and Haohao Zhou

Science and Technology on Information Systems Engineering Laboratory National University of Defense Technology, Changsha 410073, China; wb_ma@nudt.edu.cn (W.M.); haohaozhou@nudt.edu.cn (H.Z.)
* Correspondence: changsha18@nudt.edu.cn (S.C.); wuyahui@nudt.edu.cn (Y.W.)

Abstract: In Mobile Crowdsensing (MCS), sensing tasks have different impacts and contributions to the whole system or specific targets, so the importance of the tasks is different. Since resources for performing tasks are usually limited, prioritizing the allocation of resources to more important tasks can ensure that key data or information can be collected promptly and accurately, thus improving overall efficiency and performance. Therefore, it is very important to consider the importance of tasks in the task selection and allocation of MCS. In this paper, a task queue is established, the importance of tasks, the ability of participants to perform tasks, and the stability of the task queue are considered, and a novel task selection and allocation scheme (TSAS) in the MCS system is designed. This scheme introduces the Lyapunov optimization method, which can be used to dynamically keep the task queue stable, balance the execution ability of participants and the system load, and perform more important tasks in different system states, even when the participants are limited. In addition, the Double Deep Q-Network (DDQN) method is introduced to improve on the traditional solution of the Lyapunov optimization problem, so this scheme has a certain predictive ability and foresight on the impact of future system states. This paper also proposes action-masking and iterative training methods for the MCS system, which can accelerate the training process of the neural network in the DDQN and improve the training effect. Experiments show that the TSAS based on the Lyapunov optimization method and DDQN performs better than other algorithms, considering the long-term stability of the queue, the number and importance of tasks to be executed, and the congestion degree of tasks.

Keywords: mobile crowdsensing; task importance; Lyapunov optimization; double deep Q-network; action mask

MSC: 60K25; 68M20; 90C15

1. Introduction

Mobile Crowdsensing (MCS) [1] introduces a novel sensing methodology by leveraging intelligent devices embedded with a variety of sensors to gather sensory information. In comparison to traditional static sensor networks, MCS eliminates the prerequisite of pre-deploying a substantial sensor infrastructure, thereby offering significant benefits, including reduced costs, scalability, real-time capabilities, and enhanced efficiency. Consequently, MCS has found widespread application in domains such as smart city management, environmental surveillance, and data acquisition [2–5].

Typically, an MCS system architecture, as illustrated in Figure 1, encompasses a central platform collaborating with a multitude of participant devices [6]. The process is initiated when external applications requiring data transmit sensing tasks to the platform. The platform assumes the pivotal role of task mediation, meticulously evaluating both the specific requirements of the tasks and the current state of the participants. By undertaking

a comprehensive analysis, it formulates a strategic task allocation scheme to match tasks with the most suitable participants [7].

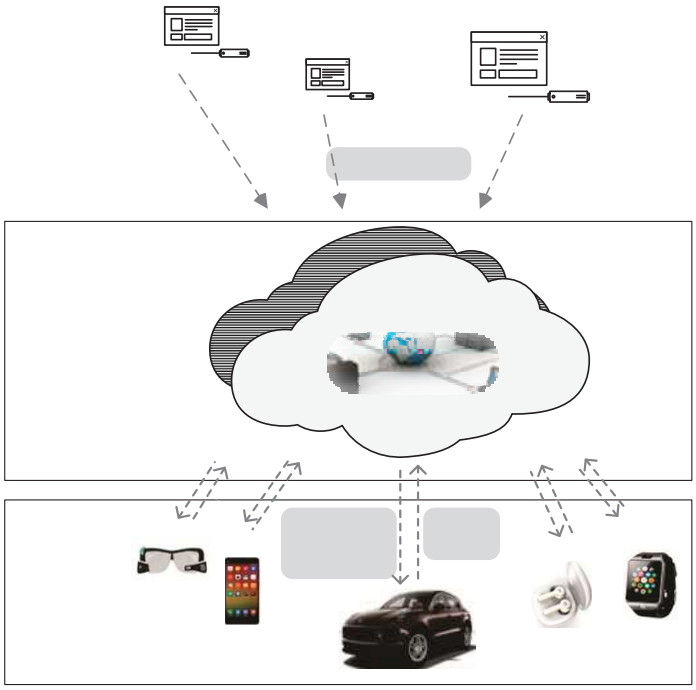


Figure 1. The composition of the MCS system.

Participants undertake the assigned tasks and subsequently forward the accumulated data back to the platform. Here, the platform undertakes the tasks of securely storing and processing the data and subsequently transmitting them to the requesting external applications and generating revenue. Thus, MCS effectively integrates the sensing capacity of participants with the diverse demands of data-reliant applications [8,9].

In Mobile Crowdsensing (MCS), tasks vary in their impacts and contributions to the overall system or specific objectives, leading to a differentiation in their levels of importance. Take environmental monitoring [10] as an illustration: detecting hazardous gas leaks may hold greater importance than routine air quality monitoring, given its direct implications for public safety. By identifying and prioritizing tasks that pose higher risks or possess greater importance, potential hazards and losses can be mitigated while enhancing the system's responsiveness and crisis management capacities.

Similarly, in military strategizing, the prioritization and importance of tasks exert a direct influence on higher-level decision-making processes. Tasks deemed highly important may demand data of superior quality or a higher completion rate. The gathering of vital data can arm decision-makers with timely and crucial information, thereby facilitating more efficacious decision-making [11,12].

Consequently, considering the importance of tasks during the task selection and allocation phases within MCS is of great significance. Given the usual constraints of finite resources, such as participant availability, energy, time, and financial budgets, prioritizing these resources for more crucial tasks ensures the prompt and precise collection of vital data. This strategic allocation enhances the overall effectiveness and performance of the MCS system.

In many studies concerning task allocation, the importance of the tasks is consistently recognized as an indispensable factor.

To guarantee the timely execution of urgent tasks, ref. [13] introduced a new multi-objective optimization problem based on the quality of service of urgent task-aware cloud manufacturing service composition. It gives a higher importance to urgent tasks in the

production process. It proposes two service composition methods based on vertical collaboration and speed selection to speed up task completion. Ref. [14] proposed a radar task selection method based on task importance. The method first sorts tasks according to their dwell times and priorities and defines a set of reward values according to the initial ranking. Then, the priority of the tasks is changed iteratively according to the reward and punishment strategy. Finally, the optimal sequencing is determined to schedule tasks. This method significantly improves the performance of task scheduling and shortens the time for task selection and allocation. Ref. [15] ranked different nodes in wireless rechargeable sensor networks through the importance of data transmission so as to distinguish the priority of node charging. The importance of data transmission is mainly determined by the deadlines of the tasks and the penalty values of the tasks. In order to minimize data loss, charging tasks are divided into early tasks and delayed tasks. Nodes with high importance and short deadlines are included in early tasks. Ref. [16] proposed a scheduling scheme that can be directly applied to actual wind turbine maintenance tasks. The authors considered the type of maintenance tasks, the importance of the maintenance tasks, the remaining time, the task maintenance time, and the waiting time before maintenance; determined the priority of different maintenance tasks; and constructed a scheduling model based on task priority.

To address the problem of multiple permanent node failures in multicore real-time systems, ref. [17] proposed two protocols based on task importance: a recovery time distribution protocol and a graceful degradation protocol. These two protocols are based on the importance of the tasks. For single-node failure and multiple-node failures, the task is delayed or discarded after the importance is degraded. Based on a cloud computing platform, ref. [18] proposed a new deadline allocation strategy with level importance. The task level importance is obtained by calculating the proportion of a task level in all task levels. Based on the level importance, the authors proposed a heuristic task scheduling method. This method can minimize the execution cost by effectively scheduling tasks under deadline constraints. Ref. [19] studied the relationship between the processing time of task execution and the importance of the task in mobile edge computing. The authors distinguish the importance of the tasks according to the processing time of task execution. The corresponding task priority is given according to the average processing time of task execution. Then, the tasks with different priorities are weighted to allocate computing resources to ensure that the high-priority tasks can obtain sufficient computing resources. Ref. [20] summarized the factors related to the importance of tasks in industrial software into three aspects: intrinsic importance, the longest time that processing can be delayed, and whether it is on the critical path. The importance of tasks is negatively correlated with the longest processing time and positively correlated with the other two factors. The authors proposed an online task scheduling algorithm based on task importance. The algorithm sorts and schedules all tasks in descending order of importance. In addition, this paper also establishes resource reservation, preemptive scheduling, and online adaptive adjustment methods, which further improve the efficiency and adaptability of the algorithm.

Currently, research on the significance of tasks in task selection and allocation within MCS remains scarce.

Ref. [21] set a weight for each task according to the importance of the task. The higher the weight, the more important the corresponding task. In this paper, the revenue obtained by the data requester is defined as the total weighted quality of all sensing data. By designing a reasonable task allocation scheme, the revenue of the data requester is maximized under the constraint of a limited budget. It should be noted that the weight in this paper is a relative value, which refers to the importance of a task relative to other tasks when the set of all tasks is determined. However, in practical applications, the arrival of tasks is asynchronous and uninterrupted, and the system cannot predict all tasks in advance. Therefore, this method is not suitable for long-term and online dynamic task allocation in the MCS system.

In the participatory MCS system, the allocation and management of some tasks are based on auctions. In auctions, there will be untruthful bidding and malicious participants, which will lead to the redundant assignment of tasks; that is, a task is assigned to multiple participants. This will not only waste resources but also reduce the actual execution rate of tasks. In order to solve this problem, ref. [22] associated the importance of tasks with the number of bidders. Unpopular tasks, that is, tasks with fewer bidders, will be given higher importance and will be assigned and executed first, thus significantly improving the completion rate of tasks. But, the author did not distinguish the importance of the task itself.

The assumption underlying this prior research is premised on having adequate participants ready to undertake tasks. Nevertheless, in practical deployments, MCS systems diverge from other task allocation frameworks, such as those in edge computing systems or industrial manufacturing systems. Unlike these, the participants in MCS systems are autonomous, free to engage or exit from the system at their discretion. This autonomy may result in scenarios where certain areas or periods face a lack of participants, consequently leading to a backlog of pending tasks. In addition, the current research mainly determines task selection and allocation methods based on the current state, without considering the impact of future arrival tasks on current decisions. Hence, this paper focuses on devising a forward-looking and long-term task selection and allocation scheme based on the importance of tasks when the tasks and participants are unknown. The main contributions of this paper encompass the following:

1. A task queue is established, and the Lyapunov optimization method is introduced to propose the task selection and allocation scheme (TSAS) in the MCS system. When there are many tasks but not enough participants, participants mainly perform tasks of high importance and stabilize the task queue by controlling the number of tasks entering the queue; when the number of tasks is small but participants are sufficient, participants can perform the backlog tasks in the queue. So, tasks of lower importance can also be executed, which not only improves the completion of tasks but also improves the utilization of resources. Therefore, the TSAS is a long-term and online scheme that can dynamically balance the relationship between the execution ability of participants, the stability of the task queue, and the importance of tasks.
2. According to different system states, namely, the length of the task queue, the number of participants, and the number of tasks arriving at the platform, the TSAS is dynamically determined. This scheme has certain predictive power and foresight on the influence of future states.
3. The traditional solution of the Lyapunov optimization problem is improved, the DDQN method is introduced, and action mask, iterative training, and other methods are proposed to accelerate the training process and improve the training effect.

The rest of this paper is organized as follows. Section 2 introduces the system model and objective function. Section 3 develops the TSAS and solves the problem based on Lyapunov optimization and the DDQN. Section 4 presents a series of experiments to evaluate the performance of the TSAS and compares it with the Classical Lyapunov optimization algorithm, QPA algorithm, and offline task selection and allocation algorithm. Finally, Section 5 concludes this article.

2. System Model and Problem Formulation

2.1. System Model

In order to facilitate the management and scheduling of the system and ensure the synchronization and consistency of data collection, continuous time is discretized into discrete equal time slots t. $t \in \{0, 1, 2, \ldots\}$. The number of tasks arriving at the platform in each time slot is different. As shown in Figure 2, $O(t)$ represents the number of tasks arriving at the platform in time slot t. $O(t)$ is i.i.d. over slots with $E\{O(t)\} = \lambda$, $0 \leq O(t) \leq O^{max}$. $O_j(t)$ refers to the jth task arriving at the platform.

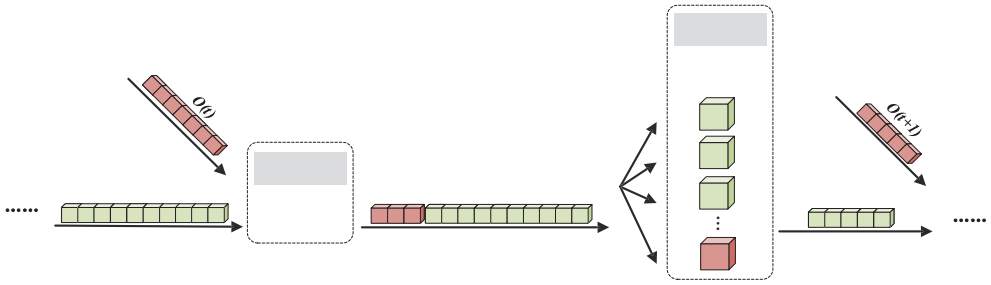


Figure 2. Schematic diagram of task queue of MCS system.

In order to establish a long-term task management method that can dynamically balance the number of participants and tasks, this paper designs a task queue to store the tasks to be performed. $Q(t)$ indicates the length of the task queue in the platform in time slot t. Each task has its own importance value. $F_s(t)$ refers to the importance of the sth task, which is determined by the data requester when generating the task. The larger the $F_s(t)$, the higher the importance of the task.

Task management includes task selection and task allocation. Task selection refers to the platform determining the tasks that can enter the task queue and be executed based on the state and number of participants in the previous time slot. Therefore, not all arriving tasks can enter the platform. When a task is dropped without entering the task queue, it indicates that the MCS system has a heavy load, a long task queue, and insufficient participants. The discarded tasks will be executed by other systems. This can ensure the stability of the task queue in the MCS system and avoid long waiting times for tasks. $o(t)$ represents the number of tasks entering the sensing platform in time slot t, and $0 \leq o(t) \leq O(t)$. Because the importance of tasks is different, in order to make full use of the limited sensing resources, the platform should select tasks to enter the task queue according to the importance of the task.

Task allocation refers to the scheme in which a platform assigns tasks from a queue to participants who execute them. Task allocation is not based on the order in which tasks enter the task queue, but rather, tasks are selectively assigned based on their importance.

i denotes different participants, $i \in \{1, \ldots, N\}$. $x(t) = \{x_{si}(t) | s \in \{1, 2, \cdots, Q(t)\}, i \in \{1, 2, \cdots, N\}\}$ indicates the task allocation scheme, that is, whether task s in the queue is allocated to participant i in time slot t. If $x_{si}(t) = 1$, participant i performs task s. If $x_{si}(t) = 0$, participant i does not perform task s. In each time slot, the same participant can only perform a single task. $r(t)$ is the number of tasks s that can be performed in time slot t:

$$r(t) = \sum_{s=1}^{Q(t)} \sum_{i=1}^{N} x_{si}(t) \tag{1}$$

$$0 \leq \sum_{i=1}^{N} x_{si}(t) \leq 1 \tag{2}$$

$$0 \leq \sum_{s=1}^{Q(t)} x_{si}(t) \leq 1 \tag{3}$$

$$0 \leq r(t) \leq N \tag{4}$$

At the end of time slot t, the remaining unexecuted tasks in the task queue will wait to be executed in the future. New tasks will arrive at the platform in the next time slot, and the platform will make the task selection decision again to select the number of tasks entering the queue. The dynamic change in $Q(t)$ follows the following formula:

$$Q(t+1) = max[Q(t) - r(t), 0] + o(t) \tag{5}$$

where $Q(t) \geq 0$. $Q(0) = 0$; that is, in the initial time slot, the length of the task queue is 0.

If $Q(t)$ satisfies

$$\lim_{t \to +\infty} \frac{E\{Q(t)\}}{t} = 0 \tag{6}$$

then $Q(t)$ is stable; that is, the task queue length is bounded. Therefore, the MCS system is stable.

2.2. System Utility

If the MCS system is stable, tasks that enter the queue can be executed. $U(t)$ represents the system utility in time slot t, which depends on the number of sensing tasks entering the task queue,

$$U(t) = \beta o(t) \tag{7}$$

where β represents the utility that the platform can obtain by performing unit tasks. β is a positive constant. The larger the $U(t)$, the more tasks that enter the task queue. A large $U(t)$ may cause the queue to continuously grow and become unstable. If $U(t)$ is too small, it may cause resource waste, meaning that some participants have no tasks to execute and remain idle. Therefore, the development of task selection plans needs to balance the stability of the task queue and the execution ability of participants.

2.3. Importance Level of the Tasks Being Executed

As mentioned earlier, when resources are limited, important tasks should be prioritized. $I(t)$ represents the total importance of all performed tasks in time slot t:

$$I(t) = \sum_{s=1}^{Q(t)} \sum_{i=1}^{N} F_s(t) x_{si}(t) \tag{8}$$

The larger the $I(t)$, the more meaningful the task selection and allocation scheme of the system. Therefore, the development of a task selection and allocation scheme needs to prioritize tasks with higher importance.

2.4. Problem Formulation

The problem to be solved in this paper is to maximize the overall system utility and the importance of the executed tasks through task selection and task allocation schemes while satisfying the stability of the task queue. Due to the long-term operation, dynamism, uncertainty, and nonlinearity of the MCS system studied in this paper, it is transformed into an optimization problem of system efficiency and task importance under time-averaged constraints. The objective function of this problem is as follows:

$$\min_{o(t)} \overline{U} = \min_{o(t)} \lim_{t \to +\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\{U(\tau)\} \tag{9}$$

$$\max_{x_{si}(t)} \overline{I} = \max_{x_{si}(t)} \lim_{t \to +\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\{I(\tau)\} \tag{10}$$

$$s.t.\ (1)(5)(6)(7)$$

3. Task Selection and Allocation Scheme

3.1. Lyapunov Optimization

In order to solve the optimization problem (9)–(10), this paper introduces queuing theory and Lyapunov optimization theory, which greatly reduces the difficulty of the problem. Lyapunov optimization theory [23] is about stochastic network optimization. The research of Professor Neely and his team on Lyapunov optimization is relatively mature and has been applied in many fields. Different from other static optimization algorithms,

the algorithm based on Lyapunov optimization theory is an online, dynamic, and adaptive algorithm. When the system state changes, there is no need for manual adjustment. The algorithm has a certain self-learning ability and can maintain progressive optimality. In addition, the algorithm based on Lyapunov optimization theory is simpler to implement, requires less prior knowledge, and has a certain decoupling ability, which can greatly reduce the complexity of the algorithm.

First, the Lyapunov function is defined as follows:

$$L(t) \triangleq \frac{Q(t)^2}{2} \tag{11}$$

$L(t)$ is the measure of task queue congestion. $L(t) \geq 0$. Generally, we cannot directly guarantee that the Lyapunov function is bounded, but we can design a method to keep the task queue backlog within a certain range so as to effectively control the congestion degree of the task queue and ensure the stability of the queue. Therefore, the Lyapunov drift function is defined as follows:

$$\Delta(Q(t)) \triangleq E\{L(t+1) - L(t)|Q(t)\} \tag{12}$$

The Lyapunov drift function represents the change in the Lyapunov function with time. Because the system is dynamic and uncertain, the Lyapunov drift function is defined as a conditional expectation in this paper. The expectation depends on the task selection scheme of the MCS system and is related to the system state.

The utility function is added as a penalty to the above drift function to obtain the drift-plus-penalty function:

$$\Delta_v(Q(t)) \triangleq E\{L(t+1) - L(t)|(Q(t))\} - VE\{U(t)|(Q(t))\} \tag{13}$$

V is a non-negative constant. By adjusting the value of V, the proportional relationship between queue stability and system utility can be controlled.

In order to ensure the stability of the queue and maximize system utility and the total importance of performed tasks, task selection decision $o(t)$ and task allocation decision $x(t)$ should be designed to minimize the drift-plus-penalty function. The objective Functions (6) and (9) is converted into

$$min\Delta_v(Q(t)) = min\ E\{L(t+1) - L(t)|(Q(t))\} - VE\{U(t)|(Q(t))\} \tag{14}$$

The classical solution of (14) is the mainstream and general solution for solving task selection and assignment problems in MCS at present. Because of the complexity of the Lyapunov optimization problem, it is difficult to obtain the minimum value of the drift-plus-penalty function directly. The classical solution of the Lyapunov optimization algorithm is to minimize the supremum of the drift-plus-penalty function so as to minimize the drift-plus-penalty function. However, in practical problems, the direct calculation of the supremum of the drift-plus-penalty function may be very complicated, sometimes even impractical. Generally, the accuracy and computational complexity will be balanced, and the problem will be solved by minimizing one of the upper bounds through appropriate scaling methods. This approximate solution to the drift-plus-penalty function can only obtain the approximate optimal solution, which is not the optimal task selection and allocation scheme and may even cause high task queue congestion. In addition, this method does not consider the impact of future states on the task selection scheme of the current time slot. Therefore, this paper introduces deep reinforcement learning to improve the defects in the classical solution of the Lyapunov optimization algorithm.

Reinforcement learning [24] is an important branch of machine learning. Reinforcement learning systems are usually composed of agents and environments. Agents can observe and interact with the environment, generate empirical data, and self-learn to maximize long-term returns so as to solve problems in different fields. In reinforcement learning,

data are generated in the process of agent interactions with the environment, which is unmarked data. When an agent executes an action in a certain state of the environment, it receives a reward from the environment to judge whether the action is good or bad; the state of the environment changes accordingly. In the next interaction, the agent makes decisions and acts according to the new state of the environment. But, the goal of the reinforcement learning problem is long-term; that is, the goal is to maximize the sum of long-term rewards, so agents cannot simply obtain the optimal action through a single reward. In general, empirical data are generated iteratively during the training process of agents. Data generation and agent training alternate. Agents obtain optimal strategies in different states through long-term training.

The Q-learning algorithm is a common reinforcement learning algorithm, which is usually calculated in table form. In the Q-learning algorithm, first, a table needs to be established to store the Q values of each action in different states, then collect training data during the interaction between agents and the environment, and then update the Q values in the table with experience playback. Obviously, this method is suitable for the case of a small state space and action space. When the state or action space is large, the table is not convenient for recording the Q value of each state and action, so it is more reasonable to use the function fitting method to estimate the Q value. The Deep Q-Network (DQN) [25] is based on the idea of function fitting and uses the powerful expression ability of neural networks to fit and estimate the Q value. The Double Deep Q-Network (DDQN) [26,27] is improved on the basis of the DQN. By separating action selection and value evaluation, the DDQN reduces the risk of overestimation and improves learning stability and final performance. These improvements make the DDQN a powerful tool for dealing with high-dimensional state space and complex decision problems. Because the environment in the system model in this paper is unknown and unpredictable, and the state space is large, it is difficult to use the traditional table-based reinforcement learning method to solve the problem. Therefore, the DDQN is used in the task selection stage, and a task selection decision algorithm based on Lyapunov optimization and the DDQN is proposed. The Lyapunov optimization method based on deep reinforcement learning no longer scales a certain upper bound of the drift-plus-penalty function but directly optimizes the task selection decision scheme through a reward function.

The DDQN-based MCS system is composed of the agent and the environment. The agent refers to the MCS platform. The environment not only includes the real physical object, namely, participants, but also includes the working mechanism and operating rules of MCS. The interaction process between the agent and the environment refers to the interaction process between the platform and participants. When tasks arrive at the platform, the platform determines which tasks can enter the platform, that is, makes the task selection decision, so as to minimize the drift-plus-penalty function. Then, the task allocation scheme is determined according to $o(t)$, $Q(t)$, and $F_s(t)$ to maximize the objective Function (10), which assigns the task to the participants. The participants complete tasks and upload data to the platform. At this time, an interaction between the agent and the environment is completed, and experience data are generated for the agent to learn and train.

Therefore, the objective Functions (9) and (10) is solved in two stages:

1. In the task selection stage, the goal is mainly to solve the problem below:

$$\min_{o(t)} \Delta_v(Q(t)) = \min_{o(t)} E\{L(t+1) - L(t) | (Q(t))\} - VE\{U(t) | (Q(t))\}$$

$$s.t.\ Q(t+1) = max[Q(t) - r(t), 0] + o(t)$$

$$r(t) = \sum_{s=1}^{Q(t)} \sum_{i=1}^{N} x_{si}(t)$$

$$U(t) = \beta o(t)$$

2. In the task allocation stage, the goal is mainly to solve the problem below:

$$\max_{x_{si}(t)} I(t) = \max_{x_{si}(t)} \sum_{s=1}^{Q(t)} \sum_{i=1}^{N} F_s(t) x_{si}(t)$$

The working mechanism of the task selection stage and allocation stage in the MCS system based on Lyapunov optimization and deep reinforcement learning is shown in Figure 3. The specific algorithm is introduced in the below sections.

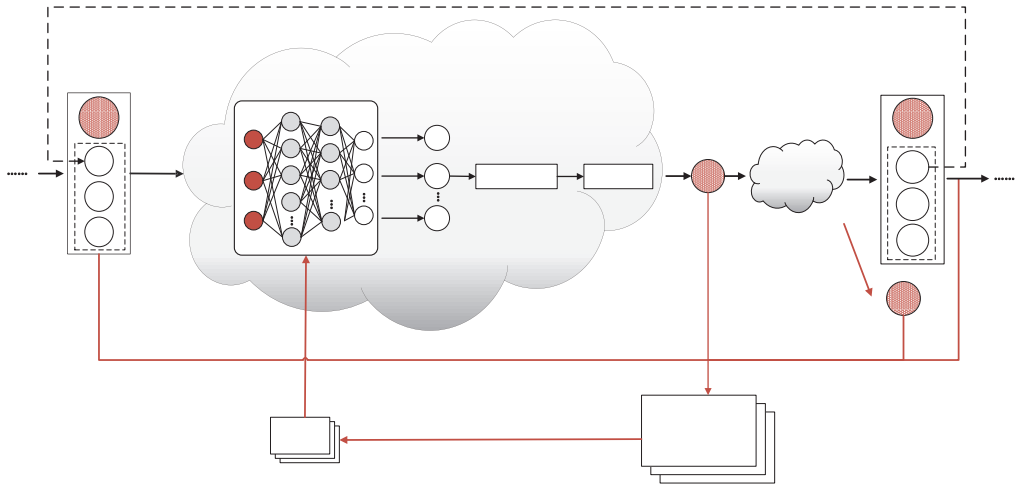


Figure 3. The working mechanism of the task selection and allocation scheme in MCS.

3.2. Task Selection Stage

In the task selection stage, the DDQN and Lyapunov optimization are used to make the task selection decision. The action-value network calculates all the action values in different states so as to obtain the optimal actions, that is, to determine the number of tasks entering the task queue. The state, action, reward function of the algorithm, and parameter update mechanism of the action-value network are introduced in detail below.

3.2.1. State S(t)

The determination of state $S(t)$ mainly considers the factors that affect the action value in the MCS system, including the number of tasks reaching the platform $O(t)$, the length of task queue $Q(t)$, and the number of participants $N(t)$. $S(t) = [O(t), Q(t), N(t)]$. There are different states in different time slots. The state space is denoted by $S \triangleq \{S(t)\}$.

$S(t)$ is dynamic and uncertain. On the one hand, the action decided by the agent is uncertain. When the agent decides the action and acts on the environment, the state of the environment may also change. On the other hand, the next state is usually determined by the current state and action, but the next state cannot be completely predicted and determined. Obviously, $O(t)$, $Q(t)$, and $N(t)$ are dynamic and uncertain. The state space S is very large. The traditional reinforcement learning method in the form of tables cannot solve this problem, so the DDQN is suitable for the platform to make decisions.

3.2.2. Action a(t)

Action $a(t)$ refers to the number of tasks entering the task queue, which is the task selection decision made by the agent in time slot t. $a(t) = o(t)$. $0 \leq a(t) \leq O(t)$. The action space is denoted by $A \triangleq \{a(t)\}$. Action values are different in different states. The optimal action is the action with the greatest value, that is, the optimal solution of the objective Function (14).

3.2.3. Reward Re(t)

As shown in Figure 4, after action $a(t)$ is executed in time slot t, the environment returns a reward to the platform, which is determined by the reward function, $Re(t)$. The sum of current and future rewards is the return. The goal of the DDQN is to find a strategy to maximize the expectation of the return. Therefore, the reward function directly affects the results of reinforcement learning.

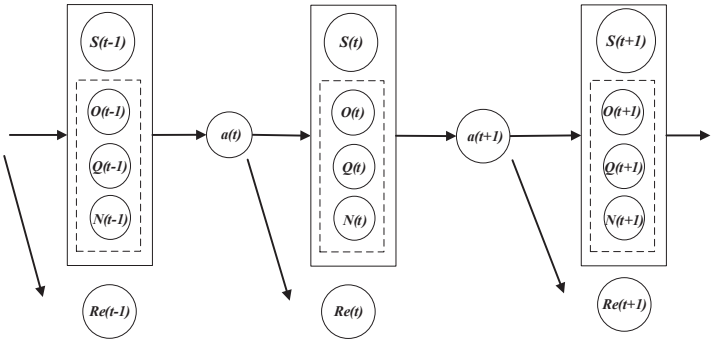


Figure 4. Flow chart of state, action, and reward.

The objective Function (14) is to maximize the system utility and stabilize the task queue. Therefore, the reward function is defined as follows, namely, the weighted sum of the Lyapunov drift function and system benefits:

$$Re(t) = -\left(\frac{Q(t+1)^2}{2} - \frac{Q(t)^2}{2} - V\beta o(t)\right) \tag{15}$$

3.2.4. Action-Value Network Training and Parameter-Updating Mechanism

The action-value network is a neural network composed of an input layer, a hidden layer, and an output layer. The network parameter is ω. The input layer has three nodes, which input three variables of state $S(t)$, namely, $O(t)$, $Q(t)$, and $N(t)$. The network output is the values of all actions in state $S(t)$.

The red line in Figure 3 shows the training process of the neural networks. The task allocation algorithm based on the DDQN is different from the supervised machine learning process. The algorithm has no prior knowledge. In order to enable the neural network to produce reliable and accurate action values, the experience data generated by the interaction between the agent and the environment are stored for the training of the neural network. In Figure 3, state $S(t)$, action $a(t)$, reward $Re(t)$, and the next state $S(t+1)$ are the experience data generated in time slot t, which will be stored in the replay buffer. During training, experience replay is used to randomly sample training data tuples $(S(t), a(t), Re(t), S(t+1))$ in the buffer for the gradient descent training of neural networks. The experience replay not only can maintain the independence between training data but also makes it possible to use samples many times, improving the efficiency of experience data.

The network parameters of the DDQN are mainly updated through Temporal Difference learning (TD learning). In order to prevent the overestimation of the action value caused by bootstrapping using the same neural network, the DDQN sets up two networks to calculate action values, namely, the target net and evaluation net, as shown in Figure 5. They have the same network structure, but the network parameters are not identical. $q_{targ}(S(t), a(t))$ represents the value of $a(t)$ calculated by the target network in state $S(t)$. $q_{eval}(S(t), a(t))$ represents the value of $a(t)$ calculated by the evaluation network in state $s(t)$.

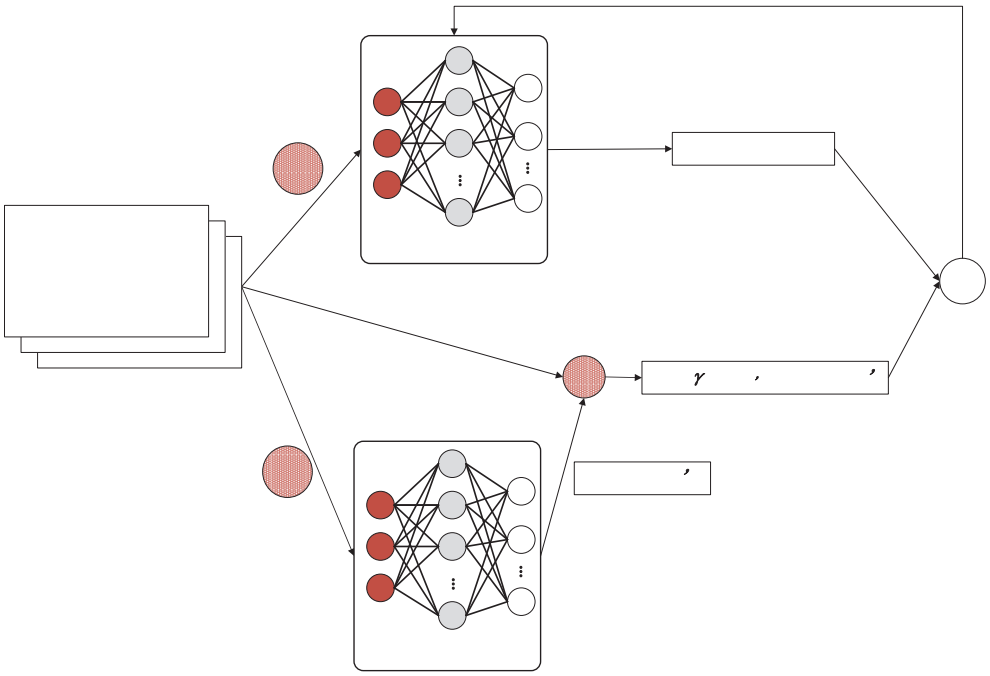


Figure 5. Training process of action-value network.

The TD target $\hat{q}(t)$ is

$$\hat{q}(t) = Re(t) + \gamma \cdot max_{a'} q_{targ}\left(S(t+1),\ a'\right) \tag{16}$$

The TD error $\delta(t)$ is

$$\delta(t) = q_{eval}(S(t),\ a(t)) - \hat{q}(t) \tag{17}$$

The loss function is defined as

$$Loss = \frac{1}{batch} \sum_{t=1}^{batch} \delta(t)^2 \tag{18}$$

The gradient descent method is used to minimize the loss of the evaluation network, and its parameters are updated. The parameters of the target network will remain relatively fixed. After the evaluation network has been trained a certain number of times, the parameters of the target network will be directly replaced with the parameters of the evaluation network for updating.

3.2.5. Iterative Training

In order to increase the state experience with small $Q(t)$ in the replay buffer, this paper designs iterative training for the infinite Markov decision problem.

In deep reinforcement learning, Markov decision problems are divided into finite periods and infinite periods, and their neural network training methods are different. When training the evaluation network of the Markov decision problem with a limited period, it is usually based on an episode for iterative training, so different states will be stored in the replay buffer and will appear repeatedly during training. But, the task selection and allocation problem in MCS is an infinite Markov decision process, and there is no termination state for one episode. Therefore, training is usually performed within an episode and stops after reaching the specified times. In the Lyapunov optimization problem, when the task queue reaches a stable state, it can no longer enter the unstable state of $Q(t)$. Since the experience data in the replay buffer are constantly updated during the training process, the amount of experience data after the queue is stable is far more

than the amount of data before stabilization. Such a lack of experience in the replay buffer will lead to the unsatisfactory performance of the TSAS before the task queue is stabilized.

Therefore, this paper designs an iterative training method that conforms to the Lyapunov optimization problem. The specific method is as follows: The training time is divided into three stages. The first stage is from the beginning of training until the loss is small enough, when the task queue is stable. In the second stage, the system state is reset to the initial state, that is, $Q(0) = 0$. When the queue reaches the stable state, it will be reset to the initial state again, which will be repeated many times with iterative training so as to fully train the queue state before stabilization. After several instances of resetting the system state and iterative training, the third stage is started; that is, the stable state is trained again until the end of the training. The training results are shown in Section 4.1.

3.2.6. Priority Experience Replay and Importance Sampling Weight

In order to further improve the training efficiency, this paper also introduces priority experience replay and importance sampling weights. In DRL, the traditional experience replay method stores the experience generated by agent interaction with the environment in the replay buffer and then randomly samples these experiences during the training. With the training of the evaluation network, $Q(t)$ gradually increases, resulting in an uneven distribution of states in the replay buffer. This random sampling may lead to some important experiences being ignored; for example, states with small $Q(t)$ cannot be fully trained, but some less important experiences are frequently sampled, resulting in a poor training effect. Priority experience playback solves this problem by assigning different priorities to experiences. These priorities are calculated based on the TD error, which represents the gap between the current estimated action value and the target action value. Experiences with high TD errors mean that they differ greatly from the expected returns, so they are considered more important and should be sampled more for training. Therefore, in priority experience replay, experiences are not sampled with equal probability but sampled according to their priority. High-priority experiences will be selected more frequently, but if they are completely relied on, the learning results may be biased toward the local optimal solution.

The sampling bias needs to be corrected by importance sampling weights; that is, when calculating the gradient, an importance weight is used to weight the loss of each sample to correct the bias. If an experience should occur more frequently under the target strategy, it will be given more weight during training; otherwise, it will be given less weight. The importance sampling weight helps balance the learning of different types of experiences by adjusting the contribution of each sampling experience so as to ensure that the learning process pays attention to both important experiences and low-priority experiences that may contain key information.

3.2.7. Action Mask

Action masking [28] is an important training skill in DRL. It improves the learning efficiency and performance by limiting the action selection of the agent and also simplifies the action selection. The action space is huge in the task selection and allocation of MCS, but only some of the actions are feasible in the current state. Without an action mask, agents may choose invalid actions, which will not only waste resources but also lead to instability in the training process. By action masking, agents will focus on feasible actions rather than invalid actions so as to reduce the time spent exploring invalid actions, improve the training efficiency, and speed up the training process.

The action mask method is designed as follows:

$$mask(a(S(t))) = mask(argmax_a(q(S(t),\ a))) = \begin{cases} q(S(t),\ a), & a\ is\ valid\ in\ S(t) \\ 0, & otherwise \end{cases} \tag{19}$$

$$a(t) = argmax_a(mask(a(S(t)))) \tag{20}$$

An action mask is usually used in policy gradient learning. It makes the gradient corresponding to the logarithm of invalid actions zero so as to prevent the network from selecting invalid actions. The DDQN belongs to value function gradient learning. In the above action mask method, the gradient of invalid actions will not change. However, the action mask method still works for the following reasons: in the DDQN, each time a tuple $(S(t), a(t), Re(t), S(t+1))$ is taken from the replay buffer, the parameters of the action-value network are updated in the following ways:

$$\omega_{new} \leftarrow \omega_{now} - \alpha \cdot \delta(t) \cdot \nabla_{\omega} Q(S(t),\ a(t); \omega_{now}) \tag{21}$$

where $\delta(t)$ is obtained from Equation (17). It can be seen from the above formula that when the state is $S(t)$, only the value function of action $a(t)$ selected by Formula (16) will be updated, and its action value will gradually increase. While the invalid action is shielded by the action mask method, its value function will not be updated, so the value of the invalid action will be significantly lower than other effective actions and will not be selected by the evaluation network.

3.3. Task Allocation Stage

In the objective Functions (9) and (10), the optimization goal of Formula (10) is to maximize the total importance of all tasks being performed. This is mainly realized through two aspects.

On the one hand, on the condition of maintaining the stability of the queue, we can maximize the number of tasks to be performed in the task queue and improve the utilization of idle participants. As the number of tasks being performed increases, the sum of task importance also increases. By selecting tasks with higher importance to enter the queue, the importance of performed tasks is further increased. This has been realized in Section 3.2.

On the other hand, we can maximize the importance by formulating a reasonable task allocation scheme $x_{si}(t)$. This is achieved in the task allocation stage.

In the task allocation stage, the platform allocates tasks in the task queue to participants to perform. Due to the limited number of participants, when the number of tasks is large, some tasks in the task queue cannot be allocated and performed immediately. The order of tasks to be performed is mainly determined by the importance of the tasks. Tasks with high importance will be assigned to execute first, while tasks with low importance need to wait for execution. The sum of the importance of all the tasks performed in Formula (10) is converted into the sum of the importance of the tasks performed in each time slot as following, which greatly reduces the difficulty and complexity of the problem, namely,

$$\max_{x_{si}(t)} I(t) = \max_{x_{si}(t)} \sum_{s=1}^{Q(t)} \sum_{i=1}^{N} F_s(t) x_{si}(t)$$

The above optimization problems can be solved by simple task importance ranking and selection methods.

It should be noted that, in fact, when training the action-value network of the DDQN, experience data need to be obtained after the above two stages. In state $S(t)$, the MCS system selects, assigns, and executes tasks and receives rewards $Re(t)$ and a new task queue $Q(t+1)$. When a new task $O(t+1)$ arrives in the next slot, the number of participants who can perform the task $N(t+1)$ is updated, and then the next state $S(t+1)$ is obtained. Therefore, the above two stages are a whole and are carried out simultaneously. Without the task allocation stage, the action-value network would not be able to obtain sufficient experience data and complete training.

Based on all of the above, the design of the TSAS for MCS is as follows (Algorithms 1 and 2).

Algorithm 1. TSAS based on Lyapunov optimization and DDQN

Initialize the evaluation network Q_ω with random network parameters ω.
Copy the same parameters $\omega' \leftarrow \omega$ to initialize the target network $Q_{\omega'}$.
Initialize *Replay buffer*.
Input: environment state $S(t)$: the number of tasks reaching the platform $O(t)$, the length of the task queue $Q(t)$, and the number of participants $N(t)$.
Output: the number of tasks entering the task queue $a(t)$ and task allocation scheme $x(t)$.
Set $Q(0) = 0$.
for episode $e = 1 \rightarrow E$ do:
Obtain environment state $S(t)$.
Input $s(t)$ into the evaluation network, output the action value $q_{eval}(S(t), a)$, and select action $a(t)$ with the ϵ-greedy policy.
Execute $a(t)$ in $S(t)$, and determine the tasks to enter the task queue according to $F_s(t)$.
Determine the task allocation and execution scheme according to $F_s(t)$: $x(t) = \{x_{si}(t)\}$.
After the tasks are completed, obtain the next state $S(t+1)$ and reward $Re(t)$.
Store the experience data tuple $\{S(t), a(t), Re(t), S(t+1)\}$ in the *Replay buffer*.
If *learn* = *True*:
Execute Algorithm 2: train the evaluation network, and update the parameters ω.
If *replace_target* = *True*:
Update target network parameters $Q_{\omega'}$: $\omega' \leftarrow \omega$.
end

Algorithm 2. Training and parameter update of action-value network

Sample N groups of data from the replay buffer: $\{(S(i),\ a(i), Re(i), S(i+1))\}_{i=1,\ \cdots,\ N}$
for $i = 1 \rightarrow N$ do:
Input $S(i)$ into the evaluation network to calculate the action value: $q_{eval}(S(i),\ a(i))$;
Input $S(i+1)$ into the target network to calculate the action value: $\hat{q}(i) = Re(i) + \gamma \cdot \max_{a'} q_{targ}(S(i+1), a')$;
Calculate the error: $\delta(i) = q_{eval}(S(i), a(i)) - \hat{q}(i)$.
end
Update the parameters ω of the evaluation network Q_ω to minimize the *Loss*: $Loss = \frac{1}{N}\sum_{i=1}^{N} \delta(i)^2$.
end

4. Experimental Validation and Performance Evaluation

Simulation experiments were conducted to evaluate the performance of the TSAS.

4.1. Training Performance

The MCS system was simulated, which consists of an MCS platform and participants. The number of participants $N(t)$ in each time slot follows a Poisson distribution of $\lambda = 5$. The number of tasks arriving at the platform $O(t)$ follows a Poisson distribution of $\lambda = 10$, $O(t)\epsilon[0, 20]$. The importance of arriving tasks $F(t)$ follows a Gaussian distribution with a mean of 20 and a variance of 10, $task_sig\epsilon(0, 40)$. Set $V = 1000$, $\beta = 1$.

In the experiments, an evaluation network and a target network were set up. The two neural networks have the same structure and are composed of an input layer, two hidden layers, and an output layer, which contain 3, 32, 16, and 21 nodes, respectively. Algorithm 1 is run to obtain experience data, which are stored in the replay buffer. The storage capacity of the replay buffer is 20,000. To enrich experience data, the ϵ-greedy strategy is adopted to select actions. $\epsilon = 0.1 + 0.0001 * Training\ times$, with a maximum of 0.9. The experiment runs 100,000 times. Each time the experiment runs, experience data are stored in the replay buffer. When the parameters of the evaluation network are updated 200 times, the target network updates its parameters once. Both the evaluation network and the target network

use the action mask method to reduce the feasible range of actions. The action mask is set as follows:

$$mask(a(S(t))) = mask(argmax_a(q(S(t),\ a))) = \begin{cases} q(S(t),\ a), & a \le O(t) \\ 0, & otherwise \end{cases} \tag{22}$$

$$a(t) = argmax_a(mask(a(S(t)))) \tag{23}$$

In order to observe the impact of iterative training methods, two training methods were adopted. Firstly, the network was trained 80,000 times using ordinary training methods without iterative training, and the training results are shown in Figure 6.

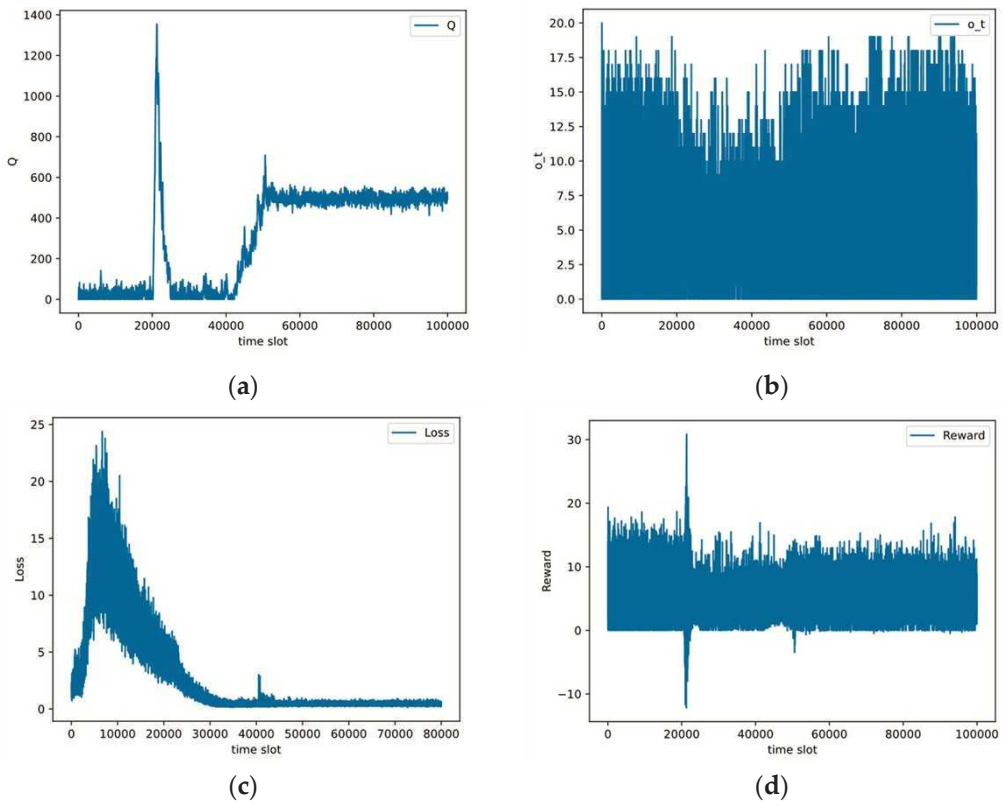


Figure 6. Training results of action-value network without iterative training: (**a**) task queue $Q(t)$, (**b**) the number of tasks that can enter the task queue $o(t)$, (**c**) *Loss*, (**d**) reward $Re(t)$.

Then, the iterative training method was used to continue training the network 100,000 times. The specific steps are as follows: First, the evaluation network runs to generate 10,000 experiences and stores them in the replay buffer; second, the evaluation network starts to be trained. Every 1000 runs of the evaluation network, the system state is initialized, so the states before queue stabilization are iteratively trained; third, after the evaluation network runs 50,000 times, ordinary training is resumed, and the states after the queue is stable are trained again until the system runs a total of 100,000 times. The training results are shown in Figure 7.

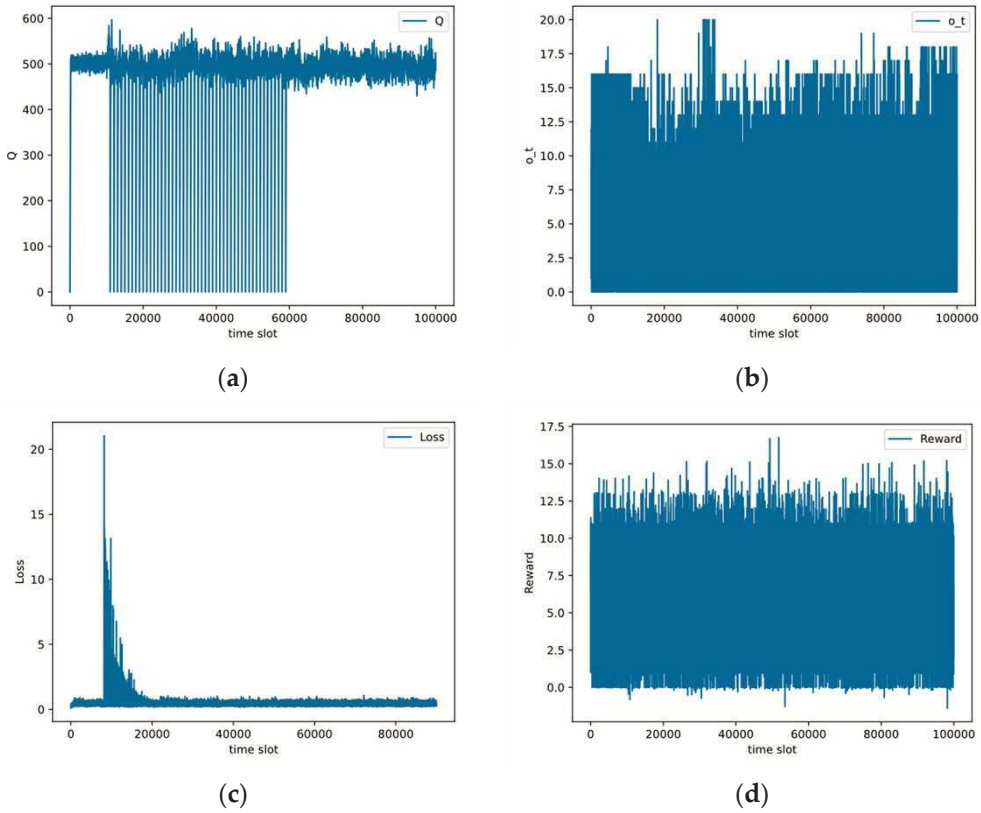


Figure 7. Training results of action-value network with iterative training: (**a**) task queue $Q(t)$, (**b**) the number of tasks that can enter the task queue $o(t)$, (**c**) *Loss*, (**d**) reward $Re(t)$.

In Figure 6, it can be seen that as the model continuously learns and optimizes, the task queue $Q(t)$ eventually tends to stabilize, and the loss decreases. This proves that the TSAS is feasible and can keep the task queue stable.

In Figure 6a, when $t \in [0, 41,000)$, $Q(t)$ fluctuates around smaller values; when $t \in [41,000, 50,000]$, $Q(t)$ significantly increases and tends to stabilize. Below is an analysis of the experimental results. When $t \in [0, 20000]$, the replay buffer is accumulating training data, and the evaluation network has not yet started training. The system adopts the ϵ-greedy strategy to select actions and randomly generates numbers in the interval $[0, 1)$. When the random number is less than ϵ, the evaluation network calculates the number of tasks that can enter the task queue $o(t)$. When the random number is greater than ϵ, $o(t)$ is generated in the interval $[0, O(t)]$ randomly. Therefore, within $t \in [0, 20,000]$, $Q(t)$ is small, and $o(t)$ fluctuates within $[0, O(t)]$. When $t \in [20,001, 41,000)$, the evaluation network starts training, and ϵ gradually increases from 0.1 to 0.9. $o(t)$ is more calculated by the untrained evaluation network, so $o(t)$ in Figure 6b decreases. Due to the storage of experience data generated by untrained networks in the replay buffer, the *Loss* is large, and $Q(t)$ is small. When $t \in [41,001, 100,000]$, as the evaluation network is trained and optimized, $o(t)$ continues to increase, and $Q(t)$ also increases and tends to stabilize, thereby maximizing the expected value of the reward.

In Figure 6c, it can be seen that the *Loss* gradually decreases with training. $Q(t)$ is relatively large and stable. In order to investigate whether states before queue stabilization, that is, states with a small value of $Q(t)$, have been trained, this paper continues to train the network using the iterative training method. The results are shown in the following figures.

In Figure 7a, it can be seen that the model trained 80,000 times can quickly make $Q(t)$ reach stability. When $t \in [10,000, 60,000]$, the state is initialized every 1000 times, so $Q(t)$ is reset to 0, and then $Q(t)$ quickly reaches stability again.

In Figure 7c, it can be seen that the *Loss* increases sharply and then gradually decreases with the training of the model. This is because, in classic training, as shown in Figure 6,

although the queue is stable and the loss gradually decreases and converges after training 80,000 times, the states before the queue is stable have not been fully trained. Therefore, when $Q(t)$ is small, the loss is still very large. After iterative training, regardless of whether the task queue is stable, that is, regardless of the size of $Q(t)$, the evaluation network can solve the objective function.

4.2. Comparison Experiment of Different Algorithms

In order to verify the performance of the TSAS, this paper compares the TSAS that has been trained according to Section 4.1 with three algorithms: the Classical Lyapunov optimization algorithm, the QPA algorithm [29], and the offline task selection and allocation algorithm.

Classical Lyapunov optimization algorithm: As described in Section 3.1, this algorithm is the mainstream and general solution for solving task allocation problems in MCS at present. In the drift-plus-penalty functions, there are unknown variables, which cannot be predicted before executing the tasks. Therefore, the Classical Lyapunov optimization algorithm solves the problem by eliminating unknown variables and minimizing one of the upper bounds of the drift-plus-penalty function through appropriate scaling methods.

QPA algorithm: This algorithm maximizes the importance of the tasks performed by controlling $o(t)$. In the QPA algorithm, all tasks can enter the platform and tasks that have not been executed will be stored in the task queue, waiting to be executed later.

Offline task selection and allocation algorithm: In order to compare the difference between the online task allocation method TSAS and the offline task allocation method, this paper also presents comparative experiments on the offline task selection and allocation algorithm. This algorithm selects and allocates tasks according to the system status of the current time slot and does not set the task queue.

The experiment runs for a total of 1000 time slots. $t\epsilon\{0, 1, \cdots, 999\}$. In reality, $O(t)$ varies at different time slots. Usually, there are more tasks that arrive during the day and fewer at night. Therefore, in order to simulate the changes in $O(t)$ more realistically, this experiment sets $O(t)$ in different time slots as follows: when $0 < t < 600$, $O(t)$ follows a Poisson distribution of $\lambda = 10$; when $600 < t < 1000$, $O(t)$ follows a Poisson distribution of $\lambda = 5$. $O(t)\epsilon[0, 20]$. The importance of arriving tasks $F(t)$ follows a Gaussian distribution with an average of 20 and a variance of 10. $F(t)\epsilon(0, 40)$.

Comparison experiments were conducted on different algorithms using the following evaluation metrics:

1. $Q(t)$ refers to the number of tasks waiting to be performed in the task queue.

$$Q(t+1) = max[Q(t) - r(t), 0] + o(t)$$

   The size of $Q(t)$ depends on the number of $o(t)$ and $r(t)$. Maintaining the stability of the task queue is an important goal of the online and long-term task allocation problem of the MCS system. By observing the change in $Q(t)$, we can intuitively judge whether the task queue has reached a stable state. The result of the comparison experiment on $Q(t)$ is shown in Figure 8.

2. $o(t)$ refers to the number of tasks entering the queue. $o(t)$ is an important part of the task selection and allocation method, which directly determines the queue length and its stability, system efficiency, and the importance of the tasks performed. When there are few participants, the system should reduce $o(t)$ in order to maintain the stability of the queue. When there are many participants, the system should increase $o(t)$ while maintaining queue stability in order to maximize the system utility and the performed task importance. Therefore, by observing $o(t)$, we can evaluate the advantages and disadvantages of different algorithms. The result of the comparison experiment on $o(t)$ is shown in Figure 9.

3. The *Return* represents the total rewards:

$$Return = \sum_{t=0}^{1000} Re(t)$$

The larger the Return, the smaller the drift-plus-penalty function, and the closer to the optimization goal. The result of the comparison experiment on the *Return* is shown in Figure 10a.

4. U_sum represents the total utility of the system. The more tasks that entered the queue, the greater the total utility.

$$U_sum = \sum_{t=0}^{1000} \beta \cdot o(t)$$

U_sum is not "the bigger, the better". Due to the limited number of participants and ability to execute tasks, when there are too many tasks in the queue, although U_sum is large, the queue congestion is serious, resulting in a large delay before the task is executed. The result of the comparison experiment on U_sum is shown in Figure 10b.

5. P_sum represents the total number of tasks actually executed:

$$P_sum = \sum_{t=0}^{1000} r(t)$$

When the number of participants and their ability to perform tasks are limited, P_sum is largely constrained by their ability to perform tasks. The result of the comparison experiment on P_sum is shown in Figure 10c.

6. I_sum indicates the total importance of the tasks performed.

$$I_sum = \sum_{t=0}^{1000} I(t) = \sum_{t=0}^{1000} \sum_{s=1}^{Q(t)} \sum_{i=1}^{N} F_s(t) x_{si}(t)$$

The larger I_sum is, the more important the tasks performed are. The result of the comparison experiment on I_sum is shown in Figure 10d.

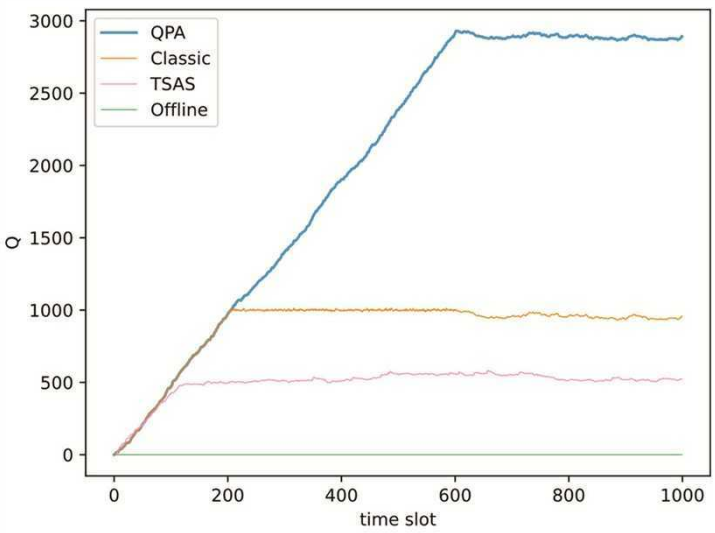


Figure 8. Variation in $Q(t)$ under different algorithms.

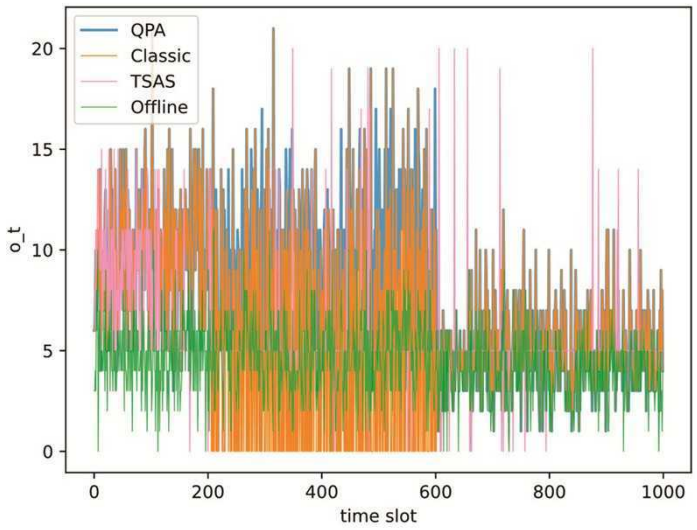
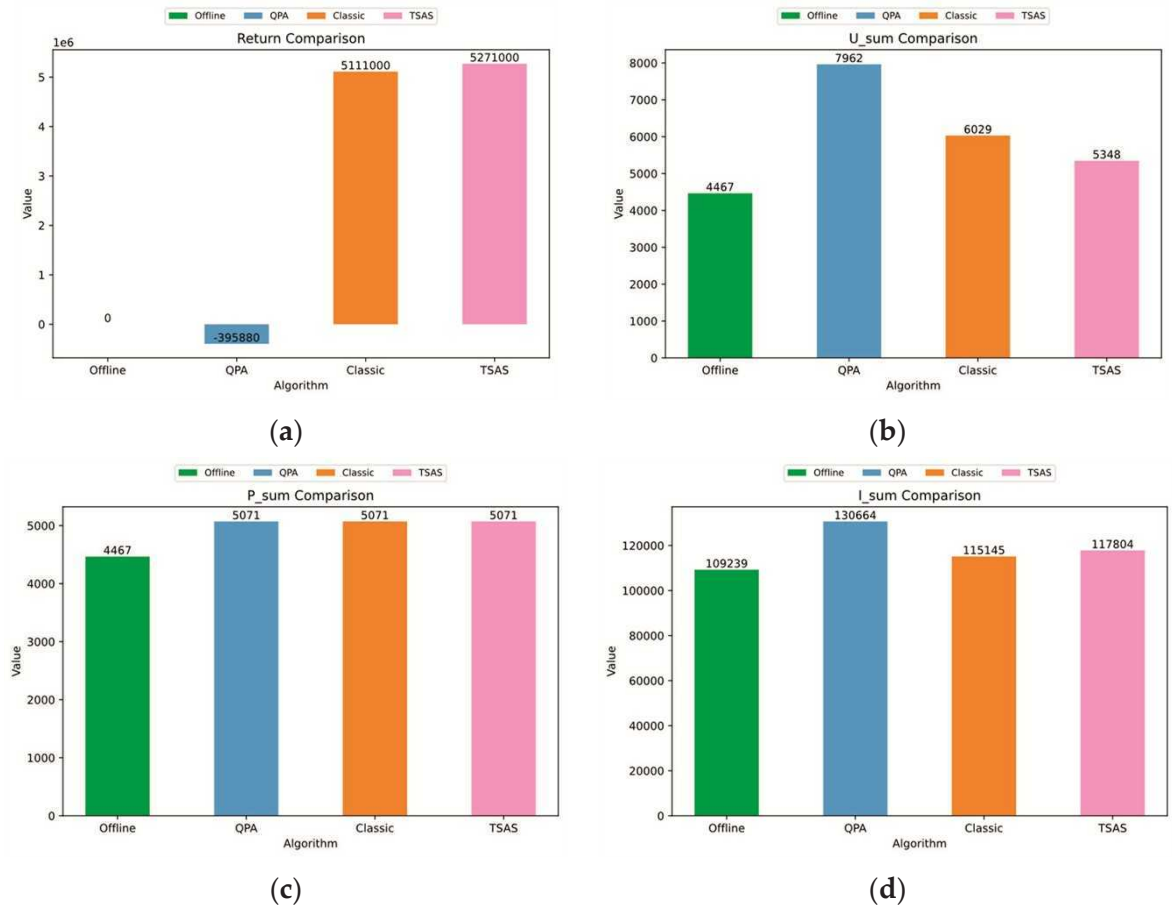


Figure 9. Variation in $o(t)$ under different algorithms.



(**a**) (**b**)

(**c**) (**d**)

Figure 10. Comparison of different indicators under different algorithms: (**a**) *Return* comparison. (**b**) *U_sum* comparison. (**c**) *P_sum* comparison. (**d**) *I_sum* comparison.

Figure 8 shows the variation in $Q(t)$ under different algorithms. The following can be seen: (1) The $Q(t)$ of the offline task selection and allocation algorithm is always 0. (2) When $0 < t < 600$, the queue of the Classical Lyapunov optimization algorithm and TSAS can gradually stabilize, while the queue of the QPA algorithm keeps growing and cannot reach a stable state. (3) When $600 < t < 1000$, the queue of the Classical Lyapunov optimization

algorithm and TSAS gradually decreases, while the queue of the QPA algorithm no longer grows and appears to remain stable.

The reasons for the above result are as follows:

1. The offline task selection and allocation algorithm makes decisions based on $O(t)$ and $N(t)$ in order to maximize the number of tasks to be executed. There is no backlog of queued tasks, so the queue length is always 0. It is precisely for this reason that some participants will be idle when the number of tasks decreases, but the number of participants increases, so the number of tasks executed by the offline task selection and allocation algorithm is lower than that executed by other online task selection and allocation algorithms, as shown in Figure 10.
2. When $0 < t < 600$, $O(t)$ follows a Poisson distribution of $\lambda = 10$, meaning that $O(t)$ is greater than $N(t)$, resulting in a backlog of queued tasks. The QPA algorithm seeks to maximize $U(t)$ greedily without limiting the queue length, so $Q(t)$ continues to grow and cannot stabilize. $Q(t)$ of the Classical Lyapunov optimization algorithm tends to stabilize at around 1000, while TSAS tends to stabilize at around 500. $Q(t)$ of the TSAS is significantly smaller than that of the Classical Lyapunov optimization algorithm. This is because the TSAS and the Classical Lyapunov optimization algorithm are both based on the Lyapunov optimization algorithm to maintain the stability of the task queue and pursue utility. The difference between the two algorithms lies in their solution. The Classical Lyapunov optimization algorithm approximates the unknown variable, minimizes an upper bound of the drift-plus-penalty function, and obtains an approximate optimal solution, thus causing high queue congestion. The TSAS can directly obtain the optimal solution of the drift-plus-penalty function, improving the defects of the Classical Lyapunov optimization algorithm. It can be seen that the TSAS can control the queue backlog and reduce the delay in task execution, which has important practical significance.
3. When $600 < t < 1000$, $O(t)$ follows a Poisson distribution of $\lambda = 5$. The QPA algorithm greedily pursues the maximization of $U(t)$; therefore, $Q(t)$ of QPA is equal to the average value of $N(t)$. At this point, the queue tends to stabilize. In fact, once $O(t)$ increases, $Q(t)$ will continue to grow, so the QPA algorithm cannot maintain the stability of the task queue. Due to the decrease in $O(t)$, the $o(t)$ of the Classical Lyapunov optimization algorithm and TSAS is reduced. The number of tasks executed is greater than the number of new tasks added to the queue, and the backlog of tasks in the queue is constantly being executed, so $Q(t)$ begins to decrease. Tasks that have not been executed are stored in the task queue instead of being directly discarded. Therefore, when participants have free time, the backlog of tasks in the queue will be executed. This increases the execution rate of tasks while also increasing the system's utility and improving the utilization of idle participants.

Figure 9 shows the selection decision $o(t)$ of the MCS system under different algorithms. (1) The $o(t)$ of the offline task selection and allocation algorithm is always consistent with the number of participants. (2) The $o(t)$ of the QPA algorithm is always consistent with $O(t)$. (3) When $0 < t < 600$, the TSAS and the Classical Lyapunov optimization algorithm both choose to enter the platform with some tasks instead of all tasks. However, the $o(t)$ of the Classical Lyapunov optimization algorithm fluctuates greatly, while the $o(t)$ of the TSAS has a small fluctuation interval. (4) When $600 < t < 1000$, the $o(t)$ of the TSAS and the Classical Lyapunov optimization algorithm is consistent with $O(t)$.

The reasons for the above results are as follows:

(1) The offline task selection and allocation algorithm does not consider the long-term utility of the system. As the $O(t)$ set in the experiment is greater than or equivalent to the number of participants, in order to maximize the number of tasks performed, $o(t)$ is consistent with the number of participants subject to the task execution capability of the system.

(2) The QPA algorithm pursues the maximization of system benefits $U(t)$ greedily without considering the stability of task queues. All tasks arriving at the platform enter the

task queue. Therefore, its $o(t)$ is consistent with $O(t)$. When $0 < t < 600$, $O(t)$ is large, and $o(t)$ is also large, which is greater than in other algorithms; when $600 < t < 1000$, it is consistent with the TSAS and the Classical Lyapunov optimization algorithm. $O(t)$ decreases, and $o(t)$ also decreases.

(3) When $0 < t < 600$, $O(t)$ is relatively large. In order to maintain the stability of the queue, the TSAS and the Classical Lyapunov optimization algorithm both choose to enter the platform with some tasks instead of all tasks. In addition, the $o(t)$ of the Classical Lyapunov optimization algorithm is obtained by solving the drift-plus-penalty function based on the classic solution, as mentioned before, which mainly depends on the current state $S(t)$, so $o(t)$ fluctuates greatly; the TSAS is based on the DDQN to solve Formula (14), so it not only depends on the current state $S(t)$ but also takes into account the future states in advance, so the fluctuation range of $o(t)$ is relatively small. In Figure 10, it can be seen that the fluctuation of $o(t)$ causes a difference in the total importance of tasks being performed, as detailed in the following analysis.

(4) When $600 < t < 1000$, $O(t)$ is equal to the number of participants. In order to maximize the system utility and the importance of performed tasks, the TSAS and the Classical Lyapunov optimization algorithm put all arriving tasks into the task queue, so their $o(t)$ is consistent with $O(t)$.

Due to the complexity of the actual situation, it is impossible to judge the merits of the algorithm by a single indicator. In the experiment, it is necessary to comprehensively consider the following four indicators to evaluate different algorithms.

In Figure 10, it can be seen that (1) *U_sum*, *P_sum*, and *I_sum* of the offline task selection and allocation algorithm are lower than those of the other three algorithms, and the *Return* is 0. (2) *U_sum* and *I_sum* of the QPA algorithm are the highest. *P_sum* is equal to the values of the Classical Lyapunov optimization algorithm and TSAS. The *Return* is negative, far less than that of the Classical Lyapunov optimization algorithm and TSAS. (3) The *Return* and *I_sum* of the TSAS are both higher than those of the Classical Lyapunov optimization algorithm, *P_sum* is equal to that of the Classical Lyapunov optimization algorithm, and only *U_sum* is slightly lower than that of the Classical Lyapunov optimization algorithm.

(1) Due to the offline task selection and allocation algorithm not considering long-term task allocation and having no backlog of queued tasks, when the number of tasks decreases and the number of participants increases, some participants will be idle. Therefore, its performance is lower than the other three long-term task allocation algorithms. The offline task selection and allocation algorithm makes decisions based on the current system state; *Return* has no meaning in this algorithm, so its *Return* was not calculated in this experiment.

(2) The QPA algorithm focuses on achieving optimal long-term utility, with all $O(t)$ entering the task queue for execution. Therefore, *U_sum* and *I_sum* are the highest. Due to the limitation of $N(t)$, *P_sum* is equal to the values of the Classical Lyapunov optimization algorithm and the TSAS. However, its *Return* is negative, far less than that of the Classical Lyapunov optimization algorithm and the TSAS, and the optimal solution cannot be obtained. This is because QPA cannot control the growth of the queue, and its task queue cannot reach stability. Therefore, it is not suitable for long-term task selection and allocation problems.

(3) Since *Return* is closely related to the objective Function (14), the larger the *Return*, the closer the solution is to the optimal solution. The TSAS can directly obtain the optimal solution of the objective function, so its *Return* is the largest. It can be seen in Figure 9 that, due to the fact that $o(t)$ of the TSAS tends to be more stable than that of the Classical Lyapunov optimization algorithm, the TSAS can select more important tasks to enter the queue. For the Classic Lyapunov optimization algorithm, in order to maintain stability of the queue, $o(t)$ is set to 0 in many time slots, so important tasks cannot be selected to enter the queue, resulting in *I_sum* being lower than that of the TSAS. Due to the

small number of participants, *P_sum* is mainly restricted by the number of participants. In order to maximize the system utility, the two algorithms enable all participants to perform tasks, so their *P_sum* values are the same. The *U_sum* of the Classic Lyapunov optimization algorithm is bigger than that of the TSAS, which leads to a large queue backlog for the Classic Lyapunov optimization algorithm. Thus, the TSAS is better than the Classic Lyapunov algorithm.

In summary, the TSAS based on the DDQN is more suitable for solving the drift-plus-penalty function in Lyapunov optimization. It dynamically adjusts the system's action to optimize performance as much as possible while maintaining system stability. Therefore, it can balance the system stability and utility in the drift-plus-penalty function.

4.3. Impact of V on TSAS

In order to study the impact of V on the TSAS, we set $V = 100$, $V = 500$, $V = 1000$, and $V = 2000$. Experiments were conducted on the TSAS. The experiments were run for a total of 1000 time slots. $t\epsilon\{0, 1, \cdots, 999\}$.

Figure 11a shows the variation in $Q(t)$ under different V. As V increases, the stable value of $Q(t)$ also increases. This is determined by the objective function (14). It can be seen that when $O(t)$ is determined, due to the fact that $o(t) \leq O(t)$, the larger the V, the longer the time required for $Q(t)$ to reach stability, and the larger the stable value of $Q(t)$. Figure 11b shows the changes in $o(t)$ with different V. It can be seen that the larger the V, the larger the $o(t)$, and its fluctuation range is also larger. The larger the V, the greater the weight of system utility. Therefore, It can be concluded that as V increases, the TSAS maximizes the *Return* by increasing $o(t)$, while the fluctuation range of $o(t)$ is also increased.

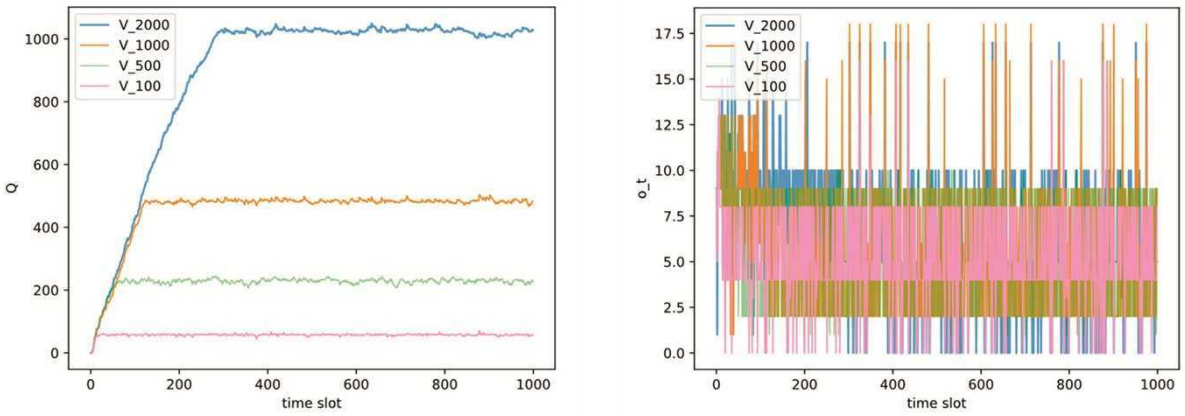
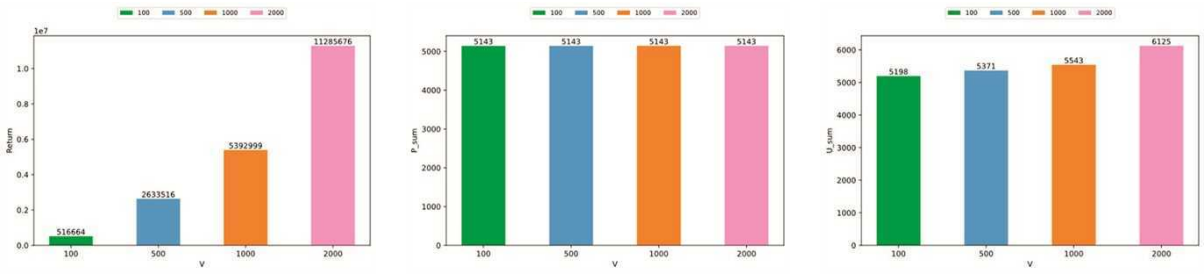


Figure 11. (**a**) Variation in $Q(t)$ with different V. (**b**) Variation in $o(t)$ with different V.

Figure 12 shows the variation in *Reward*, *U_sum*, and *P_sum* with different V. As V increases, *Return* and *U_sum* gradually increase, but *P_sum* is the same. This is determined by Formula (15). When V increases, it is obvious that the *Return* will also increase. V determines the size of the utility's weight in the drift-plus-penalty function, so when V is large, *U_sum* is high. Due to the fact that $O(t)$ is larger than $N(t)$, *P_sum* is limited to $N(t)$. When V increases, *P_sum* cannot increase accordingly.

From above, it can be seen that, in practical applications, V should be determined based on $O(t)$ and $N(t)$ in order to balance the queue length and system utility. Although a large V can increase system utility, it can also lead to an excessive queue backlog and the delayed execution of some tasks. When $N(t)$ is determined and V is high, that is, when the weight of system utility is high, the system utility is high. However, the stable value of $Q(t)$ also increases, resulting in the delayed execution of some low-importance tasks. When V is small, the weight of system utility decreases, and $U(t)$ decreases while the stable value of

$Q(t)$ decreases; thus, the delay in performing tasks is shortened. Therefore, comprehensive consideration should be given to the length of the queue backlog and system utility.



(**a**) (**b**) (**c**)

Figure 12. (**a**) Variation in *Return* with different V. (**b**) Variation in *P_sum* with different V. (**c**) Variation in *U_sum* with different V.

Further analysis reveals that there are two methods that can reduce the backlog of queued tasks. On the one hand, a backlog is caused by a large V and a large number of $o(t)$. Therefore, as mentioned above, V can be appropriately reduced to reduce the backlog. On the other hand, the ability of the system to perform tasks can be improved by recruiting more participants, thereby reducing the backlog.

5. Conclusions

This paper proposes the TSAS based on Lyapunov optimization and the DDQN, which divides the task assignment in the MCS system into a task selection stage and a task allocation stage. The TSAS is a long-term and online task allocation scheme that dynamically balances the execution ability of participants, the number of tasks, and the importance of the tasks, thus not only improving the completion of tasks but also improving the utilization of participants. The DDQN is used to improve on the traditional solution of the Lyapunov optimization problem so that the impact of future system states can be predicted. In addition, in order to accelerate the training process of the neural networks in the DDQN, this paper also proposes action-masking and iterative training methods, which are proved to be feasible.

In future research, there are still the following areas to be further explored. First, aiming at different application fields and different influencing factors, such as communication resources, participants' energy, task timeliness, etc., targeted research can be conducted on task selection and allocation in MCS. Second, in terms of the solutions of task selection and allocation problems, most of the current methods focus on obtaining analytical solutions or using heuristic algorithms to obtain approximate optimal solutions. Machine learning, deep reinforcement learning, or a combination of the two methods can be used to replace the traditional methods to solve some task allocation problems in MCS. In addition, although the focus of this paper is on centralized task allocation problems, distributed task allocation problems, which are rarely studied, are an important area for future exploration.

Author Contributions: Conceptualization, S.C., Y.W., S.D., W.M. and H.Z.; Methodology, S.C., Y.W., S.D. and H.Z.; Software, S.C.; Validation, S.C.; Formal analysis, S.C. and Y.W.; Investigation, S.C.; Resources, S.C.; Data curation, S.C.; Writing—original draft, S.C.; Writing—review & editing, S.C. and Y.W.; Visualization, S.C.; Supervision, Y.W., S.D. and W.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nguyen, T.N.; Zeadally, S. Mobile crowd-sensing applications: Data redundancies, challenges, and solutions. ACM Trans. *Internet Technol.* **2021**, *22*, 1–15. [CrossRef]
2. Minkman, E.; van Overloop, P.J.; van der Sanden, M.C. Citizen science in water quality monitoring: Mobile crowd sensing for water management in the Netherlands. In Proceedings of the World Environmental and Water Resources Congress, Austin, TX, USA, 17–21 May 2015.
3. Foschini, L.; Martuscelli, G.; Montanari, R.; Solimando, M. Edge-enabled mobile crowdsensing to support effective rewarding for data collection in pandemic events. *J. Grid Comput.* **2021**, *19*, 28. [CrossRef] [PubMed]
4. Zappatore, M.; Longo, A.; Bochicchio, M.A. Using mobile crowd sensing for noise monitoring in Smart Cities. In Proceedings of the International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, Croatia, 13–15 July 2016.
5. Habibzadeh, H.; Qin, Z.; Soyata, T.; Kantarci, B. Large-scale distributed dedicated- and non-dedicated Smart City Sensing Systems. *IEEE Sens. J.* **2017**, *17*, 7649–7658. [CrossRef]
6. Khan, F.; Ur Rehman, A.; Zheng, J.; Jan, M.A.; Alam, M. Mobile Crowdsensing: A survey on privacy-preservation, task management, assignment models, and Incentives Mechanisms. *Future Gener. Comput. Syst.* **2019**, *100*, 456–472. [CrossRef]
7. Zhang, J.; Zhang, Y.; Wu, H.; Li, W. An ordered submodularity-based budget-feasible mechanism for opportunistic mobile crowdsensing task allocation and pricing. *IEEE Trans. Mob. Comput.* **2024**, *23*, 1278–1294. [CrossRef]
8. Kang, Y.; Liu, C.; Zhang, H.; Han, Z.; Osher, S.; Poor, H.V. Task selection and route planning for mobile crowd sensing using multi-population mean-field game. In Proceedings of the IEEE International Conference on Communications (ICC), Electr Network, Virtual, 14–23 June 2021.
9. Stoeckel, B.; Kloker, S.; Weinhardt, C.; Dann, D. Quantity over quality ?—A framework for combining mobile crowd sensing and high quality sensing. In Proceedings of the 16th International Conference on Business and Information Systems Engineering (WI), Univ Duisburg Essen, Duisburg, Germany, 9–11 March 2021.
10. Ullah, N.; Khan, F.; Khan, A.A.; Khan, S.; Tareen, A.W.; Saeed, M.; Khan, A. Optimal real-time static and Dynamic Air Quality Monitoring System. *Indian J. Sci. Technol.* **2020**, *13*, 1–12. [CrossRef]
11. Huang, Y.; Chen, H.; Ma, G.; Lin, K.; Ni, Z.; Yan, N.; Wang, Z. OPAT: Optimized allocation of time-dependent tasks for mobile crowdsensing. *IEEE Trans. Ind. Inform.* **2022**, *18*, 2476–2485. [CrossRef]
12. Mourtzis, D.; Angelopoulos, J.; Panopoulos, N. UAVs for industrial applications: Identifying challenges and opportunities from the implementation point of View. *Procedia Manuf.* **2021**, *55*, 183–190. [CrossRef]
13. Wang, Y.; Dai, Z.; Zhang, W.; Zhang, S.; Xu, Y.; Chen, Q. Urgent task-aware Cloud Manufacturing Service composition using two-stage biogeography-based optimization. *Int. J. Comput. Integr. Manuf.* **2018**, *31*, 1034–1047. [CrossRef]
14. Qu, Z.; Ding, Z.; Moo, P. A machine learning task selection method for radar resource management (poster). In Proceedings of the 22th International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, 2–5 July 2019.
15. Ouyang, W.; Obaidat, M.S.; Liu, X.; Long, X.; Xu, W.; Liu, T. Importance-different charging scheduling based on matroid theory for wireless rechargeable sensor networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 3284–3294. [CrossRef]
16. Wang, L. Maintenance task scheduling of wind turbines based on Task Priority. In Proceedings of the 2020 Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling (APARM), Vancouver, BC, Canada, 20–23 August 2020.
17. Tchinda, Y.M.; Choquet-Geniet, A.; Largeteau-Skapin, G. Importance-Based Scheduling to Manage Multiple Core Defection in Real-Time Systems. Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2019.
18. Yang, L.; Xia, Y.; Ye, L. Heuristic scheduling method with the importance of earlier tasks for deadline constrained workflows in clouds. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021.
19. Shu, Z.; Zhang, K. A mechanism for network resource allocation and task offloading in mobile edge computing and network engineering. *Comput. Intell.* **2024**, *40*, e12628. [CrossRef]
20. Chen, G.; Zhang, J.; Ning, M.; Cui, W.; Ma, M. Task scheduling in real-time industrial scenarios. *Comput. Ind. Eng.* **2023**, *182*, 109372. [CrossRef]
21. Xiao, M.; An, B.; Wang, J.; Gao, G.; Zhang, S.; Wu, J. CMAB-based reverse auction for unknown worker recruitment in Mobile Crowdsensing. *IEEE Trans. Mob. Comput.* **2022**, *21*, 3502–3518. [CrossRef]
22. Gendy, M.E.; Al-Kabbany, A.; Badran, E.F. Maximizing clearance rate by penalizing redundant task assignment in Mobile Crowdsensing Auctions. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Electr Network, Virtual, 25–28 May 2020.
23. Neely, J. *Stochastic Network Optimization with Application to Communication and Queueing Systems*; Morgan and Claypool Publishers: San Rafael, CA, USA, 2010.
24. Lyu, L.; Shen, Y.; Zhang, S. The advance of reinforcement learning and deep reinforcement learning. In Proceedings of the IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 25–27 February 2022.
25. Saito, N.; Oda, T.; Hirata, A.; Nagai, Y.; Hirota, M.; Katayama, K.; Barolli, L. A Tabu list strategy based DQN for AAV Mobility in indoor single-path environment: Implementation and performance evaluation. *Internet Things* **2021**, *14*, 100394. [CrossRef]
26. Zhang, X.; Shi, X.; Zhang, Z.; Wang, Z.; Zhang, L. A DDQN path planning algorithm based on experience classification and multi steps for Mobile Robots. *Electronics* **2022**, *11*, 2120. [CrossRef]
27. Chen, L.; Wang, Q.; Deng, C.; Xie, B.; Tuo, X.; Jiang, G. Improved double deep Q-network algorithm applied to multi-dimensional environment path planning of Hexapod Robots. *Sensors* **2024**, *24*, 2061. [CrossRef] [PubMed]

28. Huang, S.; Ontañón, S. A closer look at invalid action masking in policy gradient algorithms. In Proceedings of the International FLAIRS Conference Proceedings, Jensen Beach, FL, USA, 15–18 May 2022.
29. Gong, W.; Zhang, B.; Li, C. Location-based online task assignment and path planning for mobile crowdsensing. *IEEE Trans. Veh. Technol.* **2018**, *68*, 1772–1783. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Detection of Scratch Defects on Metal Surfaces Based on MSDD-UNet

Yan Liu †, Yunbai Qin *, Zhonglan Lin †, Haiying Xia * and Cong Wang

School of Electronic and Information Engineering, Guangxi Normal University, Guilin 541004, China; lyyy_ran@stu.gxnu.edu.cn (Y.L.); lin_zhonglan@stu.gxnu.edu.cn (Z.L.); cong@stu.gxnu.edu.cn (C.W.)

* Correspondence: qinyunbai@gxnu.edu.cn (Y.Q.); xhy22@mailbox.gxnu.edu.cn (H.X.)

† These authors contributed equally to this work.

Abstract: In this work, we enhanced the U-shaped network and proposed a method for detecting scratches on metal surfaces based on the Metal Surface Defect Detection U-Net (MSDD-UNet). Initially, we integrated a downsampling approach using a Space-To-Depth module and a lightweight channel attention module to address the loss of contextual information in feature maps that results from multiple convolution and pooling operations. Building on this, we developed an improved attention module that utilizes image frequency decomposition and cross-channel self-attention mechanisms, as well as the strengths of convolutional encoders and self-attention blocks. Additionally, this attention module was integrated into the skip connections between the encoder and decoder. The purpose was to capture dense contextual information, highlight small and fine target areas, and assist in localizing micro and fine scratch defects. In response to the severe foreground–background class imbalance in scratch images, a hybrid loss function combining focal loss and D_{ice} loss was put forward to train the model for precise scratch segmentation. Finally, experiments were conducted on two surface defect datasets. The results reveal that our proposed method is more advantageous than other state-of-the-art scratch segmentation methods.

Keywords: defect detection; attention mechanism; hybrid loss; U-Net; SPD module; semantic segmentation

1. Introduction

Metals and metal products are indispensable in modern industrial manufacturing. However, defects such as scratches are often unavoidable on their surfaces due to limitations in production processes, tool and equipment wear, careless operation, or collisions and compression during transportation. These scratch defects negatively affect the aesthetic appeal of such products, making them difficult to clean and significantly reducing important properties such as their dimensional stability, surface roughness, and fatigue limits. Therefore, the timely repair of scratches is crucial to ensure the performance of downstream products. Manufacturers can take a series of measures, such as recoating or applying surface treatment techniques, to enhance the aesthetics and durability of metals and metal products. However, before proceeding with a repair, a crucial and indispensable step is detecting the quality of the metal.

Accurate metal defect detection is crucial for modern industrial production. The detection results can be used for the precise quality assessment of products, raw materials, etc., and serve as a basis for making maintenance decisions. In recent years, deep learning methods have significantly improved the performance of the detection of metal surface scratch defects. To achieve high-precision defect detection, large and complex deep learning models are typically designed. However, these models require substantial computational resources and have lengthy training and inference times, making it challenging to achieve the high defect detection efficiency required in industrial production. Moreover, the accurate detection of scratch defects remains challenging due to the inherent class imbalance

between the foreground and background in metal surface scratch defect data, as well as complex factors such as small or elongated targets. To address these issues, this paper proposes a downsampling mode based on a Space-To-Depth (SPD) [1] module and a lightweight channel attention module (LCAM) and enhances the self-attention mechanism used. Based on this, an algorithm is proposed for localizing metal scratch defects at the pixel level for defect segmentation (refer to Figure 1). The main component of the proposed model is an enhanced U-shaped network (U-Net) designed to accurately segment scratches of different sizes from a metal surface image. The primary contributions of this paper are as follows:

(1) A downsampling convolutional block is proposed based on an SPD module and LCAM. It is embedded into the U-Net to acquire more information and mitigate the loss of contextual information in feature maps after multiple convolution and pooling operations in the encoder.

(2) Building upon the above contribution, a lightweight anti-redundant self-attention module is designed to address the geometric features of scratches, such as their direction, thickness, and inherent high-frequency components, by integrating refined edge information. This module is embedded into the skip connections between the encoder and decoder, serving as a precursor for encoding–decoding feature fusion.

(3) Furthermore, a hybrid loss function that combines the focal loss [2] and D_{ice} loss [3] functions is introduced to deal with the inherent foreground–background class imbalance in metal surface scratch defect data. By leveraging their respective unique advantages, the proposed function is utilized to train a model for precise scratch segmentation.

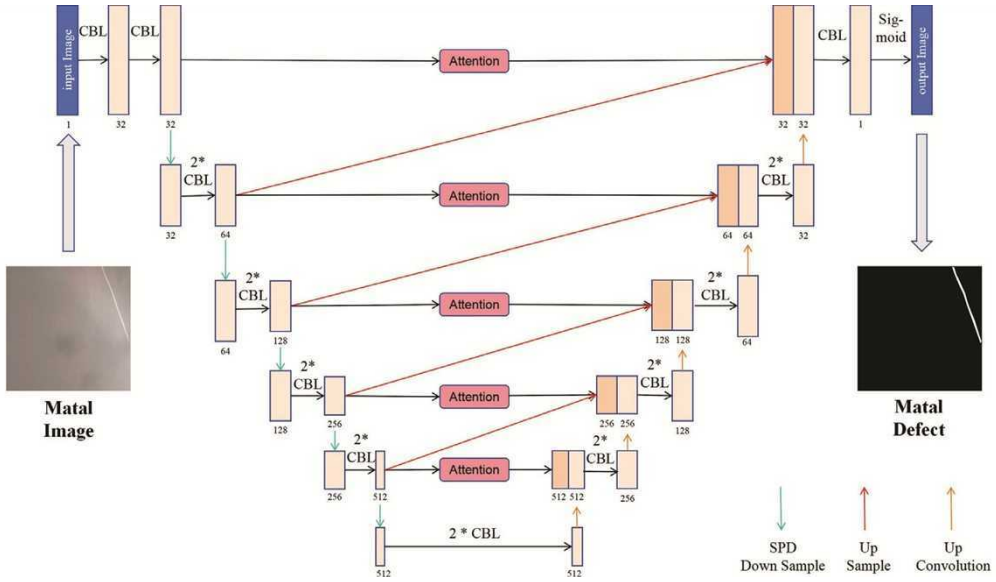


Figure 1. Architecture of the proposed segmentation network. We incorporate an enhanced self-attention block into the feature fusion of these encoding–decoding stages, which acts as the preliminary process of feature fusion. In the encoding process, we utilized an efficient downsampling method based on SPD and LCAM, replacing the traditional max pooling operation, as illustrated by the light green arrow in the figure. Additionally, the red arrow in the figure indicates the upsampling operation, which uses bilinear interpolation, while the orange arrow indicates the Up-Convolution operation.

2. Relevant Works

2.1. Traditional Methods

Early methods of metal defect detection include liquid penetrant testing [4], laser scanning detection, eddy current testing, ultrasonic testing, and magnetic particle inspection. Manual defect detection was once a common method, but it is labour-intensive. With the rapid advancement of the modern manufacturing industry, this manual method is no longer sufficient to meet the demand for efficient and high-precision defect detection in

contemporary industrial production. This is due to its vulnerability to errors and omissions, low efficiency, and excessive reliance on work experience.

2.2. Digital Signal Processing Methods

With the advancement of automation technology, many researchers and technologists have made significant contributions to defect detection which primarily involve traditional digital signal processing methods and deep learning methods.

Regarding over-detection on defect-free surfaces, H. Ono et al. [5] reported a twin illumination and subtraction technique. They developed a prototype system that can detect defects by capturing two images with a slight time difference. Their system is based on strobe lighting and a dual CCD camera. It utilizes a decision tree judgment function that analyses the light/dark patterns of concave defect features. Liu et al. [6] introduced a hardware and software system for online visual inspection. The hardware comprises an upper optical module and a lower optical module. The defect detection algorithm includes an automatic search of the regions of interest (ROI) and defect detection based on the Sobel algorithm. For large steel roll surface defects with fuzzy boundaries, uneven intensity, and complex background textures, Yang et al. [7] proposed a machine vision system based on an edgeless active contour model. This system can accurately segment blurry and non-uniform defects and typically requires less computation time.

Compared to early purely manual detection methods, the above-mentioned approaches utilize traditional image processing and machine learning algorithms for digital signal processing, offering advantages such as objective results and high computational efficiency. However, these approaches rely on specific signal acquisition tools and demonstrate high detection precision only for specific defects, lacking strong generalization ability. Additionally, these approaches require manual feature design, which requires technicians to have specialized prior knowledge of the detection object's properties, such as the metal's acoustic conductivity, as well as its optical refractivity and reflectivity.

2.3. Deep Learning Methods

Compared to digital signal processing methods, deep learning methods, such as convolutional neural networks (CNNs), can automatically extract deep representative features end-to-end from data [8]. In the past decade, deep learning and neural networks have been widely applied in the field of computer vision [9–12] due to their powerful feature extraction capabilities and ability to generalize [13]. This progress has been greatly supported by high-performance graphics cards, cloud computing power, and big data. These advancements have enabled real-time and efficient defect detection in modern industrial production.

In the field of image segmentation, Fully Convolutional Networks (FCNs) [10] and U-Net [11] have provided models for extracting features from segmentation targets more precisely and efficiently. Specifically, U-Net utilizes an encoder–decoder structure with skip connections, enabling the merging and blending of feature maps across different network levels. This process helps the decoder reconstruct feature maps to their original image size, significantly enhancing the ability of a fully convolutional model to preserve details and segment accurately.

Over the past few years, researchers in related fields have made significant progress in utilizing deep learning models to address various challenges encountered in defect segmentation and anomaly detection [14–24]. Ref. [14] proposed a composite vision system to enable simultaneous the 3D depth and 2D grey imaging of the bead surface. This system is designed to detect typical surface defects on aluminium alloy weld beads. Ashkan Moosavian et al. [15] collected a dataset comprising 400 experimental images of crankshafts with structural defects, such as scratches, pitting, and grinding. They simultaneously applied the DexiNed edge detection filter to the training set images and trained a model based on MobileNet to detect these defects accurately. In the detection of scratch defects on curved metal surfaces, inevitable halos often obscure the defects' features. To address

this issue, He et al. [16] used a multiple high-frequency projection illumination imaging method to generate high-contrast images of metal part surfaces. They utilized this method to create a dataset named RMSSC, which includes 50 images of reflective metal surface scratches. Subsequently, they developed a defect detection neural network and enhanced its generalization performance through inverse transfer learning. In response to the need for accurate manual data annotation in supervised training, Ref. [17] proposed a semi-supervised method. This method combines the predictions of an object detector with the segmentation of a zero-shot model, eliminating the need to label a dataset for semantic segmentation.

Currently, deep learning methods have achieved good performances in surface defect detection. The aforementioned works have undoubtedly made significant contributions to defect detection methods based on deep learning. However, efficient and high-precision defect detection still faces certain challenges due to the varying sizes of defects in images, complex production environments affecting data sampling, and the inherent foreground–background class imbalance in metal surface defect data.

3. The Proposed Segmentation Network

In this study, we developed an image segmentation network to localize scratch defects. The backbone network, attention module, and loss function of the proposed network are described in the following section.

3.1. Outline of the Model

Inspired by the U-Net segmentation network, we established a scratch defect segmentation system for metal surface data (Figure 1). This system consists of five components: a backbone network based on encoder–decoder architecture, a downsampling module, an attention module, skip connections, and feature fusion.

The backbone network is a modified version of the basic U-Net model. Initially, a new downsampling approach was proposed, based on LCAM and SPD modules, to capture more information. Subsequently, an improved self-attention module was developed based on image frequency decomposition, leading to the design of a lightweight attention module. This lightweight attention module has been integrated into the skip connections between the encoder and decoder to capture dense global contextual information, emphasize defect areas, and assign weights to each channel. The specific modules, methods, and the loss function will be detailed below.

3.2. Basic Framework

U-Net performs well in most image segmentation applications. Therefore, we improved the U-Net model and utilized it as the fundamental backbone network architecture for our proposed method.

The U-Net segmentation network is composed of an encoder and a decoder. In its enhanced basic architecture, each block in the encoder comprises two convolutional layers, batch normalization modules (Batch Normalize), leaky rectified linear units (Leaky ReLUs), and a downsampling module that reduces the feature maps to 1/4 of their original scale. The decoder part upsamples the feature maps using a deconvolutional layer and two convolutional layers, followed by batch normalization modules and Leaky ReLU functions. To reduce the information loss caused by pooling operations, the blocks corresponding to the encoder and decoder paths are used to splice the feature maps together via skip connections. This integration combines coarse semantic information from each section of the encoder with fine semantic information from each section of the decoder.

Thus, the shallow layers of the encoder focus on low-level features such as image contours, textures, and colours, while the deep layers of the encoder, with a large receptive field, learn high-level features. Moreover, skip connections help to recover lost edge information as much as possible, enabling the U-Net to retain high-level semantic information from the encoder while maintaining detailed information from the decoder.

3.3. Improved Self-Attention Module

To enhance the accuracy and robustness of the segmentation results, the DeepLabV2 model [13] included an Atrous Spatial Pyramid Pooling (ASPP) module with multi-scale atrous convolutions. However, this approach falls short in capturing dense contextual information, as it only collects contextual details from a limited number of points surrounding each pixel through atrous convolutions. To address this limitation and establish connections between pairs of pixels, Alexey Dosovitskiy et al. [12] incorporated a self-attention mechanism into computer vision. Traditional semantic segmentation models face challenges in fully utilizing the global information of an image due to the local receptive field constraint of convolutional encoders. To address this issue, Danpei Zhao et al. [25] introduced the RSANet, a regional self-attention-based network. This network enhances the model's logical understanding of image content by examining point-to-point relationships between pixels and surrounding regional pixels, thereby improving segmentation accuracy. However, in a convolutional neural network, neighbouring elements in the feature maps of convolutional layers share semantic information spatially. Golnaz Ghiasi et al. [26] pointed out that even if neurons are killed, convolutional neural networks can still obtain the semantic information of a pixel point according to its neighbouring pixels. Similarly, in traditional self-attention modules, the pairwise dependencies between pixels are typically computed repeatedly. Therefore, the self-attention module must be lightweight and anti-redundant because the global feature maps obtained by traditional self-attention modules often contain redundant information.

The wavelet transform is a powerful extension of the Fourier transform, renowned for its ability to capture the instantaneous characteristics of a signal by utilizing local information in both time and frequency domains. In the field of image processing, especially for tasks with local mutable objectives, the wavelet transform is commonly employed to decompose the input image into low-frequency components and high-frequency components in vertical, horizontal, and diagonal directions. The high-frequency components contain local detailed information, especially edge information. On the other hand, the low-frequency components represent the global information of the image layer, such as colour. Consequently, high-frequency components often require local and detailed processing, while low-frequency components require global structural processing.

$$Attention(X) = Vsoftmax(Q^T K) + X \tag{1}$$

$$Q = W^Q X \tag{2}$$

$$K = W^K X \tag{3}$$

$$V = W^V X \tag{4}$$

As depicted in Figure 2, the feature map generated by the convolutional encoder was a combination of high- and low-frequency features. To prepare for the merging of encoding–decoding features, the encoder's extracted feature map was implicitly divided into low-frequency and high-frequency components by introducing a pair of convolutions with a kernel size of $(1, 1)$. Subsequently, the low-frequency components were fed into the self-attention module to extract their overall features. These overall features were then combined with the high-frequency components into a decoder feature map. This methodology draws inspiration from octave convolution [27] and CSP [28] modules. In the self-attention module, calculations were conducted across channel dimensions, as shown in Equation (1), instead of across spatial dimensions. This adjustment aimed to reduce the computational burden of the self-attention mechanism and implicitly encode global information. Consequently, this approach prevented the self-attention block from generating redundant information while establishing relationships among the pixels of low-frequency

components, resulting in the extraction of efficient global features and the creation of dense contextual information. Simultaneously, the high-frequency components highlighted the target areas, aiding in the localization of scratch defect areas and improving the module's ability to segment minor and fine defects.

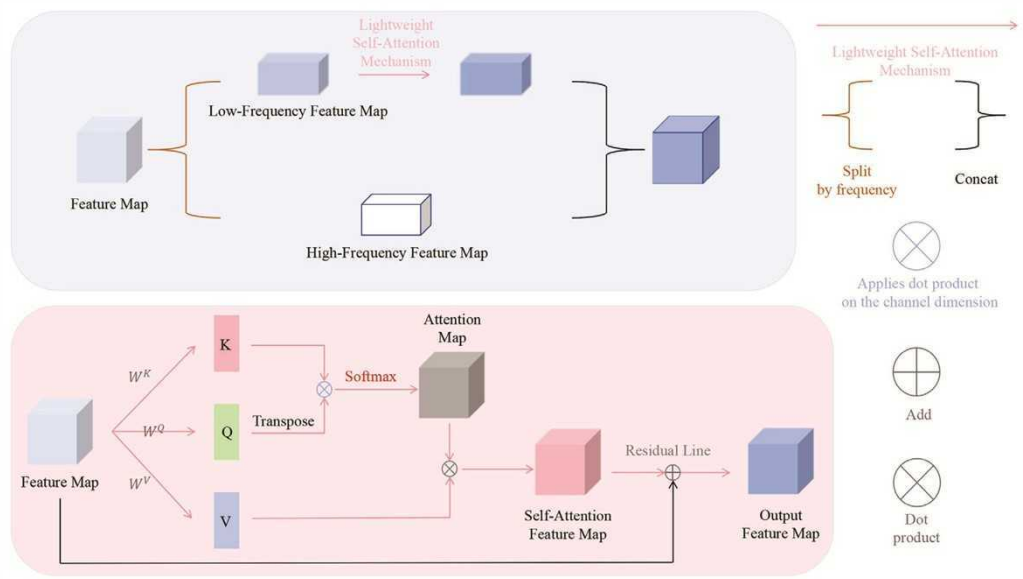


Figure 2. Network structure of the attention module.

3.4. Efficient Downsampling Module

Relying solely on skip connections for feature fusion between the encoder and decoder is far from sufficient for recovering lost features, especially in the initial stage of network training. Before training and evaluating AlexNet, VGGNet16, and ResNet-50, Koziarski et al. [29] reduced the size of the input image in the luminance channel and then filled it again using the bilateral interpolation method. They found that when the resolution of the images used for ResNet-50 training was reduced to 1/5 and 1/8 of its original resolution, the classification accuracy of ResNet-50 dropped to 75% and 50% of its original accuracy, respectively.

To address this issue, Chen et al. [30] proposed a global context-aware progressive aggregation network based on the U-Net framework. This network consists of four modules: Feature Interweaving Aggregation (FIA), Self-Refinement (SR), Head Attention (HA), and Global Context Flow (GCF). These modules interweave and fuse the global contextual information, low-level detail information, and high-level semantic information of the U-Net, obtaining richer feature information. Zhao et al. [31] proposed an edge-guided network (EGNet) for salient object detection, which progressively integrates and complements local edge information and global position information to extract prominent edge features. However, we believe that these approaches do not tackle the core issue; instead, the focus should be on alleviating the loss of detailed information at the source.

In this paper, a downsampling convolution block based on an SPD module and a lightweight channel attention module was proposed and embedded into the encoder to reduce the contextual information loss caused by the multiple convolutions and pooling operations conducted in the encoder. This approach prevented the loss of valuable information at its source, thereby capturing more information.

3.4.1. Space-to-Depth Module

A feature map X with dimensions (S, S, C_1) is sliced as follows, and as displayed in Figure 3:

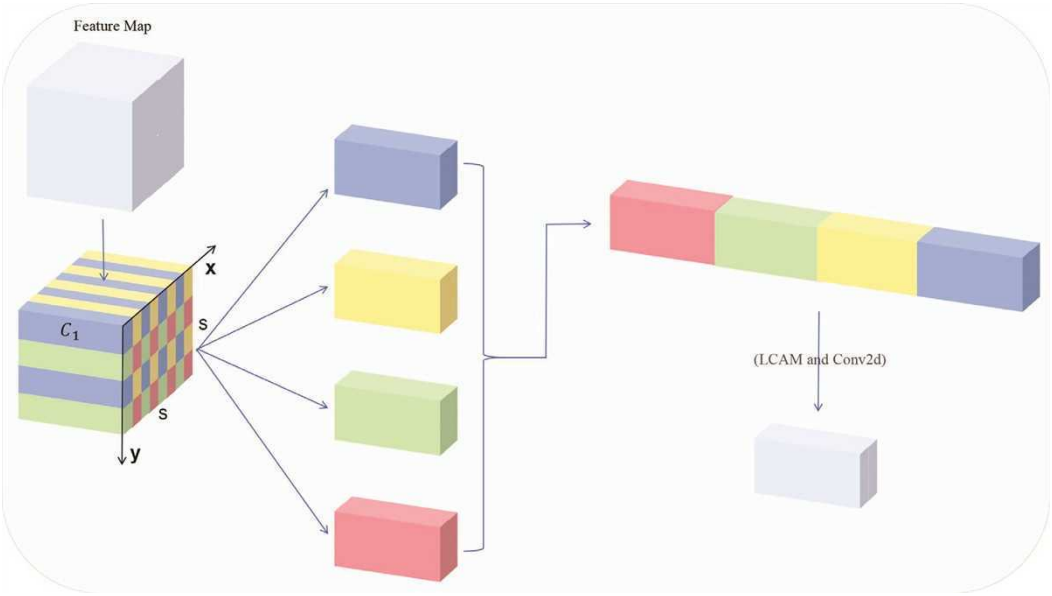


Figure 3. Framework of the downsampling module.

$$
\begin{aligned}
&f_{0,0} = X[0:S:scale, 0:S:scale],\\
&f_{1,0} = X[1:S:scale, 0:S:scale],\\
&f_{scale-1,0} = X[scale-1:S:scale, 0:S:scale];\\
&f_{0,1} = X[1:S:scale, 0:S:scale],\\
&f_{1,1} = ...,\\
&f_{scale-1,1} = X[scale-1:S:scale, 1:S:scale];\\
&f_{0,scale-1} = X[0:S:scale, scale-1:S:scale],\\
&f_{1,scale-1} = ...,\\
&f_{scale-1,scale-1} = X[scale-1:S:scale, scale-1:S:scale];
\end{aligned}
$$

3.4.2. Lightweight Channel Attention Module

On feature maps, each channel corresponds to the extraction result of a specialized feature extractor. However, not all channels have features of the same importance. Each channel needs to be assigned a weight. Generally, for an input, its channel weight is determined based on Equation (5), where MLP represents a multi-layer perceptron with shared weights, and GAP and GMP represent the global average pooling and global max pooling, respectively.

To downsample efficiently, the channel attention module must be lightweight. Therefore, an MLP was designed with a single hidden layer containing only one neuron, and pointwise convolution layers were used instead of fully connected layers. The structure of this module is illustrated in Figure 4.

$$W = \sigma\{MLP[GAP(x)] \oplus MLP[GMP(x)]\} \tag{5}$$

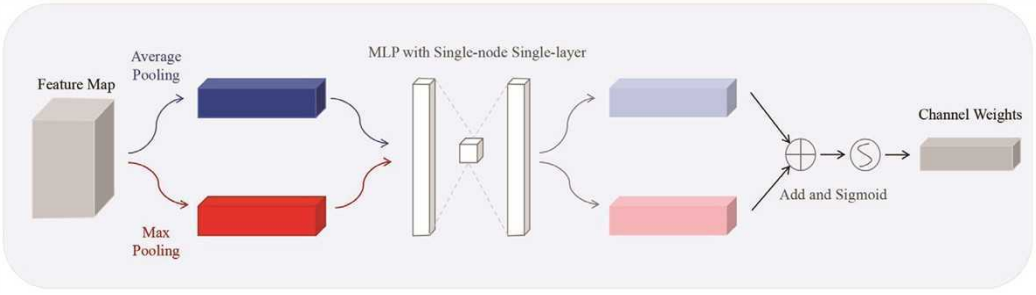


Figure 4. Framework of the LCAM.

3.4.3. Our Downsampling Module

Based on the two preceding modules, we have designed an efficient downsampling module, as detailed in Figure 3. For the input feature map, first, its size is reduced to 1/4 of its original while quadrupling the depth (i.e., the number of channels) using the SPD module, as detailed in the previous section. Subsequently, the LCAM assigns weights to each channel of the feature map, as illustrated in Figure 4. Finally, a convolution block with a kernel size of $(1, 1)$ is utilized to restore the depth of the feature map to its original size.

As shown in Figure 1, we have integrated our downsampling module into our model. Specifically, it is designed to efficiently downsample the feature map during the encoding process, thereby replacing the max pooling layer used in the original U-Net model. Notably, in Figure 1, we have represented the downsampling process with the light green arrows.

3.5. Hybrid Loss Function

In deep learning segmentation tasks, the loss function is used to measure the discrepancy between the predicted segmentation result and the ground truth mask. Its purpose is to guide the model to effectively learn from the data during training, enabling the model to predict segmentation results that closely resemble the true masks. To achieve efficient and accurate defect segmentation, an effective loss function is crucial. Binary Cross-Entropy (BCE) is widely used as an efficient loss function in computer vision tasks such as image classification [32]. Its expression is as follows:

$$L_{BCE} = -\sum -(1 - t_i)log(1 - \hat{p}_i) + t_i log\hat{p}_i \tag{6}$$

where p_i and t_i represent the predicted value and the true value, respectively.

In contrast to most segmentation tasks, a scratch defect pixel area accounts for a very low proportion of a full image's pixels. The segmentation of scratch defects on a metal surface presents a typical foreground–background class imbalance challenge. The loss for the entire sample is expressed as follows:

$$Loss = \frac{\sum_{i=1}^{N} L(y_i, \hat{p}_i)}{N} \tag{7}$$

For the binary classification of metal scratches, the loss expression is presented as follows:

$$Loss = \frac{\sum_{i=1}^{m} -log(\hat{p}) + \sum_{i=0}^{n} -log(1 - \hat{p})}{N} \tag{8}$$

where $N = m + n$ represents the total number of samples, with m and n denoting the numbers of positive and negative samples, respectively. When $m \ll n$ (namely, when negative samples dominate), this indicates that the model tends to learn from the class with more samples, severely impacting its ability to recognize the class with fewer samples.

As such, the BCE is severely affected by class imbalance issues. Therefore, we adopted the focal loss function, as displayed in Equation (9), to train the model for improved classification performance.

$$L_{focal} = -(1 - p_t)^{\gamma} log(p_t) \tag{9}$$

where $p_t = \begin{cases} \hat{p} & , if\ y = 1 \\ 1 - \hat{p} & , otherwise \end{cases}$

In Equation (9), when $\gamma > 0$, samples difficult to identify are weighted and treated as hyperparameters. When γ is set to 0, L_{focal} is degraded into the original CE (Cross-Entropy) loss function. $(1 - p_t)^{\gamma}$ can reduce the weights of samples that are easy to identify and increase the weights of samples that are difficult to identify, thereby addressing class imbalance issues [2].

For a segmentation model, it is necessary to evaluate the similarity between the areas segmented by the model and the true scratch areas, in addition to measuring the classification effect of the model using a loss function. Hence, we introduced the D_{ice} loss function to assess the similarity between the areas segmented by the model and the ground truth masks [3], thus supervising the segmentation network's learning effect. The D_{ice} loss function is defined as follows:

$$L_{D_{ice}} = \frac{2\sum_{i=1}^{N} p_i t_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} t_i^2} \tag{10}$$

where N is the total number of pixels, p_i is the value of the i-th pixel in the result segmented by the model, and t_i is the value of the i-th pixel in the ground truth scratch mask.

To train the model to segment metal scratches more efficiently, we introduced a hybrid loss function that combines the characteristics of both the focal and D_{ice} loss functions. By weighting and summing these two losses, their respective strengths were effectively leveraged. The hybrid loss function was finalized and expressed as follows:

$$Total\ Loss = \alpha L_{focal} + L_{D_{ice}} \tag{11}$$

where α serves as a hyperparameter and is used to balance the focal loss and D_{ice} loss. As such, the hybrid loss function possesses the sensitivity of BCE towards pixel misclassification, the sensitivity of D_{ice} loss towards the discrepancy between segmented results and true scratch areas, and the feature of focal loss to weight samples that are difficult to classify.

4. Experiment and Analysis

4.1. Experimental Environment

The proposed network was implemented using PyTorch 1.13.1 and the OpenCV 4.8.1.78 extension library. The GPU utilized was the NVIDIA GeForce RTX 3050 Laptop, and the CPU was the AMD Ryzen 7 5800H with Radeon Graphics. The initial settings were as follows: a learning rate of 0.01, a decay of 0.001, a momentum of 0.88, a batch size of 32, and a maximum of 150 iterations. Additionally, the weight α for the focal function in the hybrid loss function was set to 0.5, and the γ factor in the focal loss function was set to 2.0.

4.2. Dataset

Our work aims to enhance the segmentation capabilities of the model with respect to fine and micro targets. In this study, MSDD-UNet is employed to detect scratches on metal surfaces, addressing issues such as the inherent foreground–background imbalance in the scratch segmentation dataset. To this end, we specifically created a dataset for segmenting fine and micro scratches. In total, we collected 1575 metal surface images and extracted the sections containing scratch defects (see Figure 5 for a sample of these images).

As shown in Figure 5, the dataset comprises images of scratches on various metal surfaces captured under different lighting conditions. The scratches exhibit a range of shapes and colours and are predominantly elongated, with some shorter ones as well. Recognizing and segmenting scratches in such a dataset proves challenging for the model, as it faces several difficulties:

(1) The scratches vary in colour and shape, and the surface features of different metals are quite different.

(2) The images contain elongated scratches, which makes it difficult for the model to accurately segment the tips of these targets.

(3) The dataset includes various metals with diverse surface textures. These textures may lead the model to misidentify them as scratches, causing disturbances.

(4) The scratch segmentation dataset inherently suffers from an imbalance between foreground and background, with scratch pixels constituting a relatively small proportion of the overall image.

After preprocessing, the images were uniformly resized to $96 \times 96 \times 3$. We used LabelMe to annotate scratch defects and generate corresponding mask labels. The dataset was split into a training set and a test set at an 8:2 ratio, with the training set comprising 1260 images and the test set comprising 315 images. Before initiating training, we randomly subjected the training set to horizontal, vertical, and combined horizontal and vertical flips to augment the data.

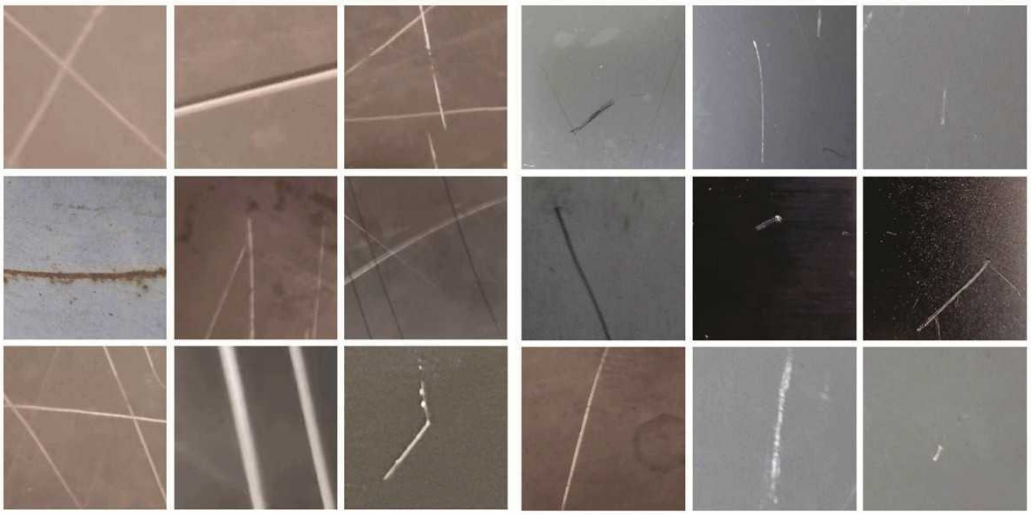


Figure 5. Examples of metal surface defects. This figure displays some of the metal surfaces and scratches of different shapes and colours present in the dataset. Notably, we have specifically highlighted some shorter scratches alongside some elongated ones. Additionally, some metallic surface textures can also be observed in the figure.

4.3. Indicator Assessment

During this assessment, the following four types of samples may be generated:

(1) TP (True Positive): this is used to represent the pixels that are labelled as scratch defects and detected as scratch defects by the model.
(2) TN (True Negative): this is used to represent the pixels that are labelled as non-scratch defects and detected as non-scratch defects by the model.
(3) FP (False Positive): this is used to represent the pixels that are labelled as non-scratch defects but detected as scratch defects by the model, indicating a false detection.
(4) FN (False Negative): this is used to represent the pixels that are labelled as scratch defects but detected as non-scratch defects by the model, indicating a missed detection.

In the experiments, we utilized the Intersection over Union (IoU) as the primary performance indicator to evaluate the proposed method. The calculation formula for the IoU is shown below:

$$IoU = \frac{TP}{TP + FP + FN} \tag{12}$$

In addition, the model assessment also involved the following three indicators for supplementary evaluations:

(1) Precision: This refers to the proportion of samples labelled as scratch defects among all scratch defect samples segmented by the model. A higher precision value indicates a more precise and reliable segmentation of scratch defects by the model. The expression of this indicator is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

(2) Recall: This refers to the proportion of pixels detected as scratch defects by the model among the pixels labelled as scratch defects in the samples. A higher recall value indicates that scratch defects were segmented more comprehensively by the model. The formula for this metric is provided below:

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

(3) F_1 Score: This takes into account both the precision and recall of the model, defined as their harmonic mean. The calculation formula is as follows:

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{15}$$

4.4. Ablation Experiment

To validate the effects of the loss functions and modules used in this method on the U-Net model, ablation experiments were conducted. The experimental setup and results are listed in Table 1.

Table 1. By comparing precision, recall, F1 score, and IoU, we demonstrate the benefits of each step in our model improvement.

	Ablations			**Metrics**			
Experiment	**Hybrid Loss**	**Our Downsampling**	**Our Attention**	**Precision**	**Recall**	**F_1 Score**	**IoU**
NO.1				0.8332	0.8990	0.8649	0.7501
NO.2	✓			0.8587	0.8642	0.8614	0.7665
NO.3	✓	✓		0.8776	0.8796	0.8786	0.7934
NO.4	✓		✓	0.8717	0.9012	0.8862	0.8027
NO.5	✓	✓	✓	0.8740	0.9046	0.8890	0.8039

In Experiment 1, the original U-Net model was trained on the metal scratch dataset 150 times. In Experiment 2, the proposed hybrid loss function was introduced in addition to the steps of Experiment 1. Comparatively, Experiment 2 showed a slight improvement in both precision and IoU. Notably, the IoU increased by 1.64%.

Based on Experiment 2, Experiment 3 incorporated our downsampling module. This module was based on SPD, fundamentally mitigating the loss of edge information caused by convolutions and pooling operations. Subsequently, a lightweight channel attention module computed the channel weights, followed by convolutions with a kernel size of 1×1 for the final processing. According to the experimental data, Experiment 3 maintained the recall as much as possible while improving the precision. Therefore, when compared to Experiment 1 and Experiment 2, the model from Experiment 3 achieved a higher F1 score, reaching 0.8786. Compared to Experiment 2, which only added the loss function, Experiment 3 achieved a further 2.69% increase in the IoU.

In Experiment 4, we integrated our attention mechanism into the skip connections between the encoder and decoder, in addition to the procedures outlined in Experiment 2. Based on frequency decomposition and lightweight self-attention, this approach largely preserved useful and detailed information and encoded global features implicitly. Compared to Experiments 1 and 2, Experiment 4 not only improved the precision and IoU but also exhibited a significant increase in recall. In terms of F1 score, it reached above 0.88 for the first time, specifically 0.8862. Meanwhile, the IoU was improved by 3.62% from that obtained in Experiment 2.

In Experiment 5, both the downsampling module and attention module were incorporated into the model while the steps of Experiment 2 were conducted. As a result, the model ensured high precision while achieving the highest defect recall among the five experiments, reaching the highest F1 score. Moreover, the IoU achieved in Experiment 5

reached the highest among all five experiments, presenting a 3.74% improvement from Experiment 2 and a 5.38% improvement from Experiment 1. These results demonstrate that our proposed method is an effective algorithm for enhancing the U-Net.

4.5. Contrast Experiment

4.5.1. Contrast Experiment on Scratch Data

To further validate the effects of the U-Net improvement in our proposed method on segmenting metal scratch defects, we compared this model with several common semantic segmentation models. The experimental results are provided in Table 2, with parameters, Multiply-Adds (MAdds), and floating-point operations (FLOPs) included as additional assessment indicators.

Table 2. This table presents a comparison of four selected models from the U-Net series and our model, demonstrating the superiority of our model in terms of its computational efficiency and IoU, among other factors.

Model	Params	MAdds	FLOPs	IoU
U-Net	1.97 G	3.93 G	1.97 G	0.7501
U-Net++ [33]	4.87 G	9.72 G	4.87 G	0.7834
U-Net with CBAM	8,599,178	5.03 G	2.52 G	0.7829
U-Net with Self-Attention	8,850,850	5.03 G	2.52 G	0.7940
Ours	8,502,210	4.95 G	2.48 G	0.8039

As shown in Table 2, our proposed model improved the IoU by 5.38% compared to the classic U-Net model, by 2.05% compared to the dense connection-based U-Net++, by 2.10% compared to the U-Net model fused with a Convolutional Block Attention Module (CBAM), and by 0.99% compared to the U-Net model fused with traditional self-attention mechanisms. Additionally, our proposed model was not only lightweight but also surpassed the U-Net model with traditional self-attention alone in its scratch segmentation IoU.

The IoU–epoch curves and loss–epoch curves generated by each model during training are illustrated in Figure 6. It is evident that our model consistently outperformed U-Net and U-Net++ in IoU throughout the training process. Due to foreground–background class imbalance and similar issues, all models exhibited low IoU values at the beginning of their training, but the loss–epoch curves show the faster convergence of our model. The U-Net fused with traditional self-attention began to overfit after the IoU reached its peak. However, our model incorporated a series of design improvements such as lightweighting the self-attention mechanism. Additionally, in the loss function, we applied weighted adjustments to hard-to-train samples. As a result, our model did not exhibit significant signs of overfitting in the later stages of training.

Figure 7(1) depicts the precision-recall curves of each model. As observed, our model yielded a higher precision in detecting defects than the other four models at a high recall. At a high recall, our model not only maintained a high precision but also experienced less precision decay than the other models, which is obvious in Figure 7(1). The experimental results combining Figure 7(1) and Table 2 indicate that the proposed improved U-Net model can achieve a higher IoU, recall, and precision.

Figure 7(2) provides the IoU–epoch curves generated during the training of our proposed model using BCE, D_{ice} loss, and our hybrid loss function, respectively. Notably, the model trained with the hybrid loss function achieved the best training performance in terms of the IoU. The BCE focuses solely on pixel classification errors, regardless of the discrepancy between the model's segmentation results and the true scratches. Therefore, it is necessary to introduce the D_{ice} loss function [2]. Additionally, as implied in Figure 7(2) and Table 1, it is essential to apply weighting to the difficult-to-classify samples when training the model, even though the D_{ice} function is less susceptible to foreground–background class imbalance.

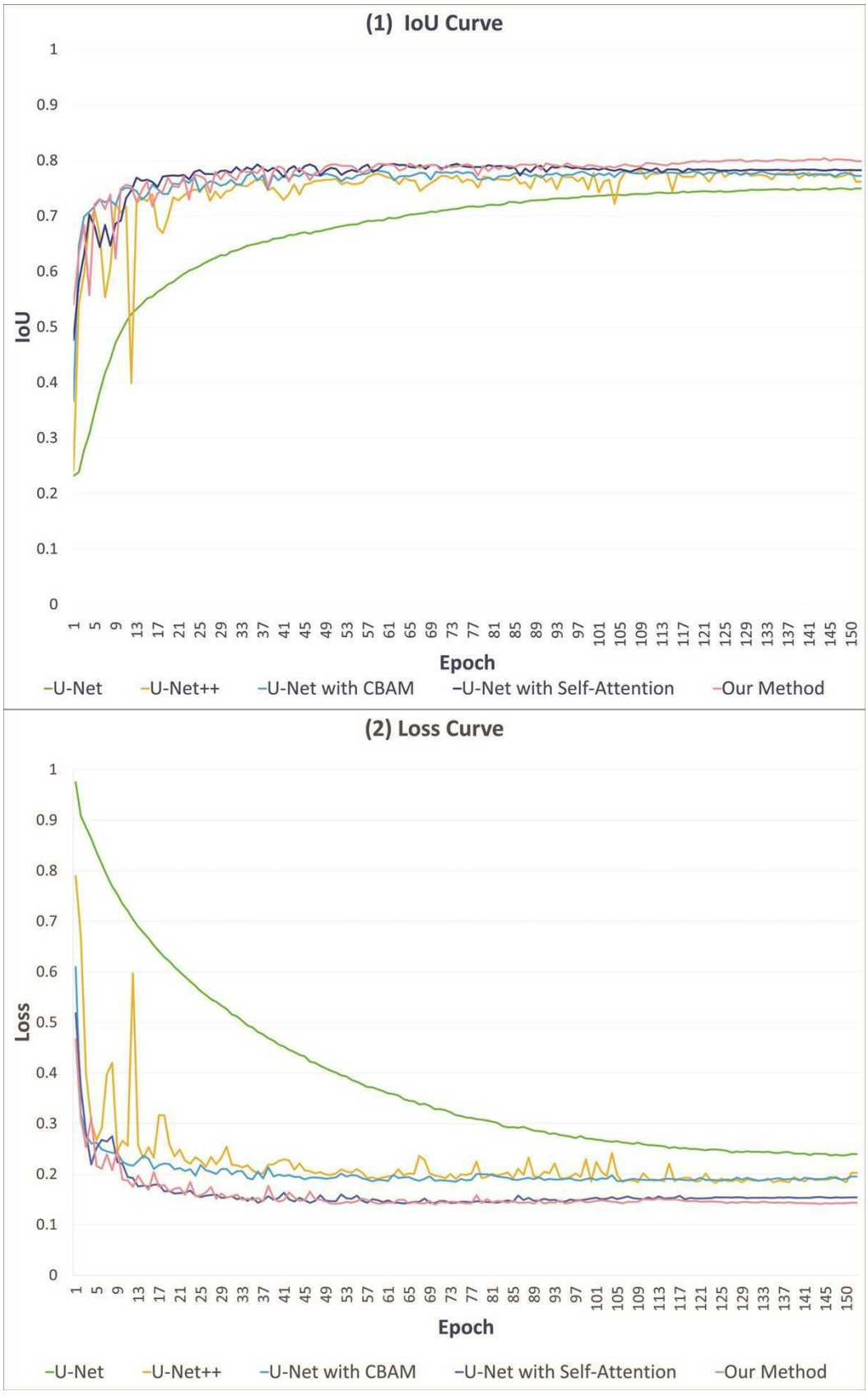


Figure 6. IoU and loss curve.

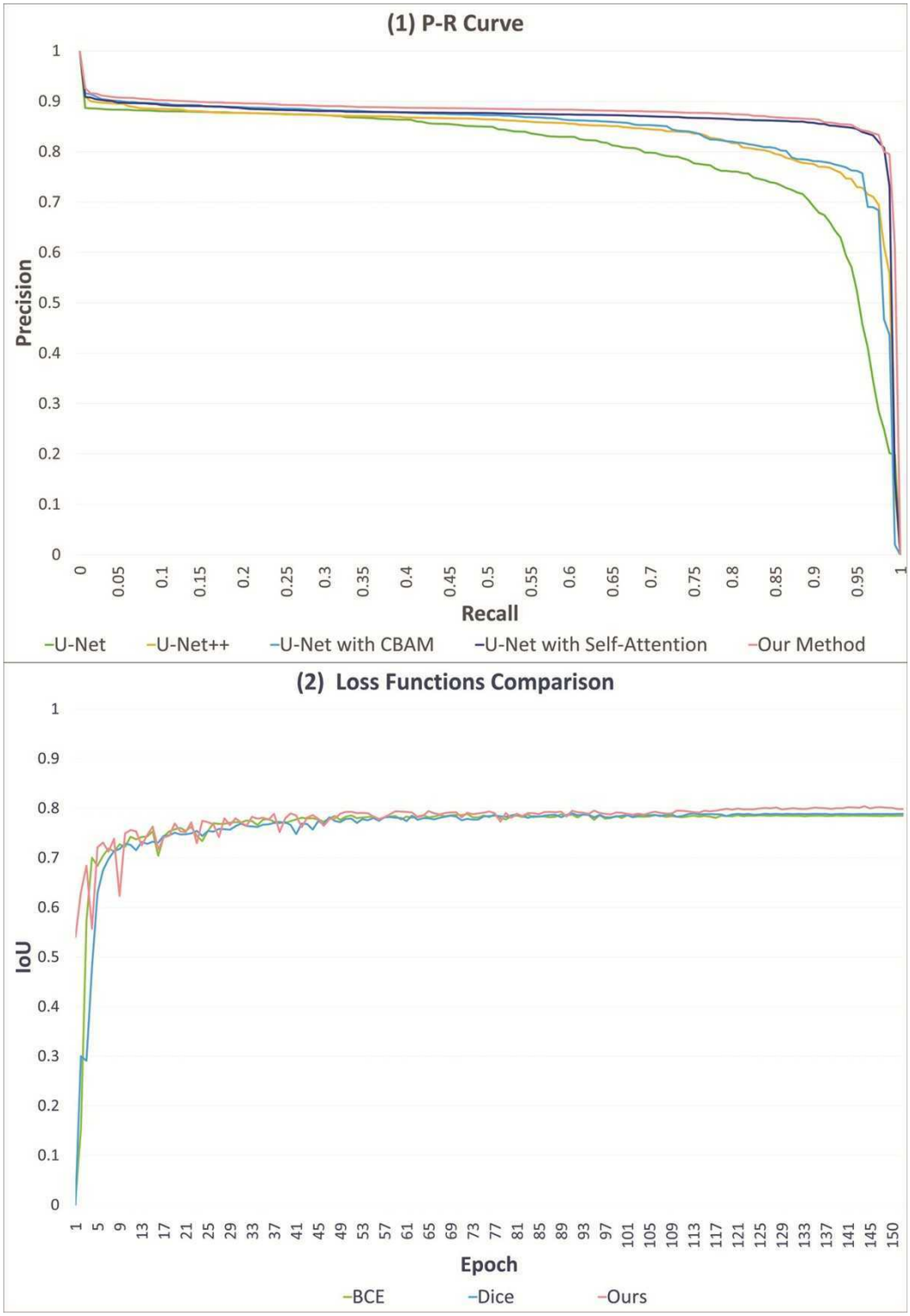


Figure 7. P-R curve and comparison of loss functions.

4.5.2. Contrast Experiment on Hot-Rolled Steel Strip Surface Defect Data

To further validate the segmentation capability of our model, we compared it with five of the most commonly used models in image segmentation. We utilized the NEU-Seg dataset [34] to evaluate the segmentation abilities of each model. The NEU-Seg dataset is a public segmentation dataset of surface defects in hot-rolled steel strips [34]. Compared to our dataset, the NEU-Seg dataset features a wider variety of targets, enabling us to assess whether our model's ability to segment targets extends beyond just scratches. This dataset

comprises three types of defects, with 300 examples of each type given. In our experiment, we did not differentiate among the three types of defects. That is to say, we considered the three types of defects as 1, and non-defective areas as 0.

We selected UNet, UNet++, DeepLabv3+ [35], and PSPNet [36] to conduct a comparative experiment with our proposed model. Prior to the experiment, the models underwent a 100-epoch pretraining phase utilizing our dataset of metal surface scratch defects. The experimental results are shown in Table 3. Notably, DeepLabv3+$_{(1)}$ denotes the use of MobileNetV2 [37] as its backbone, whereas DeepLabv3+$_{(2)}$ denotes the use of ResNet101 [38] as its backbone. As shown in Table 3, our model achieves the highest IoU compared to other segmentation methods, without requiring extensive hardware resources or high computational power. The segmentation IoU of our model is significantly higher than that of UNet and UNet++. Compared with DeepLabv3+$_{(1)}$, which uses MobileNetV2 as its backbone, our model achieves a 4.98% higher IoU score. In comparison to DeepLabv3+$_{(2)}$, which has ResNet101 as its backbone, our model consumes fewer hardware resources, has lower computational complexity, and achieves a higher IoU. When compared to PSPNet, our model utilizes 70% of its hardware resources and only 1/3 of its computation, resulting in a segmentation IoU that is 1.14 percent higher.

Table 3. Based on the NEU-Seg dataset, we compared our model to the five most commonly used models to further verify the advantages of our model in terms of computational efficiency and IoU. This table shows the results of the experiment.

Model	Memory	FLOPs	IoU
U-Net	28.09 MB	1.97 G	0.6991
U-Net++	56.78 MB	4.87 G	0.7082
DeepLabv3+$_{(1)}$	22.10 MB	629.65 M	0.7097
DeepLabv3+$_{(2)}$	35.18 MB	2.79 G	0.7392
PSPNet	48.16 MB	7.14 G	0.7481
Ours	34.45 MB	2.48 G	0.7595

To sum up, our proposed MSDD-UNet demonstrated a competitive performance in metal scratch segmentation and other similar segmentation tasks. The proposed downsampling module, attention mechanism, and loss functions can also be utilized in similar segmentation tasks, facilitating the development of an end-to-end segmentation network.

5. Conclusions

On account of metal surface scratch datasets containing class imbalance, this paper proposes an MSDD-UNet-based scratch defect localization algorithm for pixel-level segmentation. Extensive analyses and contrast experiments validated that our proposed segmentation network can accurately identify and localize scratch defects that contain class imbalance. The main contributions of this paper are as follows: (1) a downsampling mode based on an SPD module and LCAM is proposed to overcome the loss of contextual information after multiple convolutions and pooling operations; (2) based on image frequency decomposition and a lightweight self-attention module, the attention module is improved and then integrated into the skip connections to improve the model's precision in localizing scratch defects; (3) to address the foreground–background class imbalance problem in scratches, a hybrid loss function integrating the advantages of both the focal and D_{ice} functions is introduced to realize precise segmentation training.

Author Contributions: Y.L.: methodology and research design; Y.Q.: resources; Z.L.: verification and manuscript writing; H.X.: resources; C.W.: verification and software. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article. The data supporting the findings of this study are available upon request from the first author, [Liu], upon reasonable request.

Acknowledgments: Throughout the writing of this dissertation, I have received a great deal of support and assistance. First and foremost, I would like to express my gratitude to my two supervisors, Qin and Xia, whose expertise was invaluable in formulating my research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. I would particularly like to acknowledge my team members, Lin and Wang, for their exceptional collaboration and patient support. Additionally, I would like to thank my families for their wise counsel and sympathetic ear. Last but not least, I could not have completed this dissertation without the support of my friends, who provided stimulating discussions as well as happy distractions to rest my mind outside of my research. You are always there for me.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sunkara, R.; Luo, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer Nature: Cham, Switzerland, 2022.
2. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
3. Milletari, F.; Navab, N.; Ahmadi, S.A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016.
4. Metals Engineering Institute. *Fundamentals of Nondestructive Testing: Lessons 10–15[M]*; American Society of Metals: Newbury, OH, USA, 1972.
5. Ono, H.; Ogawa, A.; Yamasaki, T.; Koshihara, T.; Kodama, T.; Iizuka, Y.; Oshige, T. Twin-illumination and subtraction technique for detection of concave and convex defects on steel pipes in hot condition. *ISIJ Int.* **2019**, *59*, 1820–1827. [CrossRef]
6. Liu, H.W.; Lan, Y.Y.; Lee, H.W.; Liu, D.K. Steel surface in-line inspection using machine vision. *First Int. Workshop Pattern Recognit.* **2016**, *10011*, 187–191.
7. Yang, J.; Li, X.; Xu, J.; Cao, Y.; Zhang, Y.; Wang, L.; Jiang, S. Development of an optical defect inspection algorithm based on an active contour model for large steel roller surfaces. *Appl. Opt.* **2018**, *57*, 2490–2498. [CrossRef] [PubMed]
8. Yi, L.; Li, G.; Jiang, M. An end-to-end steel strip surface defects recognition system based on convolutional neural networks. *Steel Res. Int.* **2017**, *88*, 1600068. [CrossRef]
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 84–90. [CrossRef]
10. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
11. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015*; Proceedings, part III 18; Springer International Publishing: Cham, Switzerland, 2015.
12. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Houlsby, N. An Image is Worth 16 $\times$ 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
13. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef]
14. Yan, Z.; Shi, B.; Sun, L.; Xiao, J. Surface defect detection of aluminum alloy welds with 3D depth image and 2D gray image. *Int. J. Adv. Manuf. Technol.* **2020**, *110*, 741–752. [CrossRef]
15. Moosavian, A.; Bagheri, E.; Yazdanijoo, A.; Barshooi, A.H. An Improved U-Net Image Segmentation Network for Crankshaft Surface Defect Detection. In Proceedings of the 2024 13th Iranian/3rd International Machine Vision and Image Processing Conference (MVIP), Tehran, Iran, 6–7 March 2024.
16. He, Z.; Zhao, J.; Zhao, X. Scratch Defects Detection of Curved Metal Surface based on Multiple High Frequency Projection and Inverse Transfer Learning. *IEEE Trans. Instrum. Meas.* **2024**. [CrossRef]
17. Lema, D.G.; Usamentiaga, R.; García, D.F. Enhancing automated inspection in metal industries: Zero-shot segmentation of surface defects using bounding box prompts. *Meas. Sci. Technol.* **2024**, *35*, 085604. [CrossRef]
18. Song, Y.; Xia, W.; Li, Y.; Li, H.; Yuan, M.; Zhang, Q. AnomalySeg: Deep Learning-Based Fast Anomaly Segmentation Approach for Surface Defect Detection. *Electronics* **2024**, *13*, 284. [CrossRef]
19. Arafin, P.; Billah, A.M.; Issa, A. Deep learning-based concrete defects classification and detection using semantic segmentation. *Struct. Health Monit.* **2024**, *23*, 383–409. [CrossRef] [PubMed]

20. Ardiyanto, I. Edge devices-oriented surface defect segmentation by GhostNet Fusion Block and Global Auxiliary Layer. *J. Real-Time Image Process.* **2024**, *21*, 13. [CrossRef]
21. Kong, D.; Hu, X.; Gong, Z.; Zhang, D. Segmentation of void defects in X-ray images of chip solder joints based on PCB-DeepLabV3 algorithm. *Sci. Rep.* **2024**, *14*, 11925. [CrossRef] [PubMed]
22. Feng, H.; Song, K.; Cui, W.; Zhang, Y.; Yan, Y. Cross position aggregation network for few-shot strip steel surface defect segmentation. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–10. [CrossRef]
23. Zhou, Z.; Yan, L.; Zhang, J.; Zheng, Y.; Gong, C.; Yang, H.; Deng, E. Automatic segmentation of tunnel lining defects based on multiscale attention and context information enhancement. *Constr. Build. Mater.* **2023**, *387*, 131621. [CrossRef]
24. Kumar, D.D.; Fang, C.; Zheng, Y.; Gao, Y. Semi-supervised transfer learning-based automatic weld defect detection and visual inspection. *Eng. Struct.* **2023**, *292*, 116580. [CrossRef]
25. Zhao, D.; Wang, C.; Gao, Y.; Shi, Z.; Xie, F. Semantic Segmentation of Remote Sensing Image Based on Regional Self-Attention Mechanism. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 8010305. [CrossRef]
26. Ghiasi, G.; Lin, T.Y.; Le, Q.V. DropBlock: A regularization method for convolutional networks. *Neural Inf. Process. Syst.* **2018**, *31*, 10750–10760.
27. Chen, Y.; Fan, H.; Xu, B.; Yan, Z.; Kalantidis, Y.; Rohrbach, M.; Feng, J. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
28. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020.
29. Koziarski, M.; Cyganek, B. Impact of low resolution on image recognition with deep neural networks: An experimental study. *Int. J. Appl. Math. Comput. Sci.* **2018**, *28*, 735–744. [CrossRef]
30. Chen, Z.; Xu, Q.; Cong, R.; Huang, Q. Global context-aware progressive aggregation network for salient object detection. *Proc. Aaai Conf. Artif. Intell.* **2020**, *34*, 10599–10606. [CrossRef]
31. Zhao, J.X.; Liu, J.J.; Fan, D.P.; Cao, Y.; Yang, J.; Cheng, M.M. EGNet: Edge Guidance Network for Salient Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8778–8787. [CrossRef]
32. Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
33. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Proceedings of the 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, 20 September 2018*; Proceedings 4; Springer International Publishing: Cham, Switzerland, 2018.
34. Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surf. Sci.* **2013**, *285*, 858–864. [CrossRef]
35. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
36. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
37. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

remote sensing

MDPI

Article

An Auditory Convolutional Neural Network for Underwater Acoustic Target Timbre Feature Extraction and Recognition

Junshuai Ni [1], Fang Ji [1,2,*], Shaoqing Lu [1] and Weijia Feng [1]

[1] China Ship Research and Development Academy, Beijing 100101, China; nijunshuai2022@163.com (J.N.)
[2] College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China
* Correspondence: heujifang@163.com

Abstract: In order to extract the line-spectrum features of underwater acoustic targets in complex environments, an auditory convolutional neural network (ACNN) with the ability of frequency component perception, timbre perception and critical information perception is proposed in this paper inspired by the human auditory perception mechanism. This model first uses a gammatone filter bank that mimics the cochlear basilar membrane excitation response to decompose the input time-domain signal into a number of sub-bands, which guides the network to perceive the line-spectrum frequency information of the underwater acoustic target. A sequence of convolution layers is then used to filter out interfering noise and enhance the line-spectrum components of each sub-band by simulating the process of calculating the energy distribution features, after which the improved channel attention module is connected to select line spectra that are more critical for recognition, and in this module, a new global pooling method is proposed and applied in order to better extract the intrinsic properties. Finally, the sub-band information is fused using a combination layer and a single-channel convolution layer to generate a vector with the same dimensions as the input signal at the output layer. A decision module with a Softmax classifier is added behind the auditory neural network and used to recognize the five classes of vessel targets in the ShipsEar dataset, achieving a recognition accuracy of 99.8%, which is improved by 2.7% compared to the last proposed DRACNN method, and there are different degrees of improvement over the other eight compared methods. The visualization results show that the model can significantly suppress the interfering noise intensity and selectively enhance the radiated noise line-spectrum energy of underwater acoustic targets.

Keywords: gammatone filter bank; attention mechanism; line spectrum; auditory convolutional neural network; global energy pooling; underwater acoustic target recognition

1. Introduction

Underwater acoustic target recognition (UATR) is a research hotspot in the field of passive sonar and an internationally recognized technical difficulty in underwater acoustic signal processing, which has a significant demand in harbor and waterway monitoring and other aspects. Affected by sound scattering, ocean background noise, and the complex underwater environment, the UATR technology is meeting great challenges [1]. In recent years, with the rapid development of artificial intelligence technology, deep learning has been gradually replacing traditional classification methods and is widely used in UATR with remarkable achievements.

Compared with other deep learning models such as deep neural networks, recurrent neural networks, and deep belief networks, a convolutional neural network (CNN) is the most widely used in UATR due to its powerful deep feature extraction capability [2–4]. Researchers usually transform underwater acoustic target-radiated noise time-domain signals into time-frequency maps, such as the LOFAR spectrum [5], DEMON spectrum [6], and Meier spectrum [7], and then refer to the computer vision method to achieve UATR using two-dimensional (2d) CNN such as MobileNet [8], ResNet [9], VGG [10], DensNet [11], etc.

In order to suppress the background noise interference and extract the critical features of the target, attention mechanisms, including a channel attention mechanism, spatial attention mechanism, and self-attention mechanism, are added in the CNN model to select the deep features adaptively according to different weights [12–14], which effectively improves the accuracy of target recognition. In order to downscale the scale of model parameters to reduce the amount of computation, some scholars have improved the classical models structurally by introducing advanced structures such as depth-separable convolutional layers and global average pooling layers to give many new deep learning models for UATR [15]. However, there are very few samples currently available for UATR model training, which leads to serious model over-fitting. To address this problem, a generative adversarial network (GAN) is used for underwater acoustic target data augmentation prior to UATR, which utilizes mutual confrontation between the generator and the discriminator to generate a large number of new samples similar to the real data, thus enlarging the size of the underwater acoustic target sample set [16–18].

The process of converting time-domain signals into time-frequency maps will lose part of the feature information, which may reduce the accuracy of target recognition. For this reason, some end-to-end recognition methods based on time-domain signals and a 1d CNN model are proposed and proved to be effective [19–21]. Compared with the 2d CNN, the 1d CNN usually has fewer model parameters and has the advantages of low computational cost and high stability. In addition, the 1d CNN has a shorter duration of input data, which means that more samples can be generated from the same dataset, and therefore the model can be trained more adequately [22], resulting in better UATR performance. In addition, a flexible combination of convolutional kernels with different sizes is used to extract multi-scale features of underwater acoustic signals to enhance the representation of target characteristics. Due to the current insufficient human knowledge of the underwater acoustic target's inherent properties, researchers are overly concerned with accuracy improvement, but pay little attention to the interpretability of the features extracted by deep learning, which leads to poor model generalization and low recognition accuracy in complex environments [23]. It is no exaggeration to say that the recognition performance of all current deep learning models is far from that of sonarmen.

Neuroscience researchers have found that the human ear is very sensitive to different sounds and has a masking effect that allows humans to distinguish acoustic targets through slight differences and perceive critical information about specific timbres in noisy environments [24]. This phenomenon provides inspiration for using deep learning to mimic sonarmen processing hydroacoustic signals. In this paper, the characteristics of underwater acoustic targets are analyzed based on LOFAR spectrum from real data and elucidate the specific sources of line spectra on different frequency bands. The timbre feature extraction mechanisms of the human auditory system are summarized, and an end-to-end deep learning model is proposed for UATR: Auditory Convolutional Neural Network (ACNN). It realizes line-spectrum frequency perception, line-spectrum energy enlargement, critical line-spectrum selecting, and line-spectrum features fusion, and thus realizes timbre feature extraction from audio signals. Secondly, a new method of constructing the channel attention mechanism is proposed in designing the network model to globally pool the feature map using the sub-band energy based on the underwater acoustic target signal characteristics, which has a clearer physical meaning compared to the global average pooling (GAP) and global maximum pooling (GMP). Finally, simulating the information processing and decision-making functions on the brain, a one-dimensional deep convolutional network is used to extract deep abstract features of different targets and predict the categories of underwater acoustic targets.

This paper is organized as follows: The underwater acoustic target line-spectrum characteristics are explained in Section 2. Section 3 introduces the human auditory mechanism, gives a detailed introduction to the structure and design ideas of the ACNN model, and provides a new UATR method based on ACNN. The experiment details, including an

experimental dataset, experimental process and experimental results are shown in Section 4. An overall discussion is concluded in Section 5.

2. Line Spectra of Underwater Acoustic Targets

Underwater acoustic target-radiated noise is a kind of strong periodic and non-smooth signal consisting of mechanical noise, propeller noise and hydrodynamic noise, whose power spectrum is composed of line spectra and continuous spectra. Line spectra are frequency components with higher energy than continuous spectra and a stable amplitude over a period of time, and they are mainly generated by mechanical operation (both reciprocating and rotating) and structural vibration. The frequency of the line spectra is related to the vibration frequency or operating speed. Time-frequency line spectra diagrams of four typical underwater acoustic targets are shown in Figure 1.

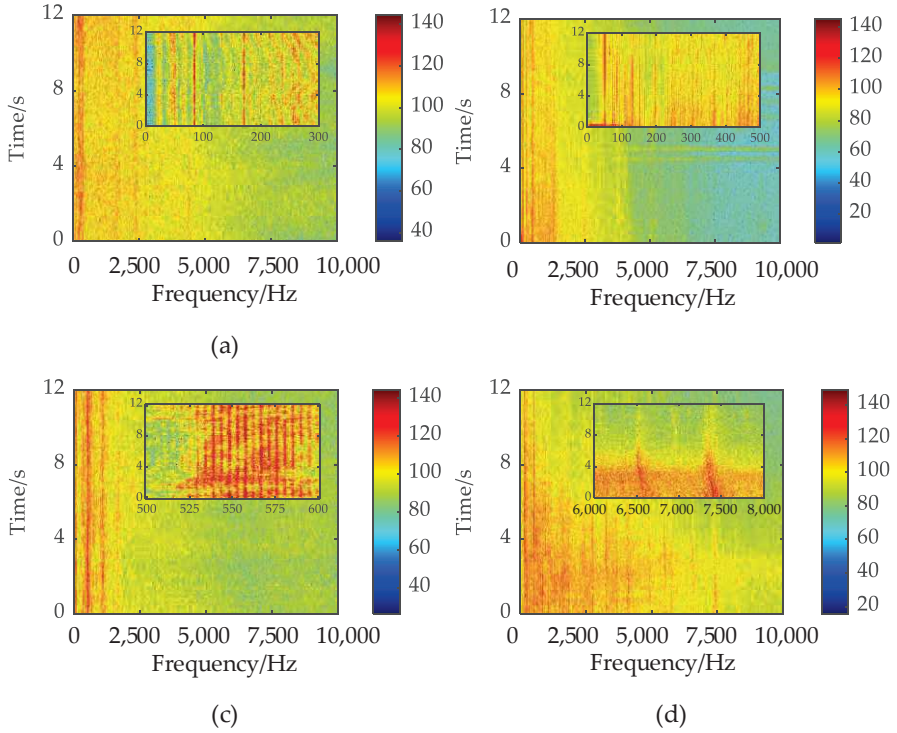


Figure 1. Time-frequency line-spectra diagrams. (**a**) Time-frequency diagram of the small boat. (**b**) Time-frequency diagram of the test vessel in a stationary state with only auxiliary machinery operation. (**c**) Time-frequency diagram of the fishing vessel with shaft system failure. (**d**) Time-frequency diagram of the motor boat-radiated noise when its propeller is rotating at a high speed.

The time-frequency diagram of the small boat is shown in Figure 1a. This boat has a diesel engine with four cylinders and a speed of 2500 revolutions per minute (rpm) when the radiated noise is measured in experiments, which means that each cylinder fires 20.8 times per second, so there are four strong line spectra at 20.8 Hz, 41.6 Hz, 62.4 Hz, and 83.2 Hz. It is easy to see that the fundamental frequency value and the number of line spectra correspond to the rotational speed of the motor and the number of its cylinders, respectively, which are the essential properties of underwater acoustic targets. Since the engine noise intensity is unsteady over long periods of time, there are several harmonic line spectra in the time-frequency diagram with a fundamental frequency of 20.8 Hz and an order greater than four, but the maximum frequency of line spectra generated by diesel engine is generally not greater than 300 Hz.

The time-frequency diagram of the test vessel in a stationary state with only auxiliary machinery operation is shown in Figure 1b. It can be seen that there is a very strong line spectrum at 50 Hz, which has the same frequency with the current output from the

diesel generator. In addition, pumps and other auxiliary machines produce a very large number of line spectra below 500 Hz, and these components are easily drowned by the main engine noise during ship navigation but are very significant in the radiated noise of non-diesel-powered ships.

The time-frequency diagram of the fishing vessel with shaft system failure is shown in Figure 1c. The bearing periodically collides and fricates with the base during rotation, thus generating rhythmic radiated noise with an intensive line-spectrum cluster normally in the range of 300 to 1000 Hz, while the center frequency of this vessel is 560 Hz. The frequency difference between any two adjacent line spectra in the cluster is approximately equal and equal to the propeller shaft frequency. This is a frequency-accompanying phenomenon resulting from the coupling of the line spectra, reflecting the vibration frequency of the ship structure and the line spectra reflecting the rotational speed of the shaft, which is a common characteristic of civilian ships.

The time-frequency diagram of the motor boat-radiated noise when its propeller rotates at a high speed is shown in Figure 1d. Along with the propeller rapidly cutting the water, its blades resonate and produce radiated noise with some high-frequency line spectra, and the frequencies of these line spectra are generally more than 1000 Hz, and the highest frequency even up to 8000 Hz. This motor boat has line spectra, respectively, at 6500 Hz and 7500 Hz, which are typical features significantly different from other classes of targets.

3. ACNN

3.1. Auditory Mechanisms

Listeners usually distinguish different sounds containing similar frequency components and the same loudness by their timbre, which can be simply represented by the structure of the energy distribution of a sound in terms of frequency [25]. For underwater acoustic targets, mechanical noise, propeller noise, bearing noise, and structure vibration noise each have characteristic line spectra. Different targets have similar line-spectra frequency bands, but the frequency energy distribution structures are different, thus being an important feature for distinguishing them. Sonarmen recognize different underwater acoustic targets mainly by sensing their timbre features.

The human auditory system is a multilevel complex system with the ability to perceive sound information, which is mainly composed of the cochlea, auditory midbrain, auditory thalamus and auditory cortex. When the auditory system perceives sound, the cochlea basilar membrane first breaks down the sound into different frequency components based on a specific excitation response, and then neurons in the primary auditory cortex that correspond to the frequencies are activated by the complex frequency components contained in the sound, which have a specific sensitivity and are connected with each other to form a frequency topology [26]. After that, different frequency components from the activation mode of the primary auditory cortex are integrated and processed by the secondary auditory cortex to get the timbre information of the sound. When people focus on hearing a certain timbre in a noisy environment, in accordance with the masking effect, the primary auditory cortex allows all frequency information to pass through, but in the secondary auditory cortex, only the neurons related to this timbre can be activated [27]. In addition, as the sound intensity changes over time, the activation mode of auditory cortical neurons also changes with the change of signal timbre energy [28]; that means the auditory cortex is more sensitive to transient and mutant audio signals. Finally, the higher cerebral cortex extracts deep features of all the timbre information integrated by the auditory cortex for final recognition.

According to the above research results in the field of neuroscience, the human auditory perception mechanism can be simply summarized into the following three aspects: (1) decomposing the acoustic signal into different frequency bands; (2) extracting timbre features and focusing on important information; and (3) using deep features to recognize acoustic targets.

3.2. Model Structure and Theory

Inspired by auditory mechanisms, this paper attempts to simulate the process of human auditory perception with deep learning algorithms and proposes a model for underwater acoustic target recognition, ACNN, which is composed of a frequency-component perception module, timbre perception module, and critical-information perception model. The model approximately emulates the whole process of acoustic signals being perceived by the auditory system, and its structure is shown in Figure 2.

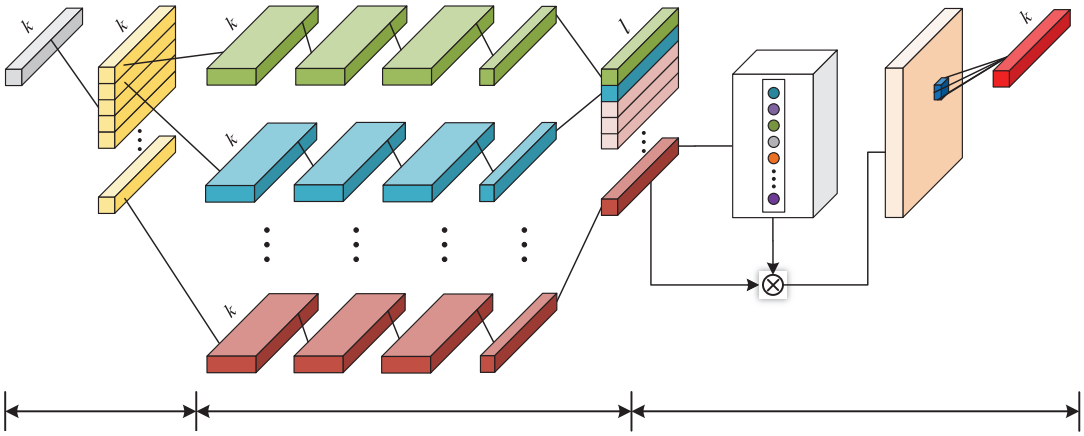


Figure 2. ACNN model structure.

3.2.1. Frequency Perception Module

In order to give the ACNN model initial guidance to extract timbre features of underwater acoustic targets and achieve signal decomposition, the gammatone filter bank with a similar excitation response to that of the cochlear basilar membrane is used in the frequency perception module, which is shown in Equation (1).

$$g(t) = at^{n-1}e^{-2\pi bt}cos(2\pi ft + \varphi) \tag{1}$$

where a is amplitude, n is filter's order, t is time in second, B is bandwidth in Hz, φ is phase of the carrier in radians, and f is center frequency in Hz. f and B are set according to an equivalent rectangular bandwidth (ERB) filter bank cochlea model in Equations (2) and (3).

$$ERB(f) = 24.7(4.37f/1000 + 1) \tag{2}$$

$$B = 1.019 \times ERB(f) \tag{3}$$

In this module, according to the line-spectrum frequency distribution characteristics of underwater acoustic targets, 128 gammatone filter channels are generated for sensing frequency information with center frequencies ranging from 10 Hz to 10 kHz, and the frequency magnitude responses of the gammatone filter bank are shown in Figure 3.

It can be seen in Figure 3 that the center frequency structures of the gammatone filter bank are approximately linearly distributed in the frequency band of 10~300 Hz, and the difference between two adjacent center frequencies is about 10 Hz, which shows a good frequency resolution performance of the low-frequency line spectra generated by mechanical noise. Moreover, in the frequency band higher than 300 Hz, the center frequency interval and finite bandwidth of the gammatone filter bank are increasing, so

that the high-frequency line spectra generated by bearing friction and propeller resonance can be detected efficiently with less computational cost.

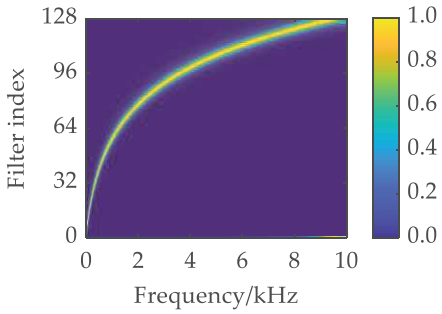


Figure 3. The frequency magnitude responses of the gammatone filter bank.

3.2.2. Timbre Perception Module

In the timber perception module, a convolutional block consisting of four 1d convolutional layers is connected after one of the gammatone filter output channels. The first three convolutional layers all have 16 channels with a convolutional kernel size of 5×1, which are used to extract line-spectrum energy distribution features in different frequency bands and also to filter the marine ambient noise. The fourth convolutional layer has only one channel with a convolutional kernel size of 1×1, which is used to change the shape of its input data and fully fuse underwater acoustic target timber features. In this module, the combination of two different-sized convolutional kernels also enables multi-scale feature extraction, allowing the model to better adapt to the effects of target velocity variations. Each gammatone filter channel has a convolutional block connected to it and the operation of the convolutional blocks can be described as follows.

$$y_n = h(w_n x_n + b_n),\ y_n \in R^{l \times 1}, n = 1, 2, \cdots, N \tag{4}$$

where x_n is the output of the nth gammatone filter channel, y_n, w_n and b_n are the output, weights matrix, and bias matrix of the nth convolutional block, respectively, h is the tanh activation function which limits the output value of the convolutional block to a range of -1 to 1, so as to ensure that the numerical range of the output data is the same as the numerical range of the original signal, and it is expressed as

$$tanh(\alpha) = \frac{e^{\alpha} - e^{-\alpha}}{e^{\alpha} + e^{-\alpha}} \tag{5}$$

Compared to the commonly used Relu function, the tanh function better preserves the original information of the input signal while introducing nonlinear operations. Driven by a large amount of underwater acoustic target-radiated noise data, the kernel parameters of convolutional blocks are optimized, so that interfering noise unrelated to target characteristics is eliminated and the line-spectrum components of underwater acoustic targets will be strongly enhanced.

3.2.3. Critical Information Perception Module

In this module, a concatenate layer is first used to concatenate the outputs of every convolutional block, obtaining a feature map Y with N channels, which is shown in Equation (6).

$$Y = \text{concatenate}(y_1, y_2, \cdots, y_N),\ Y \in R^{l \times 1 \times N} \tag{6}$$

It is worth mentioning that, unlike the feature map output from convolutional layers, the feature map here has practical significance, as each channel corresponds to a different frequency band. On this basis, the channel attention (CA) [12] module whose structure is shown in Figure 4 is cleverly used to generate a weight vector, weighting all frequency

channels in order to select a combination of frequency bands that is more critical to the target recognition task, achieving secondary extraction of timbre features. This operation is represented as

$$Z_C = Y\sigma = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_N \end{bmatrix} \tag{7}$$

where σ is the weight vector generated by the CA module, $\rho \in R^{l\times 1}$, and Z_S is the output of the CA module, $Z_S \in R^{l\times 1\times N}$. It can be seen from Figure 4 that the weight matrix σ is derived from the input feature map Y through the computation of the global pooling layer, the dense connectivity layer and the nonlinear activation layer. As the most critical step of the channel attention mechanism, the global pooling operation is currently divided into two types: GAP and GMP, and their role is to downscale the high-dimensional feature maps along the channel axes to obtain a set of one-dimensional vectors with dimensions equal to the number of channels, which is shown in Figure 5.

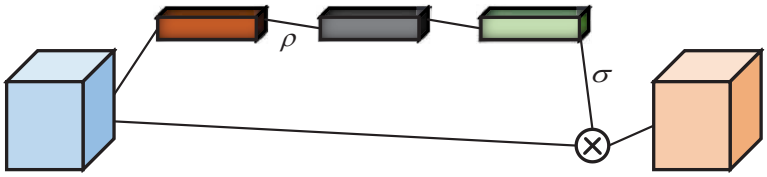


Figure 4. Channel attention mechanism.



Figure 5. Structure of the global pooling layer.

GAP and GMP are obtained by averaging and maximizing the feature maps along the channel axes, respectively, and these two pooling methods are widely used in the image processing field due to their excellent feature abstraction capabilities. However, in this paper, unlike traditional image processing, the feature maps are three-dimensional vectors with actual physical meaning, and each layer corresponds to the component of the original signal in a certain frequency band, so GAP and GMP are not applicable here. The specific reasons are as follows:

(1) The underwater acoustic target-radiated noise signal can be viewed as a superposition of simple harmonic components, and the average value of any sub-band signal is roughly 0. Therefore, sub-band features cannot be effectively extracted using global average pooling.
(2) Although the maximum value of each sub-band signal can characterize the properties of the target, the result of the maximum pooling operation is highly random due to the interference of ambient noise.

In this paper, a new weight matrix calculation method called a global energy pooling (GEP) layer is proposed in the channel attention module to achieve the operation of

dimension reduction in feature maps by replacing the mean and maximum computation with normalized energy computation. The output value of GEP is represented as

$$\rho_n = \frac{y_n^2(1) + y_n^2(2) + \cdots + y_n^2(l) + \cdots + y_n^2(L)}{L}, n = 1, 2, \cdots, N \tag{8}$$

Then, the GEP layer output vector is processed through two dense layers and activated by a sigmoid function to obtain a weight vector taking values between 0 and 1. Equation (9) is the expression of the sigmoid activation function.

$$sigmoid(\alpha) = \frac{1}{1 + e^{-\alpha}} \tag{9}$$

After that, the feature map output from the CA module with a dimension of $l \times 1 \times N$ is input into a one-dimensional convolutional layer whose channel number is 1 and kernel size is 1×1 to achieve data compression, obtaining the output of the ACNN model after the tanh function activation.

3.3. ACNN_DRACNN

In Ref. [12], an UATR method based on a deep residual attention convolutional neural network (DRACNN) was proposed by our team in July 2023, which achieved 97.1% recognition accuracy on the ShipsEar dataset [16] and had excellent generalization on the DeepShip dataset with a recognition accuracy of 89.2% [17]. The DRACNN model consists of five residual attention convolution blocks (RACB) connected in series, and it has the same input data dimensions as the ACNN output data dimensions, which ensures the connectivity of data transfer between the auditory and decision models. In this paper, the DRACNN model is used to simulate the process of abstract feature extraction and decision-making in the human cerebral cortex, and it is connected behind the ACNN model which is used to simulate the human auditory perception mechanism to obtain a new model called ACNN_DRACNN, finally achieving UATR by a Softmax classifier. The structure of the ACNN_DRACNN model is shown in Figure 6.

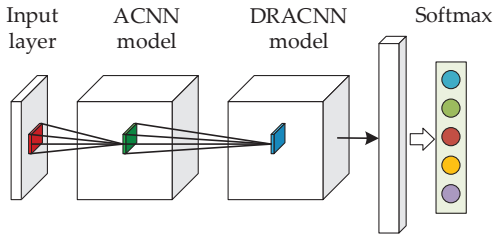


Figure 6. ACNN_DRACNN model structure.

The input shape of the ACNN_DRACNN model is set to 4096×1; thus, the number of parameters and floating-point operations are 0.61 million (M) and 1.30 M, respectively. It should be noted that the number of parameters and floating-point operations of the ACNN model are 0.35 M and 0.69 M, respectively.

4. UATR Experiment

4.1. Introduction to the Dataset

In this paper, performances of the ACNN model in line-spectrum feature extraction and target recognition are validated on the ShipsEar dataset, which is available at the web address http://atlanttic.uvigo.es/underwaternoise/ (accessed on 15 March 2023). The ShipsEar dataset was selected from audio recordings collected under different sea conditions in Vigo Harbor, Spain. This dataset covers test data from the fall of 2012 through the summer of 2013, and contains 91 sound recordings of 11 vessel types including fishing boats, ocean liners, trawlers, mussel boats, tugboats, dredgers, etc., and one background

noise class. The recordings were made with autonomous acoustic digitalHyd SR-1 recorders, with a total duration of 3 h and 10 min and a sampling rate of 52,734 Hz. As shown in Table 1, ship targets in this dataset were categorized into five categories by vessel length.

Table 1. ShipsEar data recognition details (Duration in seconds).

Category	Type of Vessel	Files	Duration
A	Fishing boats, trawlers, mussel boats, tugboats, dredgers	17	1880
B	Motor boats, pilot boats, sailboats	19	1567
C	Passenger ferries	30	4276
D	Ocean liners, ro-ro vessels	12	2460
E	Background noise recordings	12	1145

4.2. Experiments and Analysis of Results

Firstly, the data in the ShipsEar dataset are resampled to 20 kHz, which effectively retains the line-spectrum information below 10 kHz and doubly increases the duration of the signal at a certain sampling point number. Then, all the data in the ShipsEar dataset are divided into frames according to 4096 points per frame, and 2048 points overlap between two adjacent frames, so the sample duration of each frame is about 0.2 s. After each sample is zero-averaged and normalized, which is described in (10), a sample set containing 16,537 samples is obtained. Next, 80% of all samples are randomly selected as the training set and the remaining 20% as the test set, so there are 13,230 samples in the training set and 3307 samples in the test set, and it can be seen that this sample set is unbalanced from Table 2.

$$p(n) = \frac{s(n) - \frac{1}{4096}\sum_{j=1}^{4096} s(j)}{\max\left(\left|s(n) - \frac{1}{4096}\sum_{j=1}^{4096} s(j)\right|\right)}, n = 1, 2, \cdots, 4096 \tag{10}$$

Table 2. Recognition results of each class.

Category	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
A	99.83	99.83	99.83	99.83
B	99.82	99.99	99.82	99.91
C	99.99	99.72	99.99	99.86
D	99.87	99.87	99.87	99.87
E	99.85	99.99	99.85	99.92
Average	99.87	99.88	99.87	99.88

The ACNN_DRACNN model proposed in this paper is built in a deep learning development environment with the Windows 10 operating system, Python-3.6.5, Keras-2.2.4, TensorFlow-1.14.0, Cuda-10.0.130 and trained on a workstation equipped with NVIDIA RTX2080Ti GPU, Intel Xeon silver 4214R CPU, and 32 GB RAM. The parameters of the ACNN_DRACNN model are initialized with random numbers satisfying a Gaussian distribution before model training, with the optimizer set to Adam, the learning rate initial value set to 0.001, the batch-size sequentially set to 8, 16, 32, 64, and the epoch set to 100. This model is trained based on the multi-classification cross-entropy cost function as Equation (11), and the cost function curve of four times experiments with different batch sizes is shown in Figure 7.

$$J = -\sum_{i=0}^{C-1} y_i \, log(p_i) \tag{11}$$

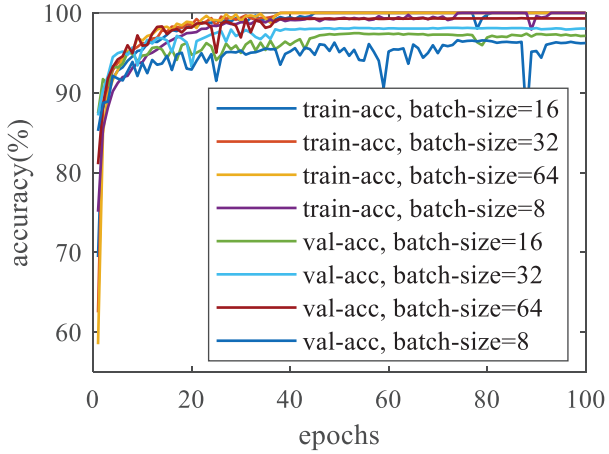


Figure 7. Training curves of ACNN_DRACNN model.

The training and validation curves in Figure 7 show that the ACNN_DRACNN model has good convergence on the ShipsEar dataset as the model iterates, and among four sets of experiments, the best convergence effect is achieved when the batch size is set to 64, at which time the recognition accuracy on the validation set is 99.87%. The confusion matrix of the recognition results is shown in Figure 8. It can be seen that the recognition error mainly stems from the model's confusion between class A and class D targets. This is due to the fact that fishing vessels in class A targets and ro-ro vessels in class D targets have similar mechanical structures and working conditions, and their underwater radiated noise is also similar.

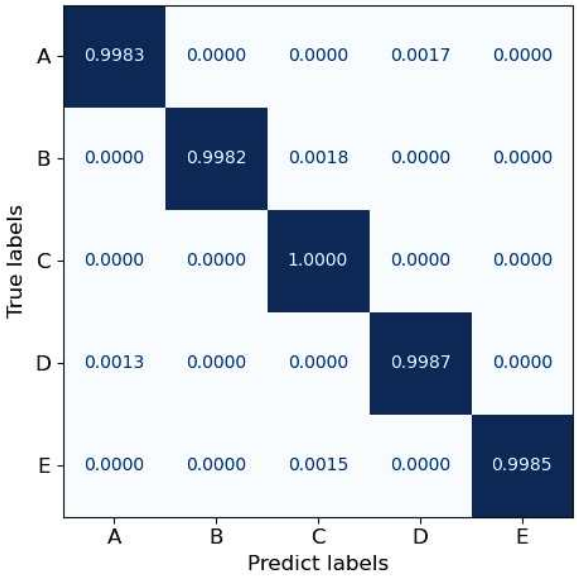


Figure 8. Confusion matrix for recognizing results when batch size is 64.

In the display of identification experiment results, recall, accuracy, precision, and F1-score are adopted to evaluate the recognition performance of the ACNN_DRACNN model. The formulas for each indicator are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F1_score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{15}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. The recognition experiment results are shown in Table 2.

Then, in order to investigate the effect of the gammatone filter number and the global pooling layer of the CA module on the model recognition performance, the ablation experiments are carried out on the test set while other parameters are kept constant, with the number of filters setting to 8, 16, 32, and 64, respectively, and the global pooling setting to GAP, GMP and GEP, respectively. The recognition results are shown in Figures 9 and 10.

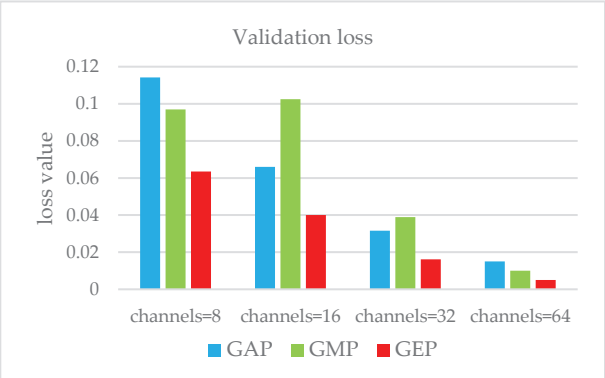


Figure 9. Cost function value of the model on validation dataset.

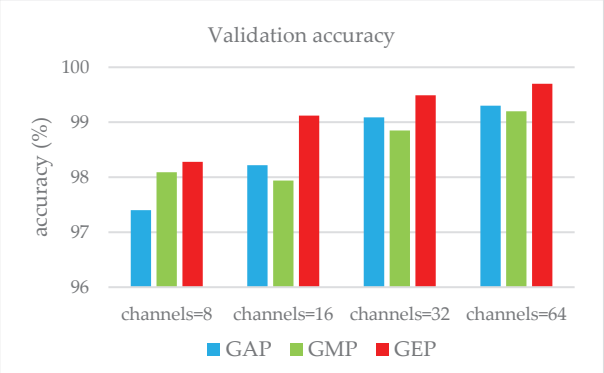


Figure 10. Recognition accuracy of the model on the validation dataset.

With the same number of gammatone filter channels, the ACNN model using the GEP layer proposed in this paper has a higher recognition accuracy and smaller cost function value compared to the model using the GAP layer or the GMP layer, which indicates that the GEP layer improves the model recognition performance and makes it have higher confidence in the recognition results. In addition, as the number of channels increases, the original signal is decomposed more finely, resulting in more explicit line-spectrum components carried by each sub-band signal, so the model recognition accuracy increases and the value of the cost function decreases. It is worth mentioning that the increase in channel number leads to a significant increase in the model parameters and computation, which needs to be weighed against the two factors for channel number selection. Next, the ACNN_DRACNN proposed in this paper is compared with other state-of-the-art models on the same sample set, and the results are shown in Table 3.

Table 3. Recognition performance comparison of this method with other state-of-the-art methods.

No.	Model	Accuracy (%)	Params (M)	Flops (G)
1	DenseNet [29]	90.15	6.96	0.610
2	DarkNet [30]	96.68	40.59	1.930
3	RepVGG [31]	97.05	7.83	0.420
4	CRNN [32]	91.44	3.88	0.110
5	Auto-encoder [20]	93.32	0.18	0.410
6	ResNet [13]	94.97	0.33	0.110
7	A-ResNet [14]	98.19	9.47	1.460
8	MobileNet [8]	94.02	2.23	0.140
9	DRACNN [12]	97.10	0.26	0.005
10	ACNN_DRACNN	99.87	0.61	0.013

Benefitting from the excellent line-spectrum feature extraction capability of the ACNN model, on the ShipsEar dataset, the ACNN_DRACNN model proposed in this paper reaches a target recognition accuracy of 99.8%, which achieves an improvement of 2.7% compared to the DRACNN model proposed by our team last year, and the recognition accuracy is improved to different degrees compared to several other state-of-the-art models. More importantly, the parameters of the ACNN_DRACNN model is only 0.61 M, which is about 1/15th that of the A-ResNet model, and it has only 13 M floating-point computations, which is about 1/100th of the A-ResNet model. In comparison, the model proposed in this paper has fewer parameters and fewer floating-point calculations, which means that less memory and computational resources are required to run the model, facilitating the deployment of the model on a minicomputer system and the fast implementation of target recognition. On our computer, model training process consumes 23 ms per sample, and recognition consumes 2 ms per sample.

The signal-to-noise ratio of the radiated noise signals from underwater acoustic targets in the ShipsEar dataset is high due to the close proximity between the target and the hydrophone. The model recognition performance was tested after adding Gaussian noise with different signal-to-noise ratios to the original data, and the results are shown in Table 4. It can be seen that the target recognition accuracy increases with the increase in SNR. When the SNR is −20 dB, the average recognition accuracy of the model is 66.3%, and when the SNR is greater than −5 dB, the average recognition accuracy of our method achieves more than 85%. In addition, the ability of the model to detect environmental noise is significantly enhanced by adding noise.

Table 4. Recognition accuracy at different signal-to-noise ratios (%).

SNR (dB)	−20	−15	−10	−5	0	5	10
A	63.7	78.5	75.5	84.9	95.1	98.8	99.5
B	65.9	79.6	80.3	85.0	97.4	99.1	99.4
C	68.2	73.7	79.4	86.2	97.1	99.0	99.6
D	52.5	67.0	77.8	84.8	96.9	98.5	99.6
E	81.2	92.1	94.3	97.2	98.4	99.0	99.9
Average	66.3	78.2	81.5	87.6	97.0	98.9	99.6

4.3. Visualization of Features

In the previous section, the auditory neural network model has been trained by samples from the ShipsEar dataset, establishing a data-driven mapping from the original signal to the output layer. In this section, the validation set samples are input into the trained model, and the output layer data is transformed to the frequency domain using the Fourier function. A segment of data in each of the five categories is selected for analysis and comparison, as shown in Figure 11. For the five types of targets, a comparison of the ACNN model output data power spectrum with the input data power spectrum shows that

the energy of the output signals is suppressed or enhanced to different degrees in different frequency bands, and the continuous spectrum intensity of the output signal decreases by about 20 dB compared with that of the input signal, which implies that the line spectra are enhanced to the same degree.

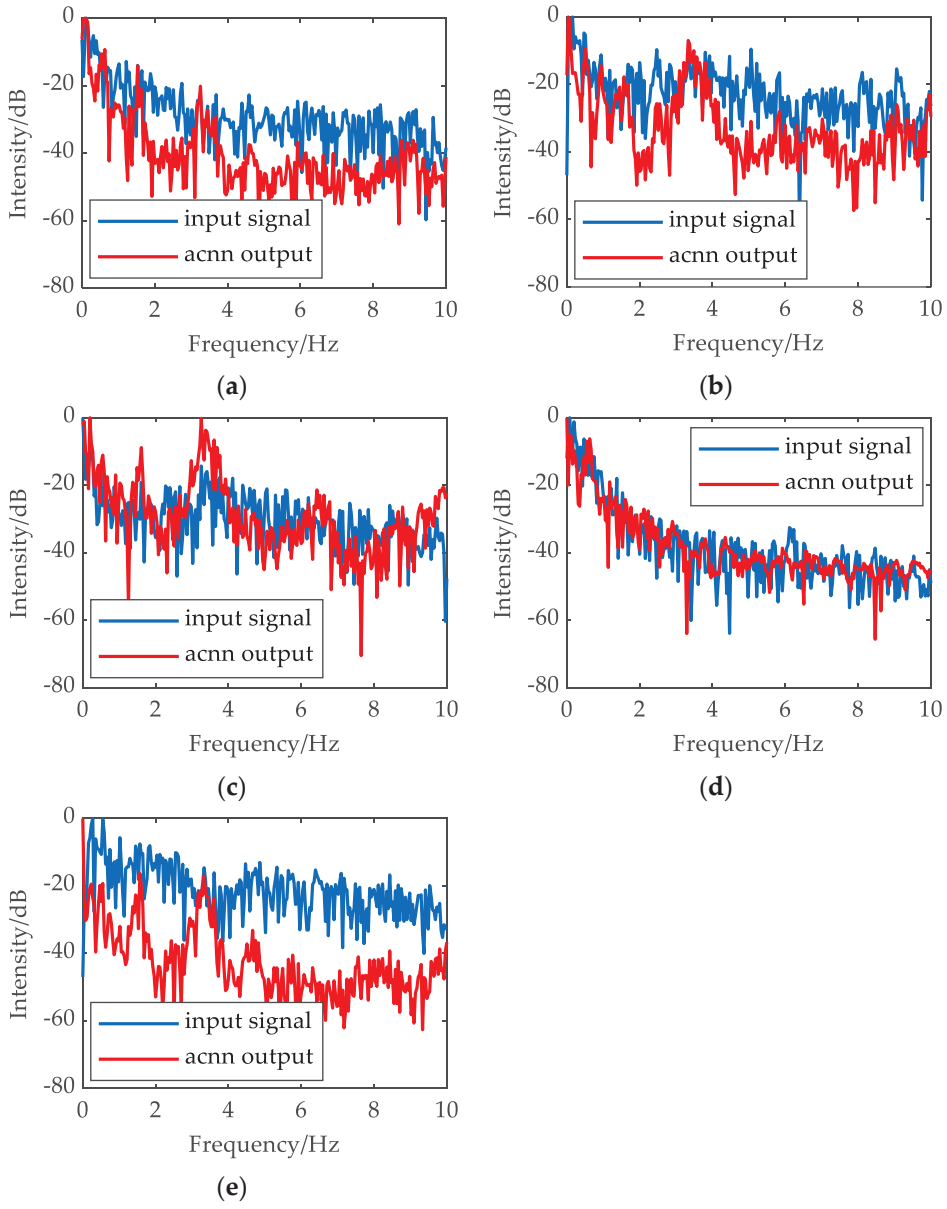


Figure 11. Power spectrum of input data and output data. (**a**) Sample of category A. (**b**) Sample of category B. (**c**) Sample of category C. (**d**) Sample of category D. (**e**) Sample of category E.

Under the constraint of the cross-entropy loss function, the model parameters are continuously updated by an error back-propagation algorithm to extract features from the original signal that can distinguish the five types of targets. It should be emphasized that not all the information in the original signal is valid for identifying the target, so this paper uses an improved channel attention mechanism to select the frequency bands that better reflect the differences between the five types of targets, and therefore the larger the attention weights are, the better they reflect the essential target characteristics. According to Figure 12, the power spectra of the five types of targets are augmented in certain frequency bands, and the common denominator is the presence of line spectra in these bands, which suggests that the main feature relied upon by the model proposed in this paper to recognize

targets is the combination of underwater radiated-noise line spectra. The dimensions of the ACNN model input data and output data are reduced from 4096 to 2 by the TSNE algorithm for visualization in feature space, as feature 12.

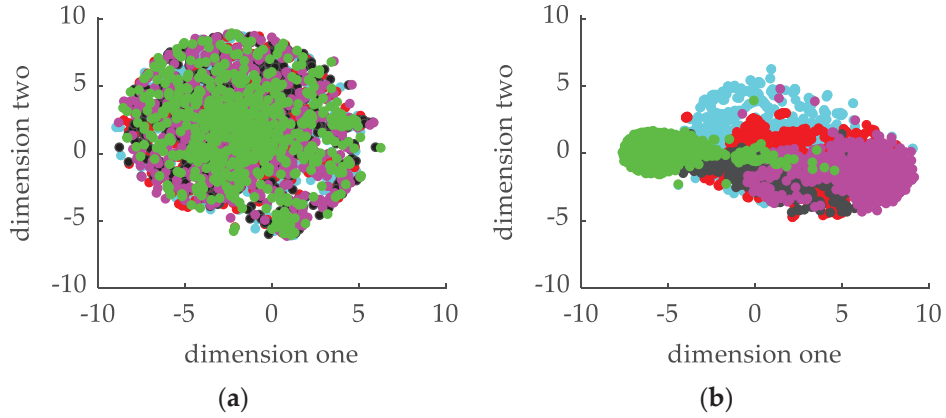


Figure 12. Data visualization by TSNE. (**a**) Raw signals of ShipsEar dataset. (**b**) Output data of the ACNN model.

The raw signals of underwater acoustic targets are disordered in the feature space, and it is almost impossible to find any information that can distinguish the targets from Figure 12a, while from Figure 12b, we can see that the distinguishability of the auditory domain line-spectrum features extracted from the ACNN model output layer is obviously improved, which indicates that the ACNN model proposed in this paper contributes to the improvement of the performance of the UATR and that the features extracted by the model are interpretable.

5. Conclusions

In this paper, an auditory neural network model with the ability of line-spectrum feature extraction of underwater acoustic targets is designed by simulating the human hearing mechanism and combined with the DRACNN model for UATR. On this basis, the main contributions of this article are as follows:

(1) The design idea of the model is inspired by the human hearing mechanism, the feature extraction process is very interpretable, and the data processing results show that the finer the decomposition of the original signal, the higher the target recognition accuracy that can be achieved.
(2) In this paper, a new global pooling method called a GEP layer is proposed, which integrates traditional features with deep learning, and can provide higher recognition accuracy and recognition result confidence for the network model compared with global maximum pooling and global average pooling.
(3) The ACNN_DRACNN model achieves 99.8% recognition accuracy on the ShipsEar dataset, which is a 2.7% improvement over the DRACNN model, and it has better-integrated performance than DarkNet, MobileNet, CRNN and other current state-of-the-art methods.

At the same time, there are still many issues worth studying based on our work, such as:

(1) Expanding the dataset using simulation data or data augmentation, to improve the recognition and generalization ability under sample imbalance and missing data for typical operating conditions.
(2) Mining the time-correlation features and transient working condition features in the original signal to improve the ability of portraying the essential characteristics of underwater acoustic targets.

Author Contributions: Conceptualization, F.J. and J.N.; methodology, F.J.; software, J.N. and S.L.; validation, J.N., F.J. and W.F.; formal analysis, J.N. and S.L.; investigation, W.F.; resources, J.N.; data curation, F.J. and F.J.; writing—original draft preparation, J.N. and S.L.; writing—review and editing, J.N. and F.J.; visualization, S.L., W.F. and J.N.; supervision, F.J.; project administration, F.J.; funding acquisition, F.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant number 52371356.

Data Availability Statement: Data available in a publicly accessible repository. Datasets are openly available at http://atlanttic.uvigo.es/underwaternoise/ at https://doi.org/10.1016/j.apacoust.2016.06.008 (accessed on 15 March 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

Abbreviation	Full Name
UATR	Underwater acoustic target recognition
CNN	Convolutional neural network
GAN	Generative adversarial network
ACNN	Auditory convolutional neural network
DEMON	Detection of envelope modulation on noise
LOFAR	Low frequency analysis and recording
GAP	Global average pooling
GMP	Global maximum pooling
GEP	Global energy pooling
CA	Channel attention
DRACNN	Deep residual attention convolutional neural networks
RACB	Residual attention convolution blocks
ERB	Equivalent rectangular bandwidth

References

1. Luo, X.W.; Chen, L.; Zhou, H.L.; Cao, H.L. A Survey of Underwater Acoustic Target Recognition Methods Based on Machine Learning. *J. Mar. Sci. Eng.* **2023**, *11*, 384. [CrossRef]
2. Jiang, J.G.; Wu, Z.N.; Lu, J.N.; Huang, M.; Xiao, Z.Z. Interpretable features for underwater acoustic target recognition. *Measurement* **2020**, *173*, 108586. [CrossRef]
3. Wang, W.; Zhao, X.C.; Liu, D.L. Design and Optimization of 1D-CNN for Spectrum Recognition of Underwater Targets. *Integr. Ferroelectr.* **2021**, *218*, 164–179. [CrossRef]
4. Kim, K.I.; Pak, M.I.; Chon, B.P.; Ri, C.H. A method for underwater acoustic signal classification using convolutional neural network combined with discrete wavelet transform. *Int. J. Wavelets Multiresolut. Inf. Process.* **2021**, *19*, 2050092. [CrossRef]
5. Yao, Q.H.; Wang, Y.; Yang, Y.X. Underwater Acoustic Target Recognition Based on Data Augmentation and Residual CNN. *Electronics* **2023**, *12*, 1206. [CrossRef]
6. Chen, L.; Luo, X.W.; Zhou, H.L. A ship-radiated noise classification method based on domain knowledge embedding and attention mechanism. *Eng. Appl. Artif. Intell.* **2024**, *127*, 10732. [CrossRef]
7. Ju, Y.; Wei, Z.X.; Li, H.F.; Feng, X. A New Low SNR Underwater Acoustic Signal Classification Method Based on Intrinsic Modal Features Maintaining Dimensionality Reduction. *Pol. Marit. Res.* **2020**, *27*, 187–198. [CrossRef]
8. Yao, H.Y.; Gao, T.; Wang, Y.; Wang, H.Y.; Chen, X. Mobile_ViT: Underwater Acoustic Target Recognition Method Based on Local–Global Feature Fusion. *J. Mar. Sci. Eng.* **2024**, *12*, 589. [CrossRef]
9. Luo, X.W.; Zhang, M.H.; Liu, T.; Huang, M.; Xu, X.G. An Underwater Acoustic Target Recognition Method Based on Spectrograms with Different Resolutions. *J. Mar. Sci. Eng.* **2021**, *9*, 1246. [CrossRef]
10. Ouyang, T.; Zhang, Y.J.; Zhao, H.L.; Cui, Z.W.; Yang, Y.; Xu, Y.J. A multi-color and multistage collaborative network guided by refined transmission prior for underwater image enhancement. *Vis. Comput.* **2024**. [CrossRef]
11. Yildiz, E.; Yuksel, M.E.; Sevgen, S. A Single-Image GAN Model Using Self-Attention Mechanism and DenseNets. *Neurocomputing* **2024**, *596*, 127873. [CrossRef]
12. Ji, F.; Ni, J.S.; Li, G.N.; Liu, L.L.; Wang, Y.Y. Underwater Acoustic Target Recognition Based on Deep Residual Attention Convolutional Neural Network. *J. Mar. Sci. Eng.* **2023**, *11*, 1626. [CrossRef]
13. Hong, F.; Liu, C.W.; Guo, L.J.; Chen, F.; Feng, H.H. Underwater Acoustic Target Recognition with a Residual Network and the Optimized Feature Extraction Method. *Appl. Sci.* **2021**, *11*, 1442. [CrossRef]

14. Li, J.; Wang, B.X.; Cui, X.R.; Li, S.B.; Liu, J.H. Underwater Acoustic Target Recognition Based on Attention Residual Network. *Entropy* **2022**, *24*, 1657. [CrossRef] [PubMed]
15. Zhang, H.Q.; Li, S.; Li, D.H.; Wang, Z.C.; Zhou, Q.X.; You, Q.X. Sonar image quality evaluation using deep neural network. *IET Image Process.* **2022**, *16*, 992–999. [CrossRef]
16. Ashraf, H.; Shah, B.; Soomro, A.M.; Safdar, Q.A.; Halim, Z.; Shah, S.K. Ambient-noise Free Generation of Clean Underwater Ship Engine Audios from Hydrophones using Generative Adversarial Networks. *Comput. Electr. Eng.* **2022**, *100*, 107970.
17. Wang, Z.; Liu, L.W.; Wang, C.Y.; Deng, J.J.; Zhang, K.; Yang, Y.C.; Zhou, J.B. Data Enhancement of Underwater High-Speed Vehicle Echo Signals Based on Improved Generative Adversarial Networks. *Electronics* **2022**, *11*, 2310. [CrossRef]
18. Jin, G.H.; Liu, F.; Wu, H.; Song, Q.Z. Deep Learning-Based Framework for Expansion, Recognition and Classification of Underwater Acoustic Signal. *J. Exp. Theor. Artif. Intell.* **2019**, *32*, 205–218. [CrossRef]
19. Ge, F.X.; Bai, Y.Y.; Li, M.J.; Zhu, G.P.; Yin, J.W. Label distribution-guided transfer learning for underwater source localization. *J. Acoust. Soc. Am.* **2022**, *151*, 4140–4149. [CrossRef]
20. Ji, F.; Li, G.N.; Lu, S.Q.; Ni, J.S. Research on a Feature Enhancement Extraction Method for Underwater Targets Based on Deep Autoencoder Networks. *Appl. Sci.* **2024**, *14*, 1341. [CrossRef]
21. Hao, Y.K.; Wu, X.J.; Wang, H.Y.; He, X.Y.; Hao, C.P.; Wang, Z.R.; Hu, Q. Underwater Reverberation Suppression via Attention and Cepstrum Analysis-Guided Network. *J. Mar. Sci. Eng.* **2023**, *11*, 313. [CrossRef]
22. Li, Y.X.; Gu, Z.Y.; Fan, X.M. Research on Sea State Signal Recognition Based on Beluga Whale Optimization-Slope Entropy and One Dimensional-Convolutional Neural Network. *Sensors* **2024**, *24*, 1680. [CrossRef] [PubMed]
23. Liu, D.L.; Shen, W.H.; Cao, W.J.; Hou, W.M.; Wang, B.Z. Design of Siamese Network for Underwater Target Recognition with Small Sample Size. *Appl. Sci.* **2022**, *12*, 10659. [CrossRef]
24. Li, N.; Wang, L.B.; Ge, M.; Unoki, M.; Li, S.; Dang, J.W. Robust voice activity detection using an auditory-inspired masked modulation encoder based convolutional attention network. *Speech Commun.* **2024**, *157*, 103024. [CrossRef]
25. Li, J.H.; Yang, H.H. The underwater acoustic target timbre perception and recognition based on the auditory inspired deep convolutional neural network. *Appl. Acoust.* **2021**, *182*, 108210. [CrossRef]
26. Yang, H.H.; Li, J.H.; Shen, S.; Xu, G.H. A Deep Convolutional Neural Network Inspired by Auditory Perception for Underwater Acoustic Target Recognition. *Sensors* **2019**, *19*, 1104. [CrossRef]
27. Reiterer, S.; Erb, M.; Grodd, W.; Wildgruber, D. Cerebral Processing of Timbre and Loudness: fMRI Evidence for a Contribution of Broca's Area to Basic Auditory Discrimination. *Brain Imaging Behav.* **2008**, *2*, 1–10. [CrossRef]
28. Occelli, F.; Suied, C.; Pressnitzer, D.; Edeline, J.M.; Gourévitch, B. A Neural Substrate for Rapid Timbre Recognition? Neural and Behavioral Discrimination of Very Brief Acoustic Vowels. *Cereb. Cortex* **2016**, *26*, 2483–2496. [CrossRef]
29. Huang, G.; Liu, Z.; Maaten, L.V.D.; Kilian, Q.W. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [CrossRef]
30. Pathak, D.; Raju, U. Shuffled-Xception-DarkNet-53: A content-based image retrieval model based on deep learning algorithm. *Comput. Electr. Eng.* **2023**, *107*, 108647. [CrossRef]
31. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets Great Again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 13728–13737. [CrossRef]
32. Liu, F.; Shen, T.S.; Luo, Z.L.; Zhao, D.; Guo, S.J. Underwater target recognition using convolutional recurrent neural networks with 3-D Mel-spectrogram and data augmentation. *Appl. Acoust.* **2021**, *178*, 107989. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

AI-Based Prediction of Ultrasonic Vibration-Assisted Milling Performance

Mohamed S. El-Asfoury [1], Mohamed Baraya [1], Eman El Shrief [1], Khaled Abdelgawad [1], Mahmoud Sultan [1] and Ahmed Abass [1,2,*]

[1] Department of Production Engineering and Mechanical Design, Faculty of Engineering, Port Said University, Port Fuad 42526, Egypt; mohamed.saad@eng.psu.edu.eg (M.S.E.-A.); mohamed.baraya@eng.psu.edu.eg (M.B.); eman.ahmed@eng.psu.edu.eg (E.E.S.); k.abdelgawad95@eng.psu.edu.eg (K.A.); mahmoud.sultan@eng.psu.edu.eg (M.S.)

[2] Department of Materials, Design and Manufacturing Engineering, School of Engineering, University of Liverpool, Liverpool L69 3GH, UK

* Correspondence: a.abass@liverpool.ac.uk

Abstract: The current study aims to evaluate the performance of the ultrasonic vibration-assisted milling (USVAM) process when machining two different materials with high deviations in mechanical properties, specifically 7075 aluminium alloy and Ti-6Al-4V titanium alloy. Additionally, this study seeks to develop an AI-based model to predict the process performance based on experimental data for the different workpiece characteristics. In this regard, an ultrasonic vibratory setup was designed to provide vibration oscillations at 28 kHz frequency and 8 μm amplitude in the cutting feed direction for the two characterised materials of 7075 aluminium alloy (150 BHN) and Ti-6Al-4V titanium alloy (350 BHN) workpieces. A series of slotting experiments were conducted using both conventional milling (CM) and USVAM techniques. The axial cutting force and machined slot surface roughness were evaluated for each method. Subsequently, Support Vector Regression (SVR) and artificial neural network (ANN) models were built, tested and compared. AI-based models were developed to analyse the experimental results and predict the process performance for both workpieces. The experiments demonstrated a significant reduction in cutting force by up to 30% and an improvement in surface roughness by approximately four times when using USVAM compared to CM for both materials. Validated by the experimental findings, the ANN model accurately and better predicted the performance metrics with RMSE = 0.11 μm and 0.12 N for Al surface roughness and cutting force. Regarding Ti, surface roughness and cutting force were predicted with RMSE of 0.12 μm and 0.14 N, respectively. The results indicate that USVAM significantly enhances milling performance in terms of a reduced cutting force and improved surface roughness for both 7075 aluminium alloy and Ti-6Al-4V titanium alloy. The ANN model proved to be an effective tool for predicting the outcomes of the USVAM process, offering valuable insights for optimising milling operations across different materials.

Keywords: artificial intelligence; support vector regression; neural network; ultrasonic; vibration assisted cutting; milling; titanium; aluminium

1. Introduction

There is an increasing demand for precise, miniaturised products with tight tolerances and high-quality surface finishes across various industries. Titanium alloys, particularly Ti-6Al-4V, are extensively used in aerospace, automotive and medical sectors due to their exceptional strength, lightweight nature, corrosion resistance and biocompatibility [1–3]. However, their high strength, low thermal conductivity and reactive chemical properties can lead to significant tool wear during machining under thermal stress. In contrast, aluminium alloys such as Al7075 are widely employed in aerospace, automotive, machinery and medical fields. While Al7075 is valued for its superior strength and corrosion

resistance [4,5], it presents machinability challenges, including surface roughness and geometrical accuracy compared to other aluminium alloys, complicating the manufacturing process [6,7]. Ultrasonic vibration-assisted milling (USVAM) is an advanced machining technique that combines conventional milling (CM) with ultrasonic vibrations, significantly improving surface quality [8,9]. The ultrasonic-assisted milling (USAM) process builds on this by enhancing machining performance in several critical areas. It improves surface accuracy, reduces surface roughness, minimises wear and thermal stress and extends tool life. These advancements lead to better quality finishes and reduced operational costs, making USVAM a valuable technique in advanced manufacturing [10,11]. As a result, there has been increased interest in developing and optimising VAM techniques to further enhance machining efficiency and process parameters.

Hu et al. [12] carried out experiments comparing ultrasonic-assisted milling (UAM) to CM of titanium alloy. Their findings showed that UAM greatly reduced chip adhesion during cutting, which led to less burr formation and improved surface integrity. Additionally, UAM achieved a surface roughness of 96 nm, a 46.7% improvement over the 180 nm roughness observed with CM [12]. Liu et al. [13] compared CM and UVM regarding their effects on chip formation and machining quality. Their tests demonstrated that the UVM improves shear slip and chip evacuation, resulting in a smaller chip curl angle. Engelking et al. [14] optimised surface residual stress in additively manufactured Ni alloy components using UAM. Their results indicated that CM primarily induced tensile residual stress, while UAM generated compressive residual stress on the machined surface. Furthermore, their study showed that UAM significantly improved Ni alloy components' processability and surface integrity, especially at lower cutting speeds. Fang et al. [15] investigated the impact of conventional micro-milling and one-dimensional ultrasonic-assisted vibration micro-milling (UVAMM) on the surface quality of workpieces. Their findings indicated that UVAMM reduced the cutting forces in the three principal directions, F_x, F_y and F_z, by 7.6%, 11.5% and 1.3%, respectively. Additionally, when the per tooth feed rate (f_z) was set to 5 μm, the burr-inhibition effect was most pronounced, with the minimum burr sizes measuring 50.23 μm on the down-milling side and 36.57 μm on the up-milling side. Zhang et al. [10] investigated the impacts of CM and UAM on titanium alloy's surface and subsurface characteristics. They discovered that UAM produced a consistent micro-texture on the surface but resulted in higher surface roughness compared to CM. Additionally, UAM was found to intensify changes in the subsurface structure and cause greater plastic deformation. Xu et al. [16] applied longitudinal-torsional ultrasonic vibration milling (LTUVM) to address subsurface damage and severe burr formation in the production of CFRP materials. They developed the first numerical model for cutting aramid fibre-reinforced polymer (AFRP). Their tests demonstrated that LTUVM reduced burr length by 23–38%. The most significant improvement in burr length, up to 38%, occurred at a cutting speed of 4000 rpm. Additionally, at a speed of 2800 rpm, the cutting force reduction rate was approximately 48%.

Wang et al. [17] analysed the impact of ultrasound-assisted dry spiral milling processing parameters on CFRP delamination damage. Their experimental results revealed that the influence of ultrasound amplitude was relatively minor, accounting for only 18.1% of the overall effect. Amini and Bayat [18] investigated the impact of axial ultrasonic-assisted milling (UAM) on the distortion of aluminium alloy parts. Their study compared the effects of CM and UAM on distortion and milling force. The experiments demonstrated that UAM significantly reduces milling force and distortion compared to CM, showing a 29% reduction in cutting force and a 21% reduction in distortion. This improvement has crucial implications for enhancing the flatness tolerance of workpieces, resulting in components free from waves and warpages. El-Hofy and Ali [19] studied the effect of vibration-assisted helical milling (VAHM) on machining 7075 aluminium alloy. Their experiments showed that applying axial ultrasonic vibrations improved the surface finishing performance. The optimal conditions were a rotational speed of 5000 rpm, a tangential feed of 100 mm/min and a helical pitch of 0.1 mm/rev. Under these conditions, VAHM

reduced axial cutting forces and diameter deviation, enhancing overall milling performance. Haddad et al. [20] studied USVAM on AISI 1045 steel and 7075 aluminium, using resonant frequencies of 19,840 and 19,757 Hz, respectively, and increasing ultrasonic intensity from 15 to 22.5 W/cm^2 improved surface roughness (Rz) by 70% and smoothness (Ra) by 72% for aluminium. For steel, 22.5 W/cm^2 vibrations enhanced surface smoothness by 53% (average) and 45% (maximum), while 30 W/cm^2 vibrations increased smoothness by 35% and 25% on face-milled surfaces. Baraya et al. [21] developed a vibratory system for slotting using ultrasonic vibrations at 34.7 kHz and 10 µm amplitude on Al 7075 alloy. Their experiments showed up to a 50% reduction in cutting force and improved surface roughness with ultrasonic vibrations, which also reduced slot width error compared to conventional milling. Their developed statistical model accurately predicted process performance, confirming the technique's effectiveness for high-precision manufacturing.

The implementation of artificial intelligence (AI) based machine learning (ML) algorithms in milling processes has significantly advanced the monitoring and prediction of factors like tool wear, failure and process optimisation—areas where conventional methods often fall short [22]. Commonly used ML algorithms include Support Vector Machine (SVM) [23], artificial neural networks (ANN) [24] and Decision Trees, with specific subtypes like Probabilistic Neural Networks (PNN) [25] and Random Forest (RF) also being prevalent [26]. These algorithms have been particularly effective in tool condition monitoring, process parameter optimisation and enhancing product quality by predicting surface roughness, cutting force and workpiece deformation [22]. In the field of ultrasonic-assisted milling, which involves costly tools and time-consuming experiments, AI methods, particularly ANNs, offer a practical alternative for predicting the effects of various parameters on cutting processes. ANNs have a strong track record in manufacturing for tasks like process optimisation and tool condition monitoring, making them a reliable choice for modelling ultrasonic-assisted milling (UAM) processes [27]. The ANN model has demonstrated high accuracy in predicting surface roughness for CNC face milling components, and its application in the design of experiments for various machining operations further validates its effectiveness [28]. Concerning the use of AI and machine learning for the prediction and evaluation of milling process performance, Zerti et al. [29] analysed the effectiveness of response surface methodology (RSM) versus artificial neural networks (ANN) in predicting outcomes. Their findings revealed that the ANN model achieved a prediction accuracy of 99.9% for surface roughness and cutting forces.

In contrast, the RSM model provided an accuracy of 87.31% for surface roughness and 98.03% for cutting force values. Karabulut [30] investigated the optimisation of cutting forces and surface roughness in AA7039/Al_2O_3 metal matrix composites milling using an ANN model. The study's results indicated that the ANN model was highly effective, achieving prediction accuracies of 97.75% for surface roughness and 93.34% for cutting forces. Pourmostaghimi et al. [31] devised an advanced methodology utilising ANN to estimate machining parameters in the turning process of AISI D2. Their findings demonstrated that this approach could predict machining time with high accuracy.

This study explores the performance analysis and predictive capabilities of artificial neural networks (ANN) in ultrasonic-assisted milling (UAM) across various material properties. This research addresses the limitations of traditional milling processes by aiming to improve machining precision. An innovative USVAM system was designed to introduce ultrasonic vibrations to the workpiece along the feed direction. Subsequently, a statistical model was developed to predict the responses of the USVAM process, which was validated through additional experimental trials. The resulting ANN model is a tool for effectively controlling the USVAM process, considering the applied vibrations and machining parameters. This advancement demonstrates significant applicability in manufacturing industries that demand precision and high accuracy.

2. Methods

2.1. FEA Model Construction and Ultrasonic Vibratory Transducer Design

Finite Element Analysis (FEA) is pivotal in accurately simulating the behaviour of complex manufacturing systems [32,33]. Its ability to model intricate geometries and material properties is advantageous for analysing ultrasonic vibratory transducers. FEA allows for detailed simulations of complex vibratory behaviours, enabling the optimisation of transducer performance before building the physical parts without extensive physical prototyping. This approach helps identify potential issues and refine designs early in development, reducing costs and time. Additionally, FEA can accurately predict the effects of different materials and geometries, ensuring the transducer meets specific performance criteria [34]. The FEA's role in this study was to verify and refine the proposed design. Following this, the components of the finalised design were manufactured. Additionally, FEA was employed to assess the nodal plan within the workpiece. This position was crucial for workpiece fixation during milling, as the wrong fixing position degenerates the vibration waves.

The initial step in developing this study's FEA model involved creating the transducer components' geometry. The dimensions for each component were determined according to the methodology outlined by Baraya et al. [35]. The ABAQUS 13.1 software package (Dassault Systèmes, Vélizy-Villacoublay, France) was employed to construct the model geometry, as illustrated in Figure 1. Notably, the innovative approach of using FEA allows for precise analysis of the piezoelectric transducer components (front mass, piezoelectric rings, electrodes, back mass and steel bolt), depicted in Figure 1a, which are assembled and held together through a compressive pre-load using a steel bolt.

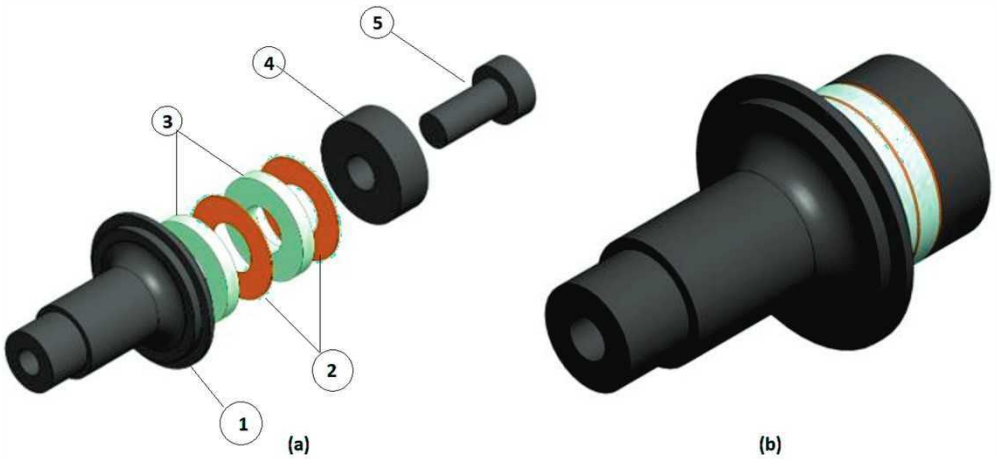


Figure 1. (**a**) Transducer components: (1) front mass, (2) piezoelectric rings, (3) electrodes, (4) back mass, (5) steel bolt and (**b**) Assembled transducer.

Following the literature [35], appropriate materials were designated for each of the transducer parts. Aluminium was selected for both the front and back masses due to its lightweight and strength properties. Copper was chosen for the electrodes because of its excellent electrical conductivity. Piezoelectric materials were used for the rings to efficiently transform electrical energy into mechanical energy. Tables 1 and 2 were used to define the characteristics of these materials in the FEA model. Subsequently, loads and boundary conditions were applied. A pre-stressed bolt load, calculated based on the recommended compression state of the piezoelectric stack (approximately 30 MPa), was applied between the back mass and the piezoelectric stack at the bolt cross-section [35]. Additionally, the boundary conditions (BC) are applied to the piezoelectric elements, where the electrical potential starts from the initial state at the surfaces between the piezoelectric rings themselves and between the piezoelectric rings and the neighbouring metal masses [36,37]. A depiction of these load and boundary conditions can be observed in Figure 2a.

Table 1. Properties of the piezoelectric elements.

Item Code	BJUPR
Dimensions (mm)	35 × 15 × 5
Density (Kg/m^3)	7.7
capacity (p Farad)	1720
Piezoelectricity (10^{-12} Coulomb/Newton)	D_{33} = 240, d_{31} = −95, d_{15} = 380
Dielectricity ($\varepsilon_0 = 8.85 \times 10^{-12}$ Farad/meter)	${\varepsilon_{33}}^S/\varepsilon_0$ = 540, ${\varepsilon_{11}}^S/\varepsilon_0$ = 800
Elastic compliance (10^{-12} meter^2/Newton)	${S_{11}}^E$ = 11.4, ${S_{33}}^E$ = 13.7

Table 2. Physical properties of transducer component materials.

No.	Material	Young's Modulus (GPa)	Density (Kg/m^3)	Poisson's Ratio	Wave Velocity (m/s)	Characteristic Acoustic Impedance (10^6·Ns/m^3)
1	Aluminium (5083)	70.3	2660	0.33	5140	13.67
2	Steel (AISI 1045)	200	7870	0.3	5040	39.7
3	Piezoelectric	73	7700	-	3080	23.72
4	Copper	115	8900	0.31	3595	31.9

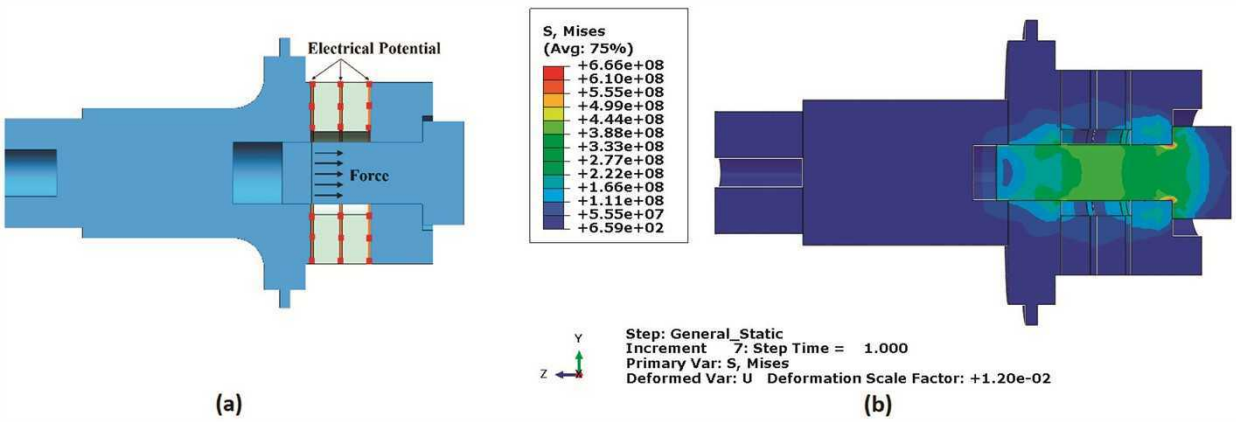


Figure 2. (**a**) Load and boundary conditions and (**b**) von Mises equivalent stress results from finite element static analysis for the transducer.

The FEA mesh design for ultrasonic transducers involved careful selection of mesh elements to precisely capture each component's physical behaviour. All metallic parts were meshed using 3D stress quadratic elements (C3D20R) to ensure detailed stress distribution and deformation characteristics. Piezoelectric quadratic elements (C3D20RE) were chosen for the piezoelectric rings. This selection allows for accurately simulating the electromechanical coupling effects inherent in piezoelectric materials.

2.2. FEA Model Outcome

The FEA results, depicted in Figure 2b, verified that the applied bolt torque does not induce high-stress levels that could fail transducer components. This verification ensures the structural integrity and reliability of the transducer under operational conditions. Therefore, the design is robust against the applied torque, safeguarding the transducer's performance.

The choice of C3D20R elements in Abaqus for metallic parts was successful, as 20-node quadratic brick elements accurately capture curved geometries and stress gradients. Their reduced integration helps avoid common numerical issues, making them more efficient and less computationally intensive. C3D20R elements also perform well in nonlinear

analyses, efficiently handling large deformations and complex material behaviours. Their higher-order interpolation improves convergence, meaning fewer iterations for the solver and quicker results. The piezoelectric ring C3D20RE elements enhanced hourglass control and improved numerical stability and accuracy. They excel in nonlinear analyses, capturing deformations and dynamic behaviours essential for optimising transducer performance.

Given that the design frequency was 28 kHz, the frequency analysis was conducted over a 26–30 kHz frequency range. Figure 3 depicts the transducer operating in the longitudinal vibration mode, wherein each particle's displacement occurs in the axial direction. The measured resonance frequency, at which the transducer naturally oscillates, was 25,866 Hz, which deviates by approximately 7.62% from the design frequency, thus confirming the accuracy of the design. Figure 3a,b present the von Mises stress and displacement distributions at longitudinal mode shape, respectively. These figures illustrate the variations in displacement and stress, highlighting regions of maximum intensity. Notably, the maximum stress was observed at the nodal point (Figure 3a), which was critical for the transducer's performance during experiments. In contrast, the maximum displacement was recorded at the front-end face of the transducer, Figure 3a,b.

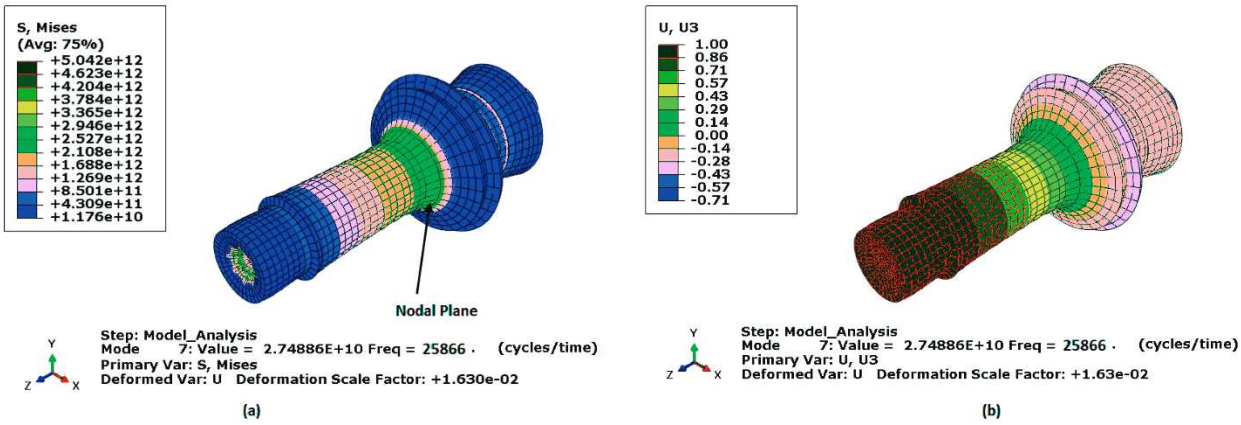


Figure 3. Longitudinal mode shape of the transducer, (**a**) normalised displacement, (**b**) and von Mises stress.

Figure 4a,b displays the von Mises stress and displacement distributions along the transducer's axial direction for the longitudinal mode shape after incorporating the horn and workpiece. The resonance frequency for this configuration was measured at 26.387 kHz, which deviates by approximately 5.76% from the design frequency for the Al7075 workpiece, thus further validating the accuracy of the design. Additionally, the resonance frequency for the Ti-6Al-4V titanium workpiece was measured at 26.328 kHz, which deviates by approximately 6% from the design frequency. The frequency error ratio of a transducer refers to the difference between the designed resonant frequency and the actual resonant frequency during operation. This ratio ensures the transducer performs as intended, especially in applications requiring precise ultrasonic frequencies. The error was due to practical constraints and operational tolerances in this case. Cost considerations and empirical data supporting acceptable performance at this error level validate its use.

A secure connection was essential for effectively transmitting stress waves from the transducer to the workpiece through the ultrasonic vibration-assisted milling device. Figure 5a depicts the longitudinal path, including the workpiece and device, as analysed through FEA methods to evaluate the mechanical behaviour at the horn-workpiece fixation point. Stress waves are continuously transmitted from the horn to the workpiece, facilitating the identification of the optimal fixation point with maximum stress and minimal displacement, as illustrated in Figure 5a,b. The propagation speed c and wavelength λ of the stress waves, which are influenced by material properties [35], guide this identification.

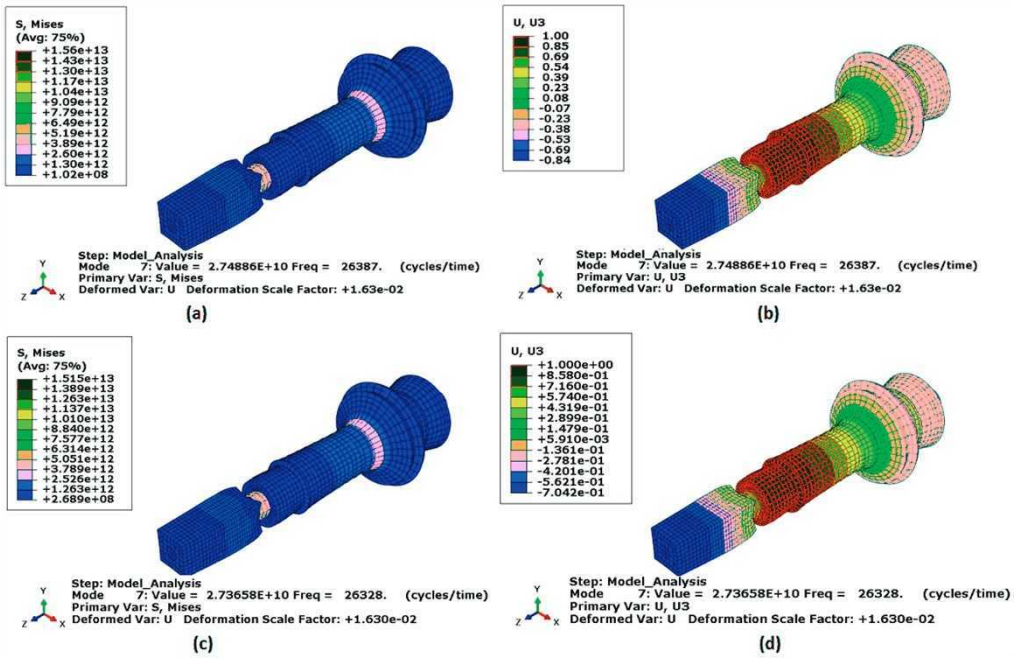


Figure 4. Longitudinal mode shape of the workpiece under the influence of an ultrasonic vibration-assisted milling device. (**a**,**b**) von Mises stress and normalised displacement, respectively, for the Al 7075 alloy workpiece. (**c**,**d**) von Mises stress and normalised displacement, respectively, for the Ti-6Al-4V titanium workpiece.

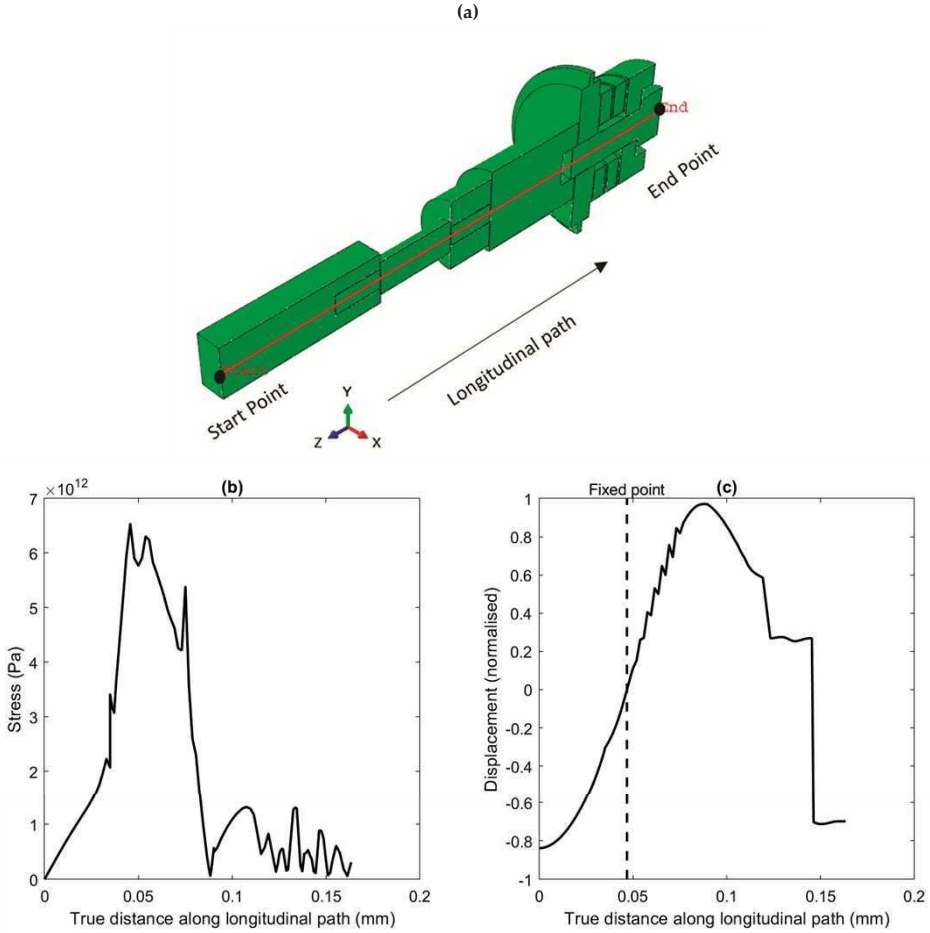


Figure 5. *Cont.*

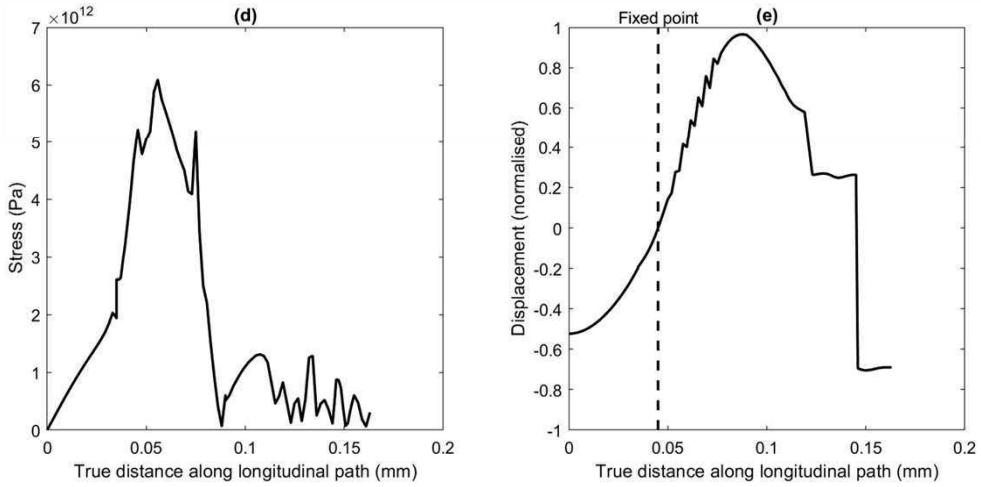


Figure 5. (**a**) A longitudinal path through the workpiece and the ultrasonic vibration-assisted milling device, (**b**) von Mises stress distribution along the path (Al 7075 workpiece), (**c**) normalised displacement distribution along the path (Al 7075 workpiece), (**d**) von Mises stress distribution along the path (Ti-6Al-4V workpiece) and (**e**) normalised displacement distribution along the path (Ti-6Al-4V workpiece).

2.3. FEA Model Validation

The experimental measurements of the vibration amplitude and the validation of the FEA results by checking resonance frequency and nodal plan position were performed according to a technique proposed by Baraya et al. [35]. The transducer components, including the stepped horn, were manufactured and assembled for experimental validation. The workpiece was attached to the transducer through the designed connector. Then, the transducer was connected to the ultrasonic generator, and iron powder was distributed along the workpiece, as shown in Figure 6a.



(**a**) (**b**)

Figure 6. (**a**) Iron powder distributed along the workpiece in the initial position and (**b**) iron powder concentrated at the nodal plane after applying vibration.

The nodal plane location was validated using the iron powder, as depicted in Figure 5c,e. When the ultrasonic generator started to send the vibration waves to the transducer, the iron powder moved along the workpiece, responding to the vibration waves except at the nodal position, as shown in Figure 6b. This experimental observation confirmed that the nodal plane location corresponded closely with the predictions from the FEA.

3. Experimental Setup

In the current study, two workpiece materials were used: Al 7075, measuring 15 mm × 50 mm × 12 mm and Ti-6Al-4V titanium alloy, measuring 15 mm × 50 mm × 10 mm.

These materials were subjected to CM and USVAM. Two different cutting tools were used: (S822) a 4 mm diameter carbide endmill with two flutes was used for Al 7075 workpieces, while (S260) a 4 mm diameter carbide endmill with four flutes was used for Ti-6Al-4V workpieces (Dormer Pramet, Sumperk, Olomoucky kraj, Czech Republic). The chemical and mechanical properties of the workpieces are listed in Tables 3 and 4, respectively.

Table 3. Chemical composition (wt.%) of Al-7075 and Ti-6Al-4V [38,39].

Al-7075									
Element	**Zn**	**Mg**	**Cu**	**Si**	**Mn**	**Cr**	**Fe**	**Ti**	**Al**
wt.%	3.25	1.9	1.8	0.5	0.4	0. 2	0.5	0.15	Balanced
Ti-6Al-4V									
Element	**Al**	**V**	**Fe**	**C**	**N**	**H**	**O**	**Ti**	
wt.%	5.5~6.8	3.5~4.5	≤0.30	≤0.08	≤0.05	≤0.015	≤0.20	Balanced	

Table 4. Mechanical properties of Al-7075 and Ti-6Al-4V [39–42].

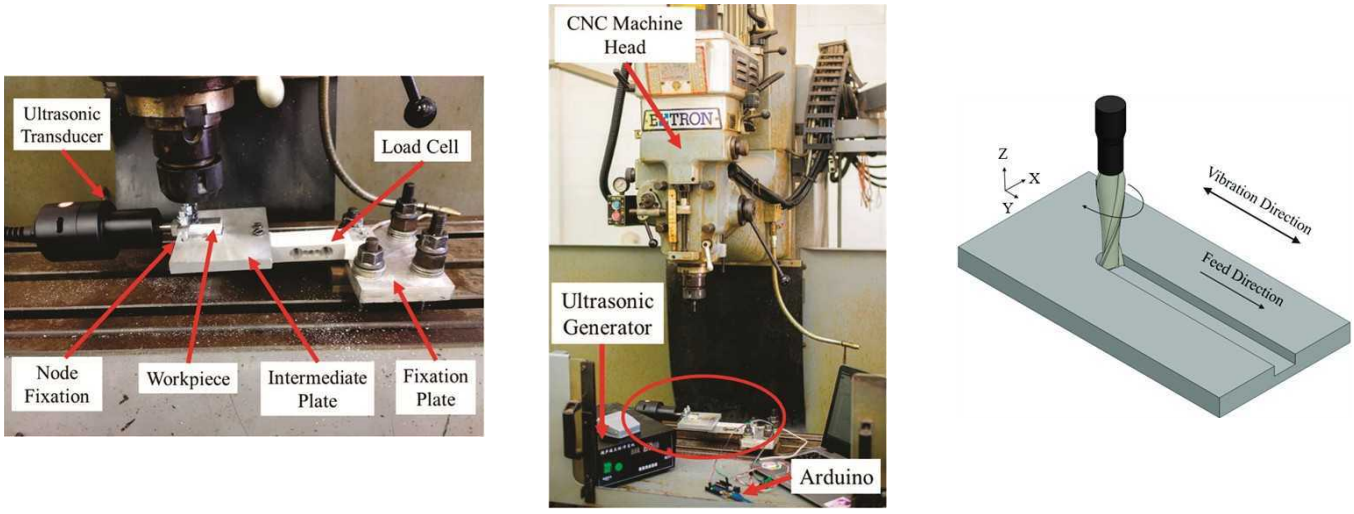
Property	**Elongation (%)**	**Hardness (HV)**	**Tensile Strength (MPa)**	**Elastic Modulus (GPa)**	**Yield Strength (MPa)**	**Poisson Ratio**
Al-7075	11	175	570	72	505	0.33
Ti-6Al-4V	14.5	350	1150	125	950	0.342

The experimental work was conducted using an Extron Vertical Milling Centre machine (CENTROID, Howard, PA 16841 USA). The vibration generation system was comprised of an ultrasonic generator unit with maximum power rating of 1 kW operating at a frequency of 28 kHz, a piezoelectric transducer and a booster to amplify the vibration amplitude. Vibration waves were transmitted to the workpiece through this setup during the cutting process.

During the machining experiments, the axial cutting force was measured using a Wagezellen load cell model 102, measuring forces up to a capacity of 500 N. The mechanical loads were converted into an electrical charge by a piezoelectric sensor. This electrical charge was processed and recorded as force units by a programmed Arduino Uno (Somerville, Somerville, MA, US), as shown in Figure 7b. Arduino Uno interfaced with the sensor and converted analogue signals into digital formats for accurate force measurement and data recording. The surface roughness of the machined slots was assessed in order to compare the slotting process performance in CM and USM. During this process, the average roughness (Ra) and average maximum height of the profile (Rz) values were measured using a Surface Roughness Tester TR-200 (Bowers Group, Albany Court, Albany Park, Camberley, UK).

In this context, the Ra surface roughness unit was micrometres (μm), where Ra stands for "arithmetic average roughness" and measures the average deviation of the surface profile from the mean line over a specified length. Rz in micrometres represented the average maximum height of the profile and was calculated by taking the average of the vertical distance between the highest peak and the lowest valley in each sampling length and then averaging these values.

For each slot, three measurements were taken along a 4 mm segment of the slot length to ensure accuracy and consistency in the roughness evaluation. This method provided a detailed comparison of the surface finish quality achieved by each milling technique.



(**a**) (**b**) (**c**)

Figure 7. (**a**) Vibration system equipment, (**b**) complete setup with force-measurement system and ultrasonic components and (**c**) vibration direction during the milling process.

3.1. Experimental Design Methodologies

The experimental study aimed to optimise the milling process by investigating various input parameters. The input parameters examined were the depth of cut (mm), feed motion (mm/min) and spindle speed (rpm). Preliminary experiments were conducted to establish suitable upper and lower levels for these parameters. Based on the results of these preliminary tests and data from the literature, the parameter levels for the comprehensive set of experiments were determined. The total number of tests conducted was 18 experiments with 3 replications for each workpiece. The selected milling parameters and their levels are listed in Table 5. All experiments were performed under dry-cutting conditions.

Table 5. Experimental parameters for Al 7075 and Ti-6Al-4V titanium alloy workpieces.

Levels	Machining Condition			
	Depth of Cut (mm)	Feed (mm/min)	Spindle Speed (rpm)	Ultrasonic Vibration
1	0.05	10	1000	ON/OFF
2	0.1	20	2000	ON/OFF
3	0.2	30	3000	ON/OFF

3.2. SVR Model Construction

Support Vector Regression (SVR) is a Support Vector Machine (SVM) tailored to predict continuous values rather than classify data. While using SVR, the goal was to find a function that predicts surface roughness and cutting force from depth of cut, cutting and feed speeds with controlled assisted vibration within the smallest margin of error while keeping the model simple. As the most effective way to compare the performance of different AI-based algorithms is to cross-validate them, SVR was initially tested in the current study following a random 70%, 15% and 15% data partition strategy for training, validation and testing. Fifty-four data points were used in building the SVR model for each material (Al and Ti). It is crucial in this context to refer to the randomisation selection used for the training, validation and testing data as "pseudo-random selection" or "deterministic selection", as the seed was fixed to a single value of 42 that controlled the random value generator. This value was selected after running that model with multiple seeds and identifying the seed that gave the best average accuracy across all AI models. A fixed seed approach was used

across the study to ensure reproducibility and comparability among the results, which means that while the selection mimics randomness, the results can be exactly replicated in subsequent runs.

The SVR model was built and trained to predict the output variables, employing a Gaussian kernel using the MATLAB Deep Learning Toolbox "fitrsvm" function. The Gaussian function was selected as a kernel for its capability to capture nonlinear relationships between inputs and outputs. The kernel scale was set to "auto", which automatically adjusted the kernel function's influence by its width (3.0932 in this case, with 26 iterations), and the input values were standardised before training, ensuring that each value contributes equally to the prediction process. After being trained, the SVR model predicted the two outputs. The accuracy of these predictions was evaluated using two key metrics: Root Mean Square Error (RMSE) and Mean Squared Error (MSE). The RMSE provided an average measure of the prediction error magnitude. At the same time, the MSE offered insight into the model precision by averaging the squared differences between the predicted and true values.

3.3. ANN Model Construction

This section describes the implementation of an ANN in MATLAB Deep Learning Toolbox for predicting target variables (surface roughness and cutting force) from input features (depth of cut, cutting and feed speeds in addition to on/off assisted vibration). Once more, a subtotal of 54 data points was used for each material, with 108 total data points.

The process started with data preparation, where the input features and target variables were rearranged to match the required dimensions for the ANN and stored as ANN inputs and outputs. Then, a feedforward ANN with 10 neurons in its 2 hidden layers was created using the "fitnet" function. This setup was chosen to balance network complexity and performance. The training parameters procedure was achieved by setting the network configuration for 100 epochs with a learning rate of 0.01, optimising the training process using the Levenberg–Marquardt algorithm handled by the "train" function. The activation function used was the sigmoid function "tansig" in the hidden layers and a linear function "purelin" in the output layer.

The number of hidden layers was set by starting with a single hidden layer and progressively increasing the number of hidden layers to five while evaluating the performance through the overall sum of Root Mean Squared error (RMSE) values. The outcome of this process indicated that two hidden layers were the best choice, as the lowest overall RMSE was recorded with them. The overfitting error was observed when more layers were tested (Figure 8); hence, the ANN model was structured as in Figure 9.

The dataset was randomly divided into 70% for training, 15% for validation and 15% for testing, ensuring a good mix and preventing overfitting. The network was trained using the prepared inputs and outputs, adjusting weights to minimise prediction errors. After training, the network was used to predict outputs based on the input data. In this case, prediction errors for the first and second output components, which were surface roughness and cutting force, were calculated. Finally, the RMSE was computed for each to evaluate network performance. RMSE was used to assess performance as it measures the average prediction error magnitude, reflecting variance and bias. RMSE is sensitive to large errors, making it an effective metric for identifying weighty prediction deviations. Additionally, its straightforward interpretation as the average deviation from actual values facilitates comparison across different models.

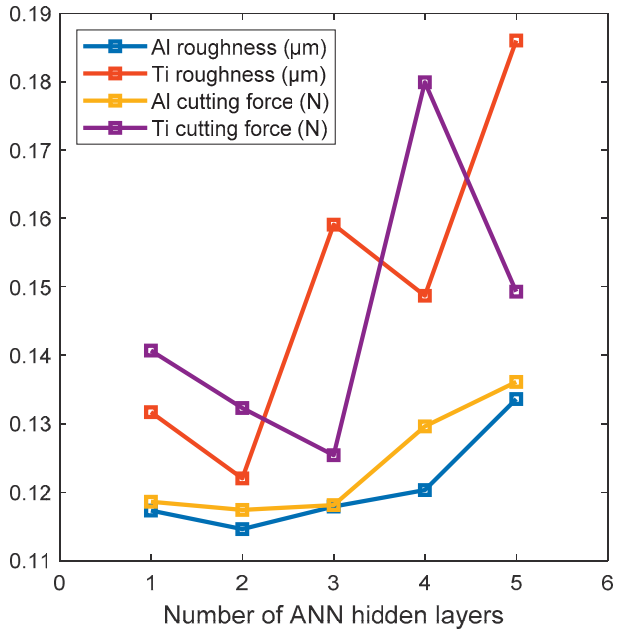


Figure 8. The process for selecting the number of hidden ANN layers was based on the overall RMSE error minimisation, where two layers were the best.

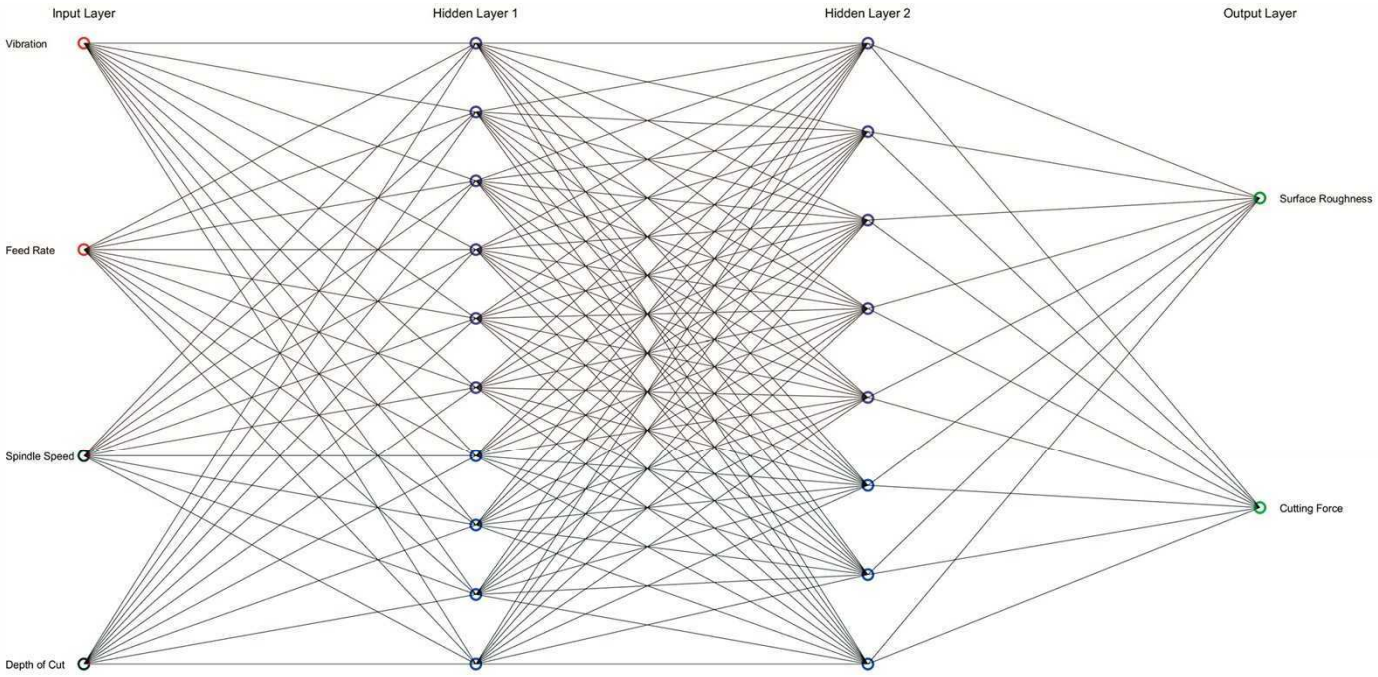


Figure 9. A simplified representation of the architecture of the ANN was built within the current study. Input neurons are coloured red, output neurons are green, and hidden layer neurons are blue. For simplicity, this figure does not display weights, bias or the activation function.

4. Results and Discussion

This section presents the outcomes of the slotting experiments conducted on the 7075 aluminium alloy and Ti-6Al-4V titanium alloy using CM and USVAM techniques. The focus was on the axial cutting force and the surface roughness of the slot button area.

4.1. Axial Cutting Force

The axial cutting force data collected from the experiments reveal intriguing insights into the influence of USVAM on machining performance. Figure 10 summarises the average cutting forces recorded for the two investigated materials and milling techniques.

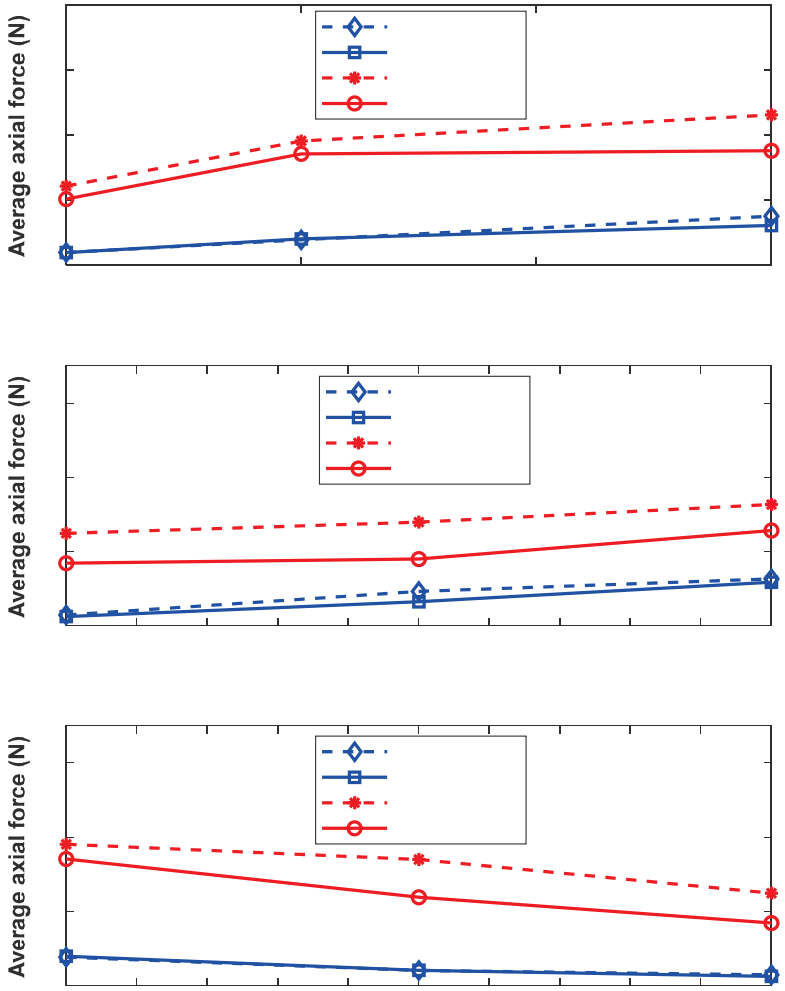


Figure 10. Average axial milling force verses (**a**) DoC (N = 1000 rpm, f = 10 mm/min), (**b**) cutting feed (N = 3000 rpm, DoC = 0.1 mm) and (**c**) cutting speed (DoC = 0.1 mm, f = 10 mm/min).

The results indicate a noticeable reduction in the axial force when US vibration is introduced for the Ti-6Al-4V alloy. Remarkably, no significant reduction was found for the Al 7075 alloy under any conditions. This phenomenon is likely attributed to the highly plastic nature of Al 7075 alloy, causing a large amount of vibration to be damped by the workpiece itself [43]. Hence, the interrupted cutting mechanism is not employed as expected. Increasing depth of cut (DoC) and cutting feed rates led to a rising trend in the axial force (see Figure 10a,b) as higher DoC and cutting feed causes an increase in the contact area between tool/workpiece interface, causing higher exerted force even when applying USM. Additionally, for the Ti-6Al-4V alloy, it could be seen that the axial force was reduced significantly by around 25% (at N = 1000 rpm, f = 10 mm/min and DoC = 0.2 mm) and by slightly more than one-third (at N = 3000 rpm, f = 20 mm/min and DoC = 0.1 mm). Figure 10c depicts a declining behaviour as the cutting speed increases; this occurs as higher cutting speed causes higher linear cutting velocity and hence higher shear rate as well as heat generation, which causes a thermal softening for the material [44]

and decreases its shear strength, leading to a lower exerted force that USM can even reduce by nearly one third (at N = 2000 rpm, f = 10 mm/min and DoC = 0.1 mm) for the Ti-6Al-4V alloy.

4.2. Surface Roughness

Surface roughness is a critical factor in determining product quality. The average surface roughness values measured for the slot bottom area of both materials and both milling techniques are summarised in Figure 11. Generally, applying US vibration led to an improvement in the surface roughness for both investigated materials. Figure 11a shows a rising trend of surface roughness as DoC increases either with or without US vibration; additionally, it could be seen that the US effect is higher as DoC increases to reach around 50% and 80% surface roughness reduction (at N = 1000 rpm, f = 10 mm/min and DoC = 0.2 mm) for the titanium and aluminium alloys, respectively. On the other side, higher cutting feed reduces the machining time and the friction time between the tool and the workpiece, leading to a better surface quality. According to Figure 11b, introducing vibrations improves the surface roughness at a lower cutting feed rate. However, this impact is decreased by increasing the cutting feed. The reason for this is that the application of USM at high levels of cutting feed does not lead to the interrupted cutting mechanism rather than additional applied cutting forces due to an extra contact between the cutting tool and the workpiece resulting from the application of vibration [45] (see the rising trend in Figure 4b). Similarly, based on Figure 11c, the reduction in surface roughness was significant for both materials as it reached a peak value for Ti-6Al-4V by nearly 100% (at N = 1000 rpm, f = 10 mm/min and DoC = 0.1 mm) and for Al 7075 by approximately 4 times (at N = 3000 rpm, f = 10 mm/min and DoC = 0.1 mm).

4.3. Tool Wear

Tool wear criteria such as flank wear, tool edge radius and tool diameter are crucial in assessing tool life. These criteria depend on several factors and significantly influence cutting forces, surface quality and burr formation. This study used a two-flute end mill to machinate the Al 7075 alloy, while a four-flute end mill was employed for the Ti-6Al-4V alloy. As shown in Figure 12, tool wear is more pronounced in conventional milling (CM) than in USVAM for both aluminium and titanium workpieces.

To evaluate tool wear, the geometry of the end mills was examined three times across cutting lengths up to 300 mm using a scanning electron microscope (SEM) (Figure 13a). The cutting edge radius was measured as a wear criterion for both materials, with each value measured three times to minimise error. The tools were also cleaned through a puff of compressed air before inspection after each cutting condition. The results, shown in Figure 13b, indicate that the cutting-edge radius increases in both CM and USM. However, when vibration was applied, the wear rate of the cutting-edge radius slowed down. For aluminium under CM, the edge radius increased from 4.5 to 5.8 μm as the cutting length extended from 100 to 300 mm. With vibration applied, this value decreased to 5 μm at the maximum cutting length, representing a reduction of approximately 16%. A similar trend was observed for titanium, with a lower reduction rate of 8% after vibration was applied. This suggests that vibration-assisted milling suppresses the blunting of the cutting edge, leading to longer tool life. The benefits of ultrasonic-assisted machining can be attributed to the enhanced material shear slip and the effective inhibition of chip sticking during chip separation, particularly in titanium [46]. This process not only facilitates more efficient chip discharge but also requires less cutting force, resulting in lower temperatures.

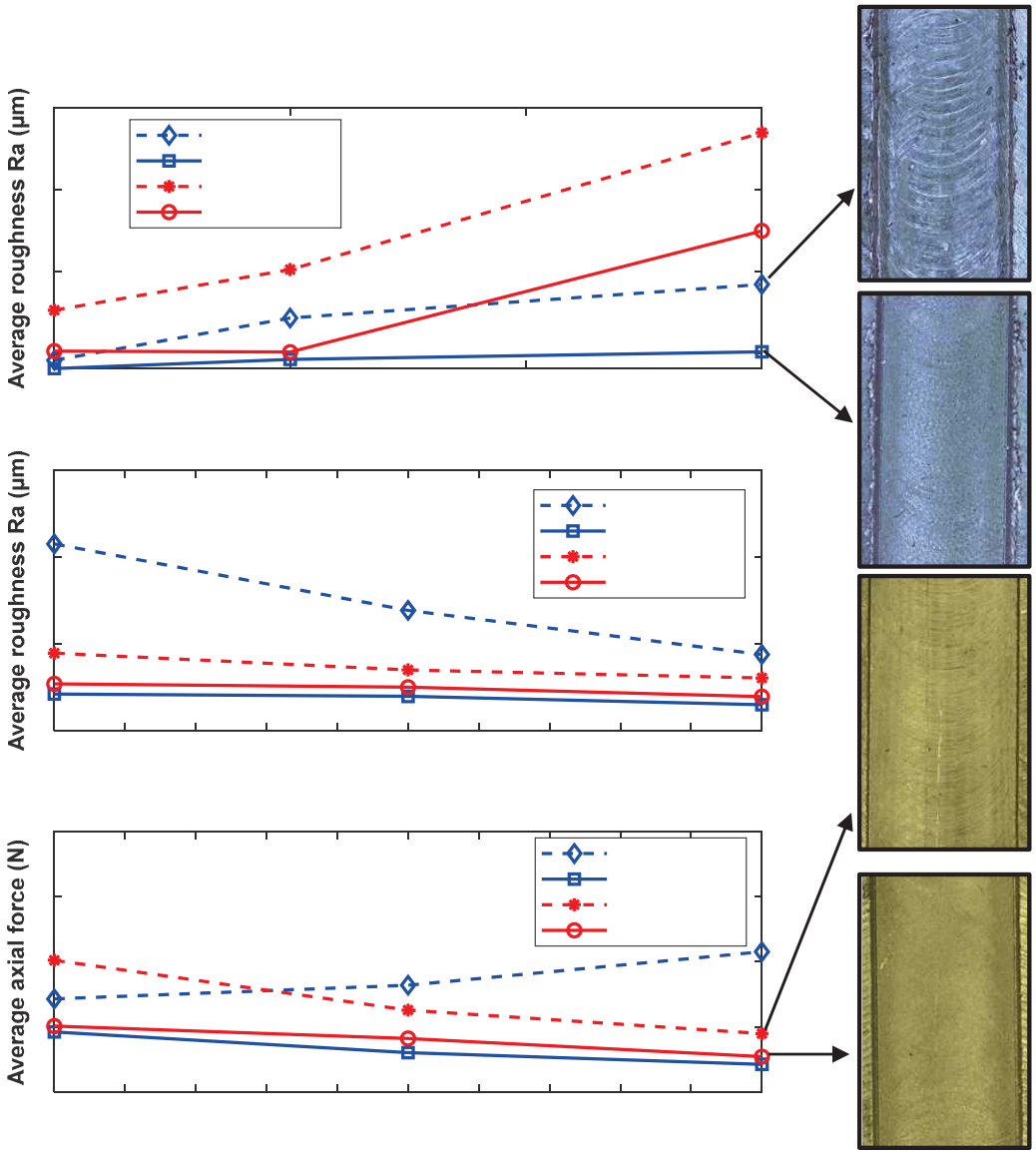


Figure 11. Average surface roughness verses (**a**) DoC (N = 1000 rpm, f = 10 mm/min), (**b**) cutting feed (N = 3000 rpm, DoC = 0.1 mm) and (**c**) cutting speed (DoC = 0.1 mm, f = 10 mm/min).

4.4. AI-Based Models Performance

Obtaining the correct cutting parameters in metalworking is essential for making the process efficient and ensuring high-quality results. Traditionally, finding comprehensive ranges of these parameters involves a lot of trial and error and extensive experimental testing, which can be both time-consuming and expensive. However, with the advent of ML, there was an innovative way to tackle this challenge. This is accomplished by running a limited number of well-designed experiments and predicting the broad feasible range of parameters through AI-based algorithms like SVR or ANN.

Table 6 presents a comparative analysis of the results of using two AI-based ML algorithms, ANN and SVR, in predicting surface roughness and cutting force for Al and Ti. The evaluation metrics used to assess model performance are RMSE and MSE.

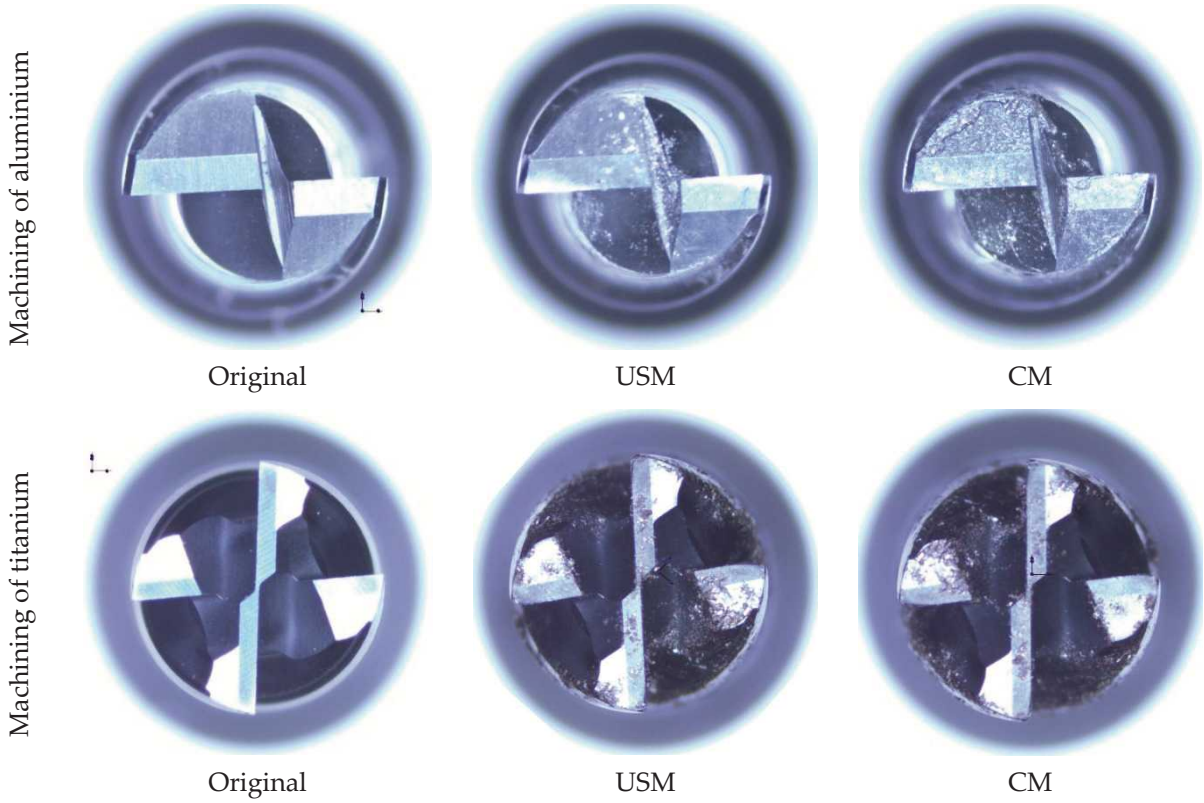


Figure 12. Optical microscope images for the cutting tool at its original state, after USM and after CM for aluminium and titanium alloys.

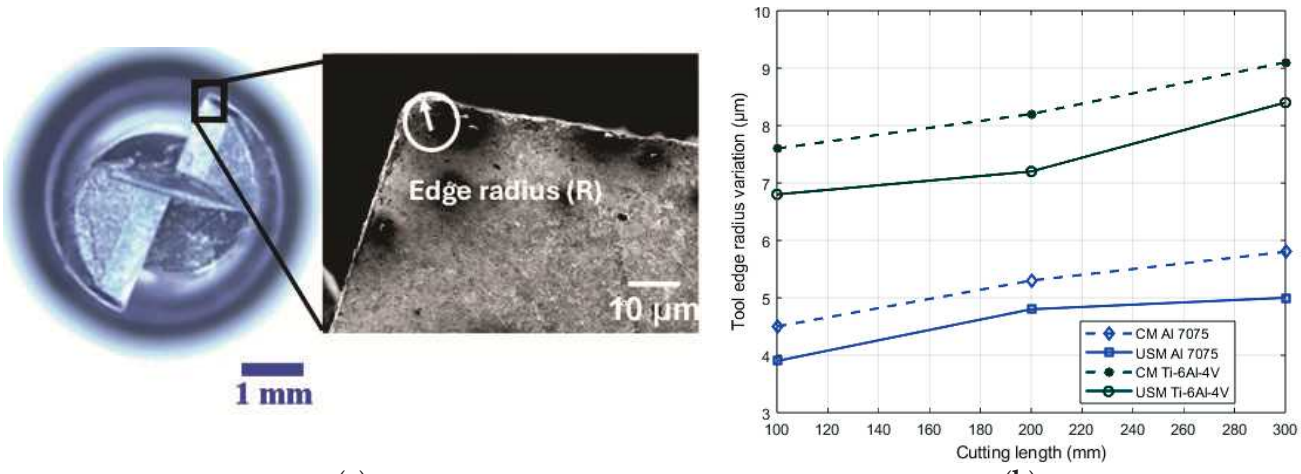
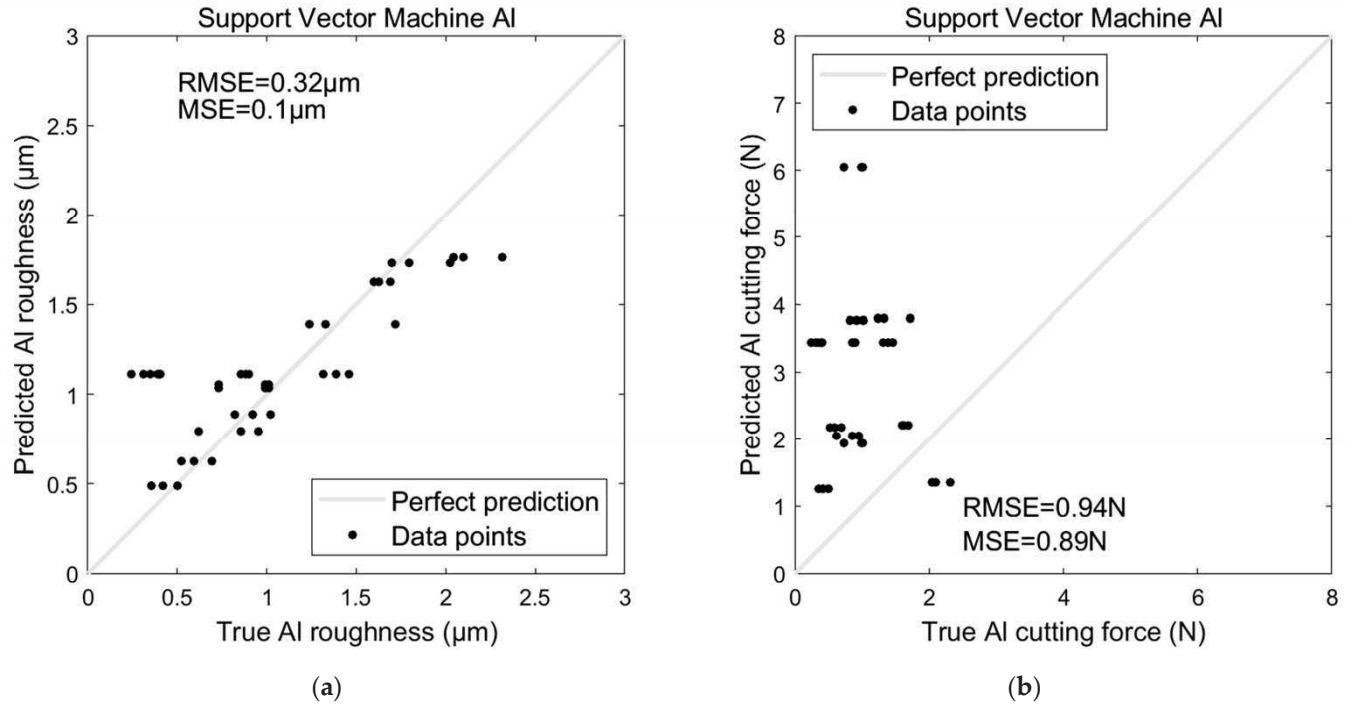


Figure 13. The measurement method of the tool edge radius (**a**,**b**) tool edge radius variation under different milling conditions and cutting lengths for aluminium and titanium alloys.

Table 6. Comparison of AI-based model performances.

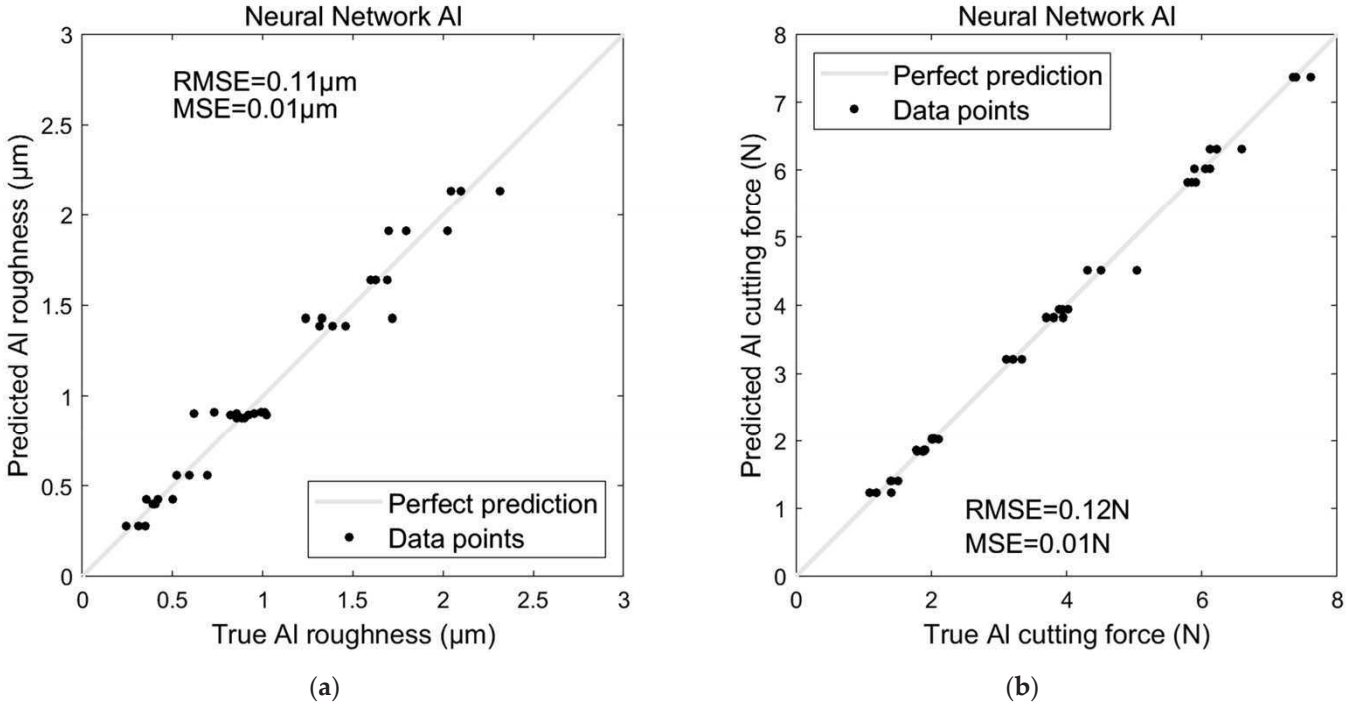
		Roughness	Roughness	Cutting Force	Cutting Force
Material	**AI-Based Model**	**RMSE (µm)**	**MSE (µm)**	**RMSE (N)**	**MSE (N)**
Al	SVR	0.32	0.1	0.94	0.89
Al	ANN	0.11	0.01	0.12	0.01
Ti	SVR	0.32	0.1	0.94	0.89
Ti	ANN	0.12	0.01	0.14	0.02

Regarding the AI-based models' performances on Al, the ANN model significantly outperformed the SVR model when predicting surface roughness for Al (see Figures 14a and 15a). The ANN achieved an RMSE of 0.11 μm and an MSE of 0.01 μm, indicating a high level of accuracy and minimal error in its predictions. In contrast, the SVR model exhibited a notably higher RMSE of 0.32 μm and an MSE of 0.1 μm, suggesting that it struggled to capture the underlying patterns in the data as effectively as the ANN.



(**a**) (**b**)

Figure 14. SVR model for predicting Al (**a**) surface roughness and (**b**) cutting force.

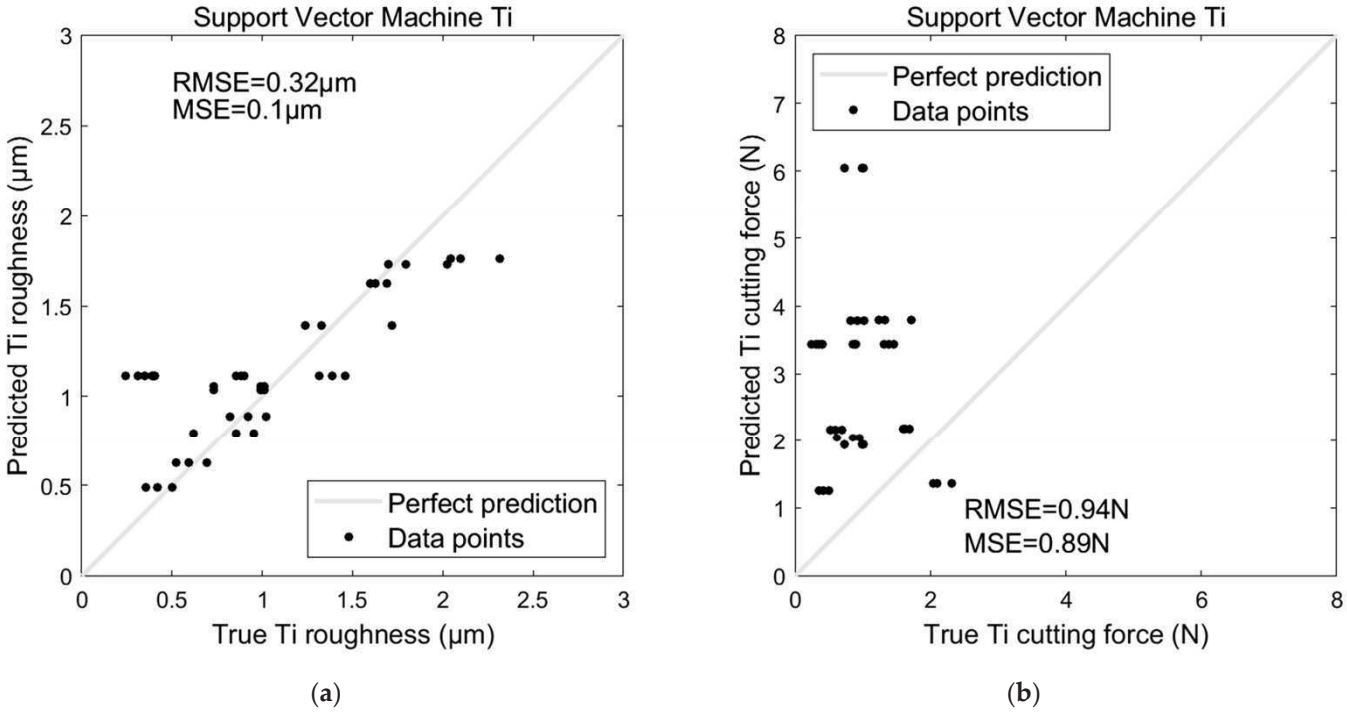


(**a**) (**b**)

Figure 15. ANN model for predicting Al (**a**) surface roughness and (**b**) cutting force.

Similarly, the ANN model again demonstrated superior performance for predicting cutting force (see Figures 14b and 15b), with an RMSE of 0.12 N and an MSE of 0.01 N. The SVR model, however, showed a significant drop in accuracy, with an RMSE of 0.94 N and an MSE of 0.89 N. These results highlight the ANN's ability to make more precise predictions than the SVR model, particularly for the Al dataset.

The performance trends observed with Al were also evident in the Ti dataset regarding the AI-based models' performances on Ti. The ANN model maintained its advantage, producing an RMSE of 0.12 µm and an MSE of 0.01 µm for roughness prediction, compared to the SVR model's RMSE of 0.32 µm and MSE of 0.1 µm (see Figures 16a and 17a). For cutting force prediction, the ANN achieved an RMSE of 0.14 N and an MSE of 0.02 N, while the SVR again performed less favourably with an RMSE of 0.94 N and an MSE of 0.89 N (see Figures 16b and 17b).



(**a**) (**b**)

Figure 16. SVR model for predicting Ti (**a**) surface roughness and (**b**) cutting force.

Validated by the experimental findings, the ANN consistently outperformed the SVR model across materials and prediction tasks. The lower RMSE and MSE values associated with the ANN indicated that it was better suited to capturing the complex relationships within the data, resulting in more accurate and reliable predictions. In contrast, while capable of making predictions, the SVR model did so with considerably more error, particularly in predicting cutting force. This comparison underscored the effectiveness of ANN in tasks requiring high precision. It suggested that ANN was a more robust choice for predicting material properties like roughness and cutting force in manufacturing processes.

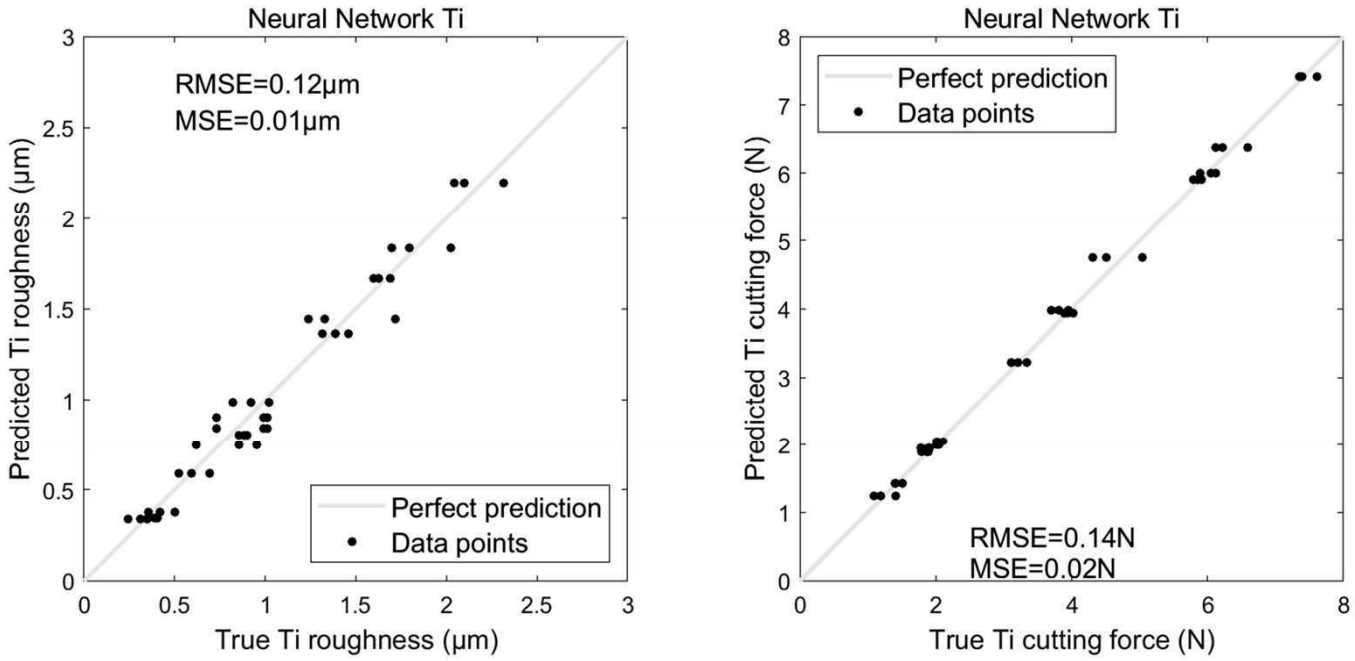


Figure 17. ANN model for predicting Ti (**a**) surface roughness and (**b**) cutting force.

5. Conclusions

The results of this study underscore the potential of ultrasonic vibration-assisted milling as a practical approach for enhancing machining efficiency and product quality. The reduction in cutting forces and improvements in surface finish indicate the technique's viability in both brittle and ductile material scenarios. This research contributes to the broader understanding of advanced machining technologies and offers a foundation for further exploration and implementation.

The introduction of ultrasonic waves causes a noticeable decrease in the axial cutting force by up to 30% in the case of Ti-6Al-4V alloy. However, the application of ultrasonic machining to the Al 7075 alloy did not significantly impact the axial cutting force, likely due to the high plasticity of the aluminium alloy, which causes vibration damping. Surface roughness was enhanced for both materials with the application of ultrasonic waves, with a maximum reduction of approximately 50% for Ti-6Al-4V and 40% for Al 7075. Additionally, increasing the feed rate resulted in a less rough surface in the case of USM, as applying ultrasonic waves at high cutting feed rates does not lead to an interrupted cutting mechanism but instead increases the applied cutting forces. Furthermore, cutting tools for both materials experienced higher wear in CM than USM.

Like most studies in vibration-assessed cutting, the current study also had experimental and numerical limitations. Experimentally, since the ultrasonic generator has a limited frequency range, this study used a single resonance amplitude of 8 μm, the maximum possible amplitude at the resonance. Numerically, a proper randomisation process in selecting AI training, validation and testing data was not practical as it makes the results irreproducible. Therefore, a MATLAB pseudo-random number generator was employed after running all models with multiple seeds and selecting the one that gave the best accuracy.

Considering the consumed time and the high cost of experimental testing for a wide range of metal cutting parameters, advanced methods, such as AI-based algorithms, are needed to accurately predict the effect of different cutting parameters and their companions with a limited experimental range of testing. Compared to SVR, the ANN has proven effective in predicting cutting patterns and relationships, making it perfect for applications in metal cutting assisted by vibration. This study explored how a well-structured

ANN can better predict important cutting parameters such as the cutting force and surface roughness than SVR. The ANN outperformed SVR because it better captured the nonlinear relationships between the machining parameters and outcomes. The ANN's flexible structure allowed it to model these interactions accurately, while SVR struggled. Additionally, ANN's ability to generalise across different conditions made it a good choice for predicting outcomes in ultrasonic vibration-assisted milling.

In conclusion, by utilising experimental data and advanced methods, AI-based ANNs provide accurate forecasts without requiring extensive full-ranged resolution physical experimentations, saving time and costs while optimising the metal-cutting process for better efficiency.

Author Contributions: Conceptualisation, M.S.E.-A., M.B., E.E.S. and A.A.; Methodology, M.S.E.-A., M.B., E.E.S., K.A., M.S. and A.A.; Software, M.S.E.-A., M.B., E.E.S. and A.A.; Validation, M.S.E.-A., E.E.S. and A.A.; Formal analysis, M.S.E.-A., M.B., E.E.S., K.A., M.S. and A.A.; Investigation, M.S.E.-A., E.E.S., K.A. and M.S.; Resources, M.B.; Data curation, M.S.E.-A., M.B., E.E.S., K.A. and M.S.; Writing—original draft, M.S.E.-A., M.B., E.E.S., K.A., M.S. and A.A.; Writing—review & editing, M.B., E.E.S. and A.A.; Visualization, M.S.E.-A. and A.A.; Supervision, M.B. and A.A.; Project administration, A.A.; Funding acquisition, M.B. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the support the Egyptian Academy of Scientific Research and Technology provided for funding the experimental work conducted in this study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to considerations regarding possible future commercialisation.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, Q.; Sun, Q.; Xin, S.; Chen, Y.; Wu, C.; Wang, H.; Xu, J.; Wan, M.; Zeng, W.; Zhao, Y. High-strength titanium alloys for aerospace engineering applications: A review on melting-forging process. *Mater. Sci. Eng. A* **2022**, *845*, 143260. [CrossRef]
2. Nguyen, H.D.; Pramanik, A.; Basak, A.; Dong, Y.; Prakash, C.; Debnath, S.; Shankar, S.; Jawahir, I.; Dixit, S.; Buddhi, D. A critical review on additive manufacturing of Ti-6Al-4V alloy: Microstructure and mechanical properties. *J. Mater. Res. Technol.* **2022**, *18*, 4641–4661. [CrossRef]
3. Bandekhoda, M.R.; Mosallanejad, M.H.; Atapour, M.; Iuliano, L.; Saboori, A. Investigation on the potential of laser and electron beam additively manufactured Ti–6Al–4V components for orthopedic applications. *Met. Mater. Int.* **2024**, *30*, 114–126. [CrossRef]
4. Namlu, R.H.; Çetin, B.; Lotfi, B.; Kılıç, S.E. Investigation of the Combined Effects of Ultrasonic Vibration-Assisted Machining and Minimum Quantity Lubrication on Al7075-T6. *J. Eng.* **2024**, *2024*, 6655471. [CrossRef]
5. Dayi, S.C.; Kiliçay, K. Repairing Al7075 surface using cold spray technology with different metal/ceramic powders. *Surf. Coat. Technol.* **2024**, *489*, 131124. [CrossRef]
6. Santos, M.C.; Machado, A.R.; Sales, W.F.; Barrozo, M.A.; Ezugwu, E.O. Machining of aluminum alloys: A review. *Int. J. Adv. Manuf. Technol.* **2016**, *86*, 3067–3080. [CrossRef]
7. Paktinat, H.; Amini, S. Ultrasonic assistance in drilling: FEM analysis and experimental approaches. *Int. J. Adv. Manuf. Technol.* **2017**, *92*, 2653–2665. [CrossRef]
8. Chao, S.; Kang, R.; Cui, H.; Xu, N.; Dong, Z.; Guo, D.; Wang, Y. Surface morphology and integrity research of ultrasonic-assisted milling of cast superalloy K4169. *J. Mater. Res. Technol.* **2024**, *31*, 1571–1581. [CrossRef]
9. Yassin, M.; Hossam, M.; El-Hofy, H. Applications and designs of vibration-assisted machining devices. *Key Eng. Mater.* **2018**, *775*, 480–486. [CrossRef]
10. Luo, H.; Wang, Y.; Zhang, P. Effect of cutting and vibration parameters on the cutting performance of 7075-T651 aluminum alloy by ultrasonic vibration. *Int. J. Adv. Manuf. Technol.* **2020**, *107*, 371–384. [CrossRef]
11. Bayat, M.; Amini, S.; Hadidi, M. Effect of ultrasonic-assisted turning on geometrical tolerances in Al 2024-T6. *Mater. Manuf. Process.* **2021**, *36*, 1875–1886. [CrossRef]
12. Hu, W.; Du, P.; Qiu, X.; Zhao, X.; Hu, Z.; Zhang, J.; Liu, Y. Enhanced dry machinability of TC4 titanium alloy by longitudinal-bending hybrid ultrasonic vibration-assisted milling. *J. Clean. Prod.* **2022**, *379*, 134866. [CrossRef]
13. Liu, Q.; Xu, J.; Yu, H. Experimental study of tool wear and its effects on cutting process of ultrasonic-assisted milling of Ti6Al4V. *Int. J. Adv. Manuf. Technol.* **2020**, *108*, 2917–2928. [CrossRef]

14. Engelking, L.; Eissel, A.; Schroepfer, D.; Treutler, K.; Kannengiesser, T.; Wesling, V. Optimisation of surface residual stresses using ultrasonic-assisted milling for wire-arc additive manufactured Ni alloy components. *Int. J. Adv. Manuf. Technol.* **2023**, *126*, 4191–4198. [CrossRef]
15. Bin, F.; Zhonghang, Y.; Depeng, L.; Liying, G. Effect of ultrasonic vibration on finished quality in ultrasonic vibration assisted micromilling of Inconel718. *Chin. J. Aeronaut.* **2021**, *34*, 209–219.
16. Xu, J.; Feng, P.; Feng, F.; Zha, H.; Liang, G. Subsurface damage and burr improvements of aramid fiber reinforced plastics by using longitudinal–torsional ultrasonic vibration milling. *J. Mater. Process. Technol.* **2021**, *297*, 117265. [CrossRef]
17. Wang, X.; Jiao, F.; Zhang, S.; Li, Y.; Tong, J.; Niu, Y. Optimization model for ultrasonic-assisted dry helical milling of CFRP based on genetic algorithm. *Int. J. Adv. Manuf. Technol.* **2024**, *131*, 2133–2143. [CrossRef]
18. Bayat, M.; Amini, S. Distortion analysis in axial ultrasonic assisted milling of Al 7075-T6. *Int. J. Lightweight Mater. Manuf.* **2024**, *7*, 678–687. [CrossRef]
19. Ali, M.N.; El-Hofy, H. Experimental Investigation of Vibration Assisted Helical Milling of 7075 Aluminum Alloy. In Proceedings of the International Manufacturing Science and Engineering Conference, Erie, PA, USA, 10–14 June 2019; p. V002T003A007.
20. Fardnam, A.P.; Shalvandi, M.; Haddad, F. An investigation of ultrasonic vibrations effect on surface roughness in aluminum 7075 and AISI 1045 carbon steel during end milling and side milling operation. *Surf. Rev. Lett.* **2022**, *29*, 2250084. [CrossRef]
21. Baraya, M.; Yan, J.; Hossam, M. Improving and Predicting the Surface Roughness and the Machining Accuracy in Ultrasonic Vibration-Assisted Milling. *J. Vib. Eng. Technol.* **2024**, 1–14. [CrossRef]
22. Kim, D.-H.; Kim, T.J.; Wang, X.; Kim, M.; Quan, Y.-J.; Oh, J.W.; Min, S.-H.; Kim, H.; Bhandari, B.; Yang, I. Smart machining process using machine learning: A review and perspective on machining industry. *Int. J. Precis. Eng. Manuf.-Green Technol.* **2018**, *5*, 555–568. [CrossRef]
23. Lu, X.; Hu, X.; Wang, H.; Si, L.; Liu, Y.; Gao, L. Research on the prediction model of micro-milling surface roughness of Inconel718 based on SVM. *Ind. Lubr. Tribol.* **2016**, *68*, 206–211. [CrossRef]
24. Jang, D.-y.; Jung, J.; Seok, J. Modeling and parameter optimisation for cutting energy reduction in MQL milling process. *Int. J. Precis. Eng. Manuf.-Green Technol.* **2016**, *3*, 5–12. [CrossRef]
25. Huang, P.B.; Ma, C.-C.; Kuo, C.-H. A PNN self-learning tool breakage detection system in end milling operations. *Appl. Soft Comput.* **2015**, *37*, 114–124. [CrossRef]
26. Arisoy, Y.M.; Özel, T. Machine learning based predictive modeling of machining induced microhardness and grain size in Ti–6Al–4V alloy. *Mater. Manuf. Process.* **2015**, *30*, 425–433. [CrossRef]
27. Namlu, R.H.; Turhan, C.; Sadigh, B.L.; Kılıç, S.E. Cutting force prediction in ultrasonic-assisted milling of Ti–6Al–4V with different machining conditions using artificial neural network. *AI EDAM* **2021**, *35*, 37–48. [CrossRef]
28. Raju, R.U.; Kottala, R.K.; Varma, B.M.; Barmavatu, P.; Aepuru, R. Precision enhancement in CNC face milling through vibration-aided AI prediction of surface roughness. *Int. J. Interact. Des. Manuf.* **2024**, 1–15. [CrossRef]
29. Zerti, A.; Yallese, M.A.; Zerti, O.; Nouioua, M.; Khettabi, R. Prediction of machining performance using RSM and ANN models in hard turning of martensitic stainless steel AISI 420. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2019**, *233*, 4439–4462. [CrossRef]
30. Karabulut, Ş. Optimisation of surface roughness and cutting force during AA7039/Al2O3 metal matrix composites milling using neural networks and Taguchi method. *Measurement* **2015**, *66*, 139–149. [CrossRef]
31. Pourmostaghimi, V.; Zadshakoyan, M.; Badamchizadeh, M.A. Intelligent model-based optimisation of cutting parameters for high quality turning of hardened AISI D2. *AI EDAM* **2020**, *34*, 421–429.
32. El Shrief, E.; El-Megharbel, A.; El Domiaty, A.; Abd El-Hafez, H. Residual stress effects on fatigue crack propagation in Butt–Welded joints for 304 stainless steel sheets. *Manuf. Rev.* **2021**, *8*, 19. [CrossRef]
33. Gadallah, E.A.E.; El-Fahhar, H.H.; Ahmed, M.M.; Seleman, M.M.E.-S.; Alzahrani, B.; Alamry, A.; Abd El-Aty, A.; El Shrief, E. Experimental study and finite element modelling of pure copper tube fabrication via the parallel tubular channel angular pressing (PTCAP) process. *Mater. Res. Express* **2024**, *11*, 076504. [CrossRef]
34. Baraya, M.; El-Asfoury, M.S.; Fadel, O.O.; Abass, A.J.S. Experimental Analyses and Predictive Modelling of Ultrasonic Welding Parameters for Enhancing Smart Textile Fabrication. *Sensors* **2024**, *24*, 1488. [CrossRef]
35. Baraya, M.Y.; Hossam, M. Design of an electromechanical system for measuring and monitoring micro-ultrasonic amplitude of Langevin transducer. *Int. J. Adv. Manuf. Technol.* **2020**, *107*, 2953–2965. [CrossRef]
36. DSS Corp. *Abaqus 6.12 Example Problems Manual Volume II: Other Applications and Analyses*; Karlesson & Sorensen, Inc.: Providence, RI, USA, 2000.
37. *6.10, A.U.s.M.V. 2000Hibbitt*; Karlesson & Sorensen, Inc.: Providence, RI, USA, 2000.
38. Singh, K.; Singh, H.; Vardhan, S.; Mohan, S. Mechanical study of Al 7050 and Al 7075 based metal matrix composites: A review. *Mater. Today Proc.* **2021**, *43*, 673–677. [CrossRef]
39. Liu, Z.; Xu, J.; Han, S.; Chen, M. A coupling method of response surfaces (CRSM) for cutting parameters optimisation in machining titanium alloy under minimum quantity lubrication (MQL) condition. *Int. J. Precis. Eng. Manuf.* **2013**, *14*, 693–702. [CrossRef]
40. Kutz, M. *Handbook of Materials Selection*; John Wiley & Sons: Hoboken, NJ, USA, 2002.
41. Pradeep, P.; Kumar, P.; Lawrence, I.D.; Jayabal, S. Characterization of particulate reinforced aluminium 7075/TiB_2 composites. *Int. J. Civil Eng. Technol.* **2017**, *8*, 178–190.

42. Shao, F.; Liu, Z.; Wan, Y.; Shi, Z. Finite element simulation of machining of Ti-6Al-4V alloy with thermodynamical constitutive equation. *Int. J. Adv. Manuf. Technol.* **2010**, *49*, 431–439. [CrossRef]
43. Alharbi, W. *Development of a Closed Loop Control System for Vibratory Milling*; Liverpool John Moores University: Liverpool, UK, 2018.
44. Yang, L.; Zhibing, L.; Xibin, W.; Tao, H. Experimental study on cutting force and surface quality in ultrasonic vibration-assisted milling of C/SiC composites. *Int. J. Adv. Manuf. Technol.* **2021**, *112*, 2003–2014. [CrossRef]
45. Kiswanto, G.; Zariatin, D.L.; Ko, T.J. The effect of spindle speed, feed-rate and machining time to the surface roughness and burr formation of Aluminum Alloy 1100 in micro-milling operation. *J. Manuf. Process.* **2014**, *16*, 435–450. [CrossRef]
46. Zheng, L.; Chen, W.; Huo, D. Investigation on the tool wear suppression mechanism in non-resonant vibration-assisted micro milling. *Micromachines* **2020**, *11*, 380. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

remote sensing

MDPI

Article

Estimating Three-Dimensional Resistivity Distribution with Magnetotelluric Data and a Deep Learning Algorithm

Xiaojun Liu [1,*], James A. Craven [2], Victoria Tschirhart [2] and Stephen E. Grasby [1]

[1] Natural Resources Canada, Geological Survey of Canada, Calgary, AB T2L 2A7, Canada; steve.grasby@nrcan-rncan.gc.ca
[2] Natural Resources Canada, Geological Survey of Canada, Ottawa, ON K1A 0G1, Canada; jim.craven@nrcan-rncan.gc.ca (J.A.C.); victoria.tschirhart@nrcan-rncan.gc.ca (V.T.)
* Correspondence: jon.liu@nrcan-rncan.gc.ca

Abstract: In this study, we describe a deep learning (DL)-based workflow for the three-dimensional (3D) geophysical inversion of magnetotelluric (MT) data. We derived a mathematical connection between a 3D resistivity model and the surface-observed electric/magnetic field response by using a fully connected neural network framework (U-Net). Limited by computer hardware functionality, the resistivity models were generated by using a random walk technique to enlarge the generalization coverage of the neural network model, and 15,000 paired datasets were utilized to train and validate it. Grid search was used to select the optimal configuration parameters. With the optimal model framework from the parameter tuning phase, the metrics showed stable convergence during model training/validation. In the test period, the trained model was applied to predict the resistivity distribution by using both the simulated synthetic and the real MT data from the Mount Meager area, British Columbia. The reliability of the model prediction was verified with noised input data from the synthetic model. The calculated results can be used to reconstruct the position and shape trends of bodies with anomalous resistivity, which verifies the stability and performance of the DL-based 3D inversion algorithm and showcases its potential practical applications.

Keywords: deep learning; neural network; inversion problem; magnetotelluric; Mount Meager

1. Introduction

Geophysical survey methods, such as reflection seismic, electromagnetic (EM) and potential fields, have been widely used in exploration for natural resources and fundamental research on deep geological structures [1–3]. Inversion is a popular technique in geophysical data analysis that produces models of the subsurface physical properties and is a reversed process based on a physical law system [4]. The essential step during the inversion process involves the iterative optimization of a created objective function. The advantages of traditional iterative inversion methods include obtaining better details of the subsurface physical structure and their suitability for large survey areas. However, iterative inversion techniques face some challenges, such as the need for an acceptable reference model, the non-uniqueness of the calculated model, and the large time consumption of the calculations.

Although regularization stabilizers and weight matrices have been employed to improve the accuracy of inverse solutions, further techniques are needed to improve prediction. An alternative inversion technique and computer hardware improvements have permitted the emergence of a data-driven deep neural network framework to predict subsurface physical models from several different geophysical methods [5,6]. Das et al. [7] proposed a convolutional neural network-based inversion framework for obtaining elastic models from full-waveform seismograms. This application of the DL algorithm demonstrates that it has great potential in predicting reservoir properties directly. Puzyrev [8] and

Moghadas [9] presented 1D inversion results of electromagnetic induction (CSEM and TEM survey) data with convolutional neural network frameworks. Guo et al. [10] employed a supervised descent method (SDM) for 2D MT data inversion to reduce uncertainty and involved the design of a general training set from prior knowledge of the survey field to balance the computation time and accuracy of the machine learning inversion procedure. For potential field inversion with deep learning, previous researchers have presented gravity and magnetic inversion with a deep learning approach [11,12]. All these studies have demonstrated the potential applicability of DL-based inversion for different geophysical methods.

Because of the output uncertainty and large computational requirements, the use of traditional 3D inversion processes is always a challenging task. To overcome the drawbacks of the high computational needs and unstable output of traditional iterative inversion methods, employing DL architectures to retrieve unknown 3D geo-electric structures can be considered. Based on the previous study of one-dimensional (1D) magnetotelluric (MT) inversion [13], here, the neural network model is extended to solve a 3D inversion problem. In this work, we propose MT 3D inversion based on the 3D U-Net network. U-Net is a fully convolutional neural network which was originally used for image segmentation and object detection [14–16] and has been used in different DL practical applications [13,16]. The 3D U-Net framework can also be used in inversion workflows by building a mathematical connection between a 3D resistivity model and the observed MT response with optimized weight parameters. The common challenge in 3D inversion is the large amount of forward modelling, which requires powerful computational hardware. DL inversion needs a large amount of modelling computation for generating training/validation datasets. Furthermore, the datasets need to show a wider coverage for neural network models. To balance these factors, one main difference in this study is that we developed random walk for data generation to enlarge the coverage of model generalization with smaller datasets, which reduces the computation time of model training. Compared with traditional iterative inversion methods, the main advantages of this inversion method are the instant prediction after the neural network is created and the relative reliability of the output to different noise levels.

In this paper, we start by summarizing the MT method, as well as its inversion, and then describe the deep learning (DL)-based 3D inversion workflow for MT data analysis. We present evidence of data generation with a random walk generator, model training/validation and parameter tuning, followed by practical testing with synthetic models and real MT data from the Mount Meager volcanic area in British Columbia, Canada. This area is of interest due to its potential for geothermal energy. Related Python codes were developed to realize the algorithm workflow, and independent MT datasets were created to validate the trained neural network model.

2. Methods

2.1. Magnetotelluric Method

The MT method is a passive electromagnetic (EM) technique that uses natural EM signals to image the subsurface resistivity distribution. The EM signal is assumed to be generated outside the Earth (solar wind and lightning), and it includes a time-varying magnetic field ($\boldsymbol{H}$) and its induced orthogonal secondary electric field ($\boldsymbol{E}$) (Figure 1). As small fraction of EM signal travels into the Earth and induces electrical currents, the measured EM field at the Earth's surface contains electrical resistivity information of the subsurface. Magnetotelluric has been used extensively in mineral and geothermal exploration by analyzing the retrieved subsurface resistivity distribution (e.g., [1,2]). During an MT survey, the electric field and magnetic fields are collected in the time domain,

and they are then converted into the frequency domain. The electromagnetic induction phenomenon can be described by Maxwell equations [1,4,17], as follows:

$$\begin{cases} \nabla \cdot \boldsymbol{E} = \frac{\rho}{\varepsilon_0} \\ \nabla \cdot \boldsymbol{B} = 0 \\ \nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \\ \nabla \times \boldsymbol{B} = \mu \boldsymbol{J} + \mu\varepsilon \frac{\partial \boldsymbol{E}}{\partial t} \end{cases}, \tag{1}$$

where $\boldsymbol{B}$ is the magnetic flux density, $\boldsymbol{B} = \mu \boldsymbol{H}$, μ is the magnetic permeability, $\boldsymbol{J} = \sigma \boldsymbol{E}$ is the electric current density, σ is the electrical conductivity, ε is the permittivity and ρ is the total electric charge density.

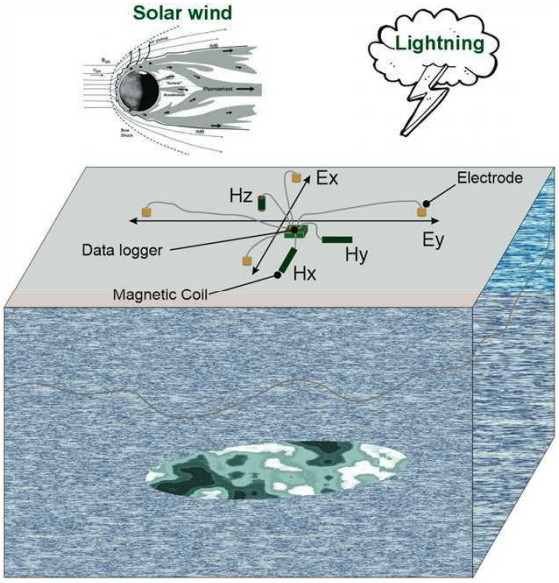


Figure 1. Diagram illustrating the principles of an MT survey (Ex and Ey: electric fields; Hx, Hy and Hz are magnetic fields).

For a 3D structure in the subsurface, the electric and magnetic fields can be obtained by solving a discrete Maxwell equation (Equation (1)) with boundary conditions. The recorded time-series signal of the electric and magnetic fields at the Earth's surface can be converted into the frequency domain, with their ratio being expressed as an impedance tensor, $\boldsymbol{Z} = \frac{\boldsymbol{E}}{\boldsymbol{H}}$ and tipper $\boldsymbol{T}$. The tensors can be written in Cartesian coordinates as

$$\begin{bmatrix} Z_{xx} & Z_{xy} \\ Z_{yx} & Z_{yy} \end{bmatrix} = \begin{bmatrix} E_x \\ E_y \end{bmatrix} \begin{bmatrix} H_x & H_y \end{bmatrix}^{-1}, \tag{2}$$

$$H_z = \begin{bmatrix} T_x & T_y \end{bmatrix} \begin{bmatrix} H_x \\ H_y \end{bmatrix}. \tag{3}$$

Impedance is associated with amplitude and phase, and is related to the apparent resistivity ρ_a and phase φ of each frequency. The vertical transfer function $\boldsymbol{T}$ can be used to define the induction vectors to understand the lateral conductivity variation [18].

An understanding of the traditional inverse problem and computational procedure will assist in designing the workflow of the deep learning inversion. In general, the MT data inversion process aims to predict the resistivity of the Earth's interior from observed data [13,19–21]. The data values (impedance or apparent resistivity) depend on the subsurface resistivity distribution. Mathematically, the traditional inversion technique conducts iterative operations to minimize the parametric function, which includes both a

misfit functional and a stabilizing functional. The objective function P^{α} can be defined as follows (Equation (4)):

$$P^{\alpha}(m,d) = f(m,d) + \alpha s(m), \tag{4}$$

where $f(m,d)$ is a parametric objective function and $s(m)$ is a model-stabilizing functional; α is a regularization parameter; m is the unknown model parameter (resistivity in this case); and d represents the observed response data. The objective function can be given by

$$f(m,d) = \|W_d(A(m) - d)\|^2. \tag{5}$$

Based on the least squares criterion, the model-stabilizing functional can be simplified as follows:

$$s(m) = \sum_{i=1}^{N} \left(W_m \left(m_i - m_i^{apr} \right) \right)^2, \tag{6}$$

where is W_d and W_m are the data weighting matrix and model weight matrix, respectively; $A(m)$ is an operator of the theoretical data from the model parameters; m^{apr} is the reference model; and N is total number of model parameters.

There are different inversion optimization methods that minimize the objective functional (Equation (4)), such as least squares, Gauss–Newton and nonlinear conjugate gradient. These techniques and software have been developed and applied on MT inversion in the past decades [4,21,22], e.g., the Occam and ModEM open-source packages. Here, we use the ModEM (version 1.2.0) package to conduct 3D forward modelling for calculating the impedance of the MT response.

2.2. Deep Learning-Based 3D Inversion Workflow

The fundamental workflow of DL-based 3D inversion is similar to a 1D CNN [13]. The main differences are the neural network structure, the feature types of the input dataset and the number of training/validation datasets required. This DL-based inversion algorithm builds a relationship between the observed data and a physical model, which can be expressed as $\boldsymbol{m} = Net(\boldsymbol{d};\theta)$ [12]. $Net(.)$ is a neural network, and θ represents the parameters in the network. The physical property is predicted with a trained neural network model by minimizing the objective function $\boldsymbol{O}(.)$, which is similar to Equation (4) of the traditional inversion, written as the optimization of the neural network parameters $\widetilde{\theta} = \min(\boldsymbol{O}(m, Net(\boldsymbol{d};\theta)))$.

2.2.1. Neural Network Architecture

Based on the previous U-Net architecture [15,23], a 3D network structure was redesigned for conducting the inversion procedure, which connects the observed response data (input) and the physical property model (output). This U-Net architecture (Figure 2) shows feature extraction and output shape reconstruction, which include an encoding path to extract feature information and a decoding path to restore the extracted information. The output data are reconstructed through a combination of convolutions and up-sampling.

Similar to the standard 2D U-Net, the 3D network contains data analysis and a synthesis path, each with four layers defined by setting different kernel sizes. In the 3D U-Net, the number of filters denoted at the bottom of the box increases from 32 to 128. The increased number of filters allows the neural network to propagate contextual information to complex layers for a better understanding of the features. Each layer contains two $3 \times 3 \times 3$ filters in convolutional operators, followed by a rectified linear unit (ReLU) activation function, $f(x) = \max(0, x)$. Compared with the sigmoid function, ReLU shows its advantage of overcoming the vanishing gradient problem. A residual block is added into each layer to make the network deeper. A $2 \times 2 \times 2$ max pooling with strides of two in each dimension is added between layers, which is followed by 0.3 dropout (Figure 2). The stride is a parameter to control the filter movement; the max-pooling layer is a method to down-sample features by calculating the maximum value for each patch of a feature, and the dropout layer is a regularization method that randomly sets some hidden layer neurons

to zero during each epoch. In the last output layer, expansive features are concatenated into one interpretation as a fully connected layer, and a $1 \times 1 \times 1$ convolutional operator is added to reset the dimensionality of the output channels to the output size. More details of this network framework can be found in Long [14], Ronneberger et al. [15], Chen et al. [16], Çiçek et al. [23], Hochreiter [24] and Zhou et al. [25].

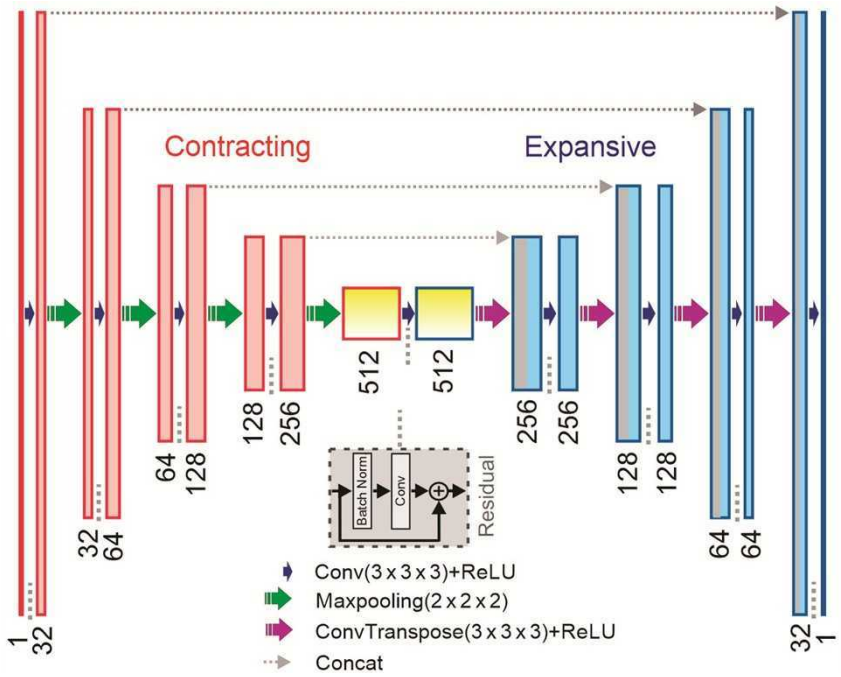


Figure 2. A schematic of the 3D U-Net architecture, consisting of an encoder analysis path and a decoder synthesis path.

2.2.2. Dataset Preparation

The generation of the dataset is an essential step in supervised learning that decides if the coverage of a trained network model is generalized. In DL-based inversion, the geophysical data are collected as normal on the Earth's surface and are then used as input for neural network training. The subsurface resistivity of a study area is used as the mask of the neural network output. Unlike the 1D deep learning inversion, the training datasets for the 3D model cannot be generated randomly in the entire solution space because of the heavy computational cost. To save computation time, and to enhance the representativeness of the dataset's general distribution characteristics, a random walk method was used for data generation [12]. This sets a maximum of four cells randomly and then extends the resistivity values to the surrounding cubic area (Figure 3). The resistivity value of cells in the random paths are assigned in the range (1–1000 Ωm) in common logarithmic space, and the surrounding values are interpolated. Different anomaly combinations are included to improve dataset diversity.

Based on the distribution of the MT stations, synthetic resistivity models are generated, and the impedance responses are simulated. For this synthetic model study, the physical model is gridded with $17 \times 17 \times 25$ cells, the thickness of the layers is fixed throughout the model and starts from 20 m at the top and increases with an index to 3500 m in the bottom layer. The number of stations is 49 (Figure 4), with 32 frequency points ranging from 0.001 to 100 Hz for data generation. A total of 15,000 pairs of model samples and responses are generated for this synthetic case study.

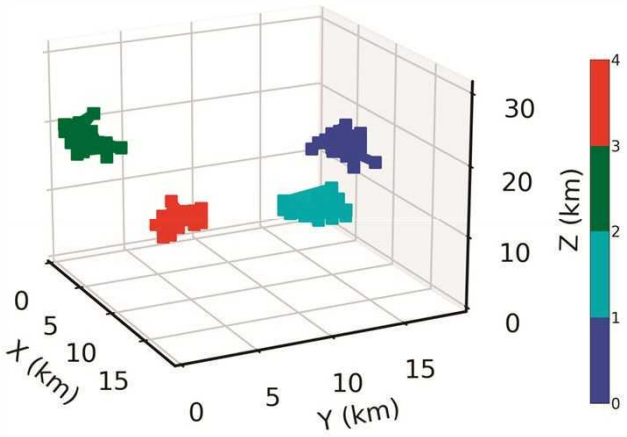


Figure 3. A sample of the resistivity model from the random walk generator (the color represents resistivity values in common logarithmic space).

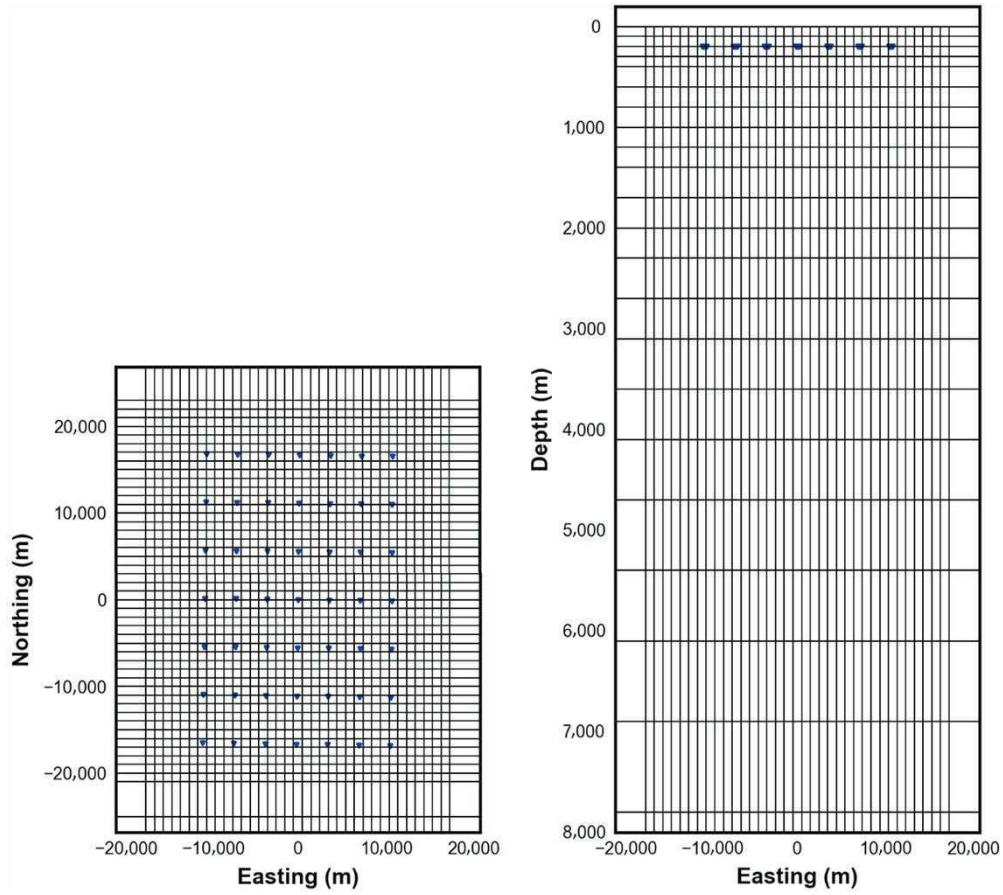


Figure 4. Plane (**left**) and vertical (**right**) views of the synthetic model mesh grid with MT stations indicated (blue dots).

The ModEM forward package is used to simulate the surface response of each physical model and to build the datasets for model training. The data format for the DL algorithm is the same as the input and output of the MTpy package [26]. The pairs of input (real and imaginary impedance) and output (resistivity model) are created for optimizing the neural network parameters. To prevent an unbalanced range of input features affecting model stability, the data are standardized before being input into model training. The common logarithmic resistivity is set as the target parameter, and the input X is scaled to

$\widetilde{X}$ with mean μ and standard deviation σ of the feature values during the model training/validation process.

$$\widetilde{X} = \frac{X - \mu}{\sigma}. \tag{7}$$

2.2.3. Loss Function

Similar to traditional inversion, the neural network training process needs to define a loss function as a metric to evaluate the model's predictive performance. A loss function is used to compare the model's predicted values and the true values so that the weights can be updated to reduce the metrics during evaluation. For the inversion regression problem, the popular loss functions are mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and their combinations (Equation (8)).

$$\begin{cases} MAE = \frac{1}{N}\sum_1^N \left| y_i^{pre} - y_i^{mod} \right| \\ MSE = \frac{1}{N}\sum_1^N \left(y_i^{pre} - y_i^{mod} \right)^2 \\ RMSE = \sqrt{\frac{1}{N}\sum_1^N \left(y_i^{pre} - y_i^{mod} \right)^2} \end{cases}, \tag{8}$$

where y_i^{pre} is the predicted value, y_i^{mod} is the input model and N is the total number of output data.

In order to select an optimal loss function, the proposed neural network and generated training datasets are used to discover the influence of different loss functions for a 3D inversion workflow. Figure 5 shows the convergence curves of different loss functions (MAE, MSE and RMSE). In general, both the training and validation curves of the loss functions converge stably. The convergence of RMSE and MAE decreases rapidly in the early training stage. Comparing the curves of MAE and RMSE, RMSE is relatively stable in the later period of epochs. The magnitude of oscillations of RMSE is smaller than that of MAE. The MSE curve shows much smoother convergence, but the difference between training and validation loss is relatively large at the beginning, since it effectively penalizes larger errors more severely. The minimum metric values of validation are 0.05883, 0.01418 and 0.09238 during 150 epochs.

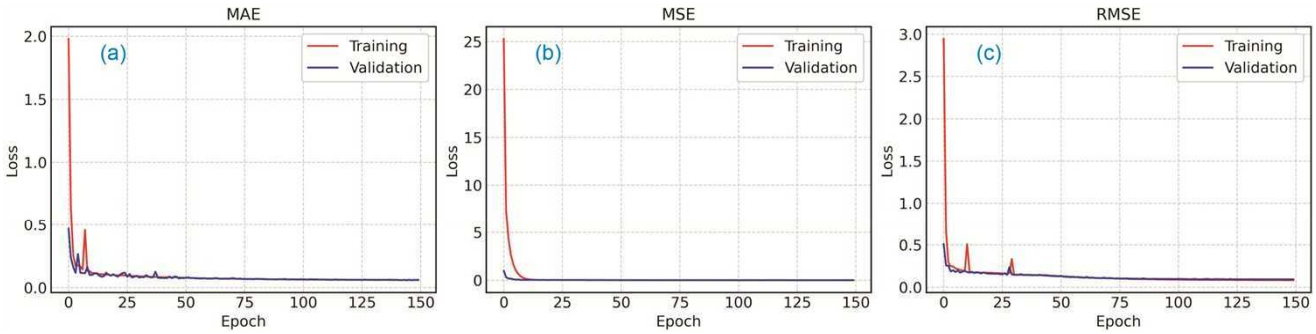


Figure 5. The convergence curves of model training and validation with different loss functions ((**a**) MAE, (**b**) MSE and (**c**) RMSE).

Based on the comparison in Figure 5, RMSE is chosen as the loss function metric for network training. The adaptive moment estimation optimizer (Adam) is used to stabilize the training process and converge the objective function [27]; it is an optimization algorithm commonly used in deep learning. During parameter testing, the initial learning rate is 0.0001 and is decreased linearly to a smaller value with a ratio of 0.8 if there is no improvement in the metric over 5 consecutive epochs.

2.2.4. Hyperparameters Tuning

During network model training, parameter selection affects the convergence of the loss function. Inappropriate parameters lead an objective function to land on the local minima, giving rise to low prediction performance. Finding optimal hyperparameters is a significant step in stabilizing model training and obtaining an accurate prediction [12]. Popular search approaches are grid search and random search, where the two major tuning parameters are learning rate and batch size. In this test, the grid search method is employed to search for all possible configurations and identify the set of the best parameters. Because of the limited computational capacity, the two-dimensional case is tested, and the parameter range is chosen based on previous model training experience.

The range of values (0.000001, 0.01) was considered for the learning rate, and training processing was conducted to check for convergence within 10 epochs as the learning rate increased. Based on Figure 6, the optimal range of the learning rate was assumed to be $(4 \times 10^{-5}, 4 \times 10^{-4})$. The learning rates selected were 0.00001, 0.00005, 0.0001 and 0.0005, and the batch sizes were set to 8, 16, 24 and 32. Each position of the grid was then searched to optimize these hyperparameter values based on the convergence loss function. A computer with a 16-core CPU, 128 G of memory and one Nvidia Quadro-RTX 5000 GPU was used for conducting the grid search calculation. Figure 7 displays the minimum value distribution of the metrics. The region in yellow represents the best area for choosing a learning rate and a batch size for model training. Based on the scores, the final optimal hyperparameters chosen were 0.00005 and 32 (purple ellipse area). During model training and validation, the weights with the best validation metrics were saved for prediction. The maximum epoch was set to 150, with an early-stopping parameter of 30.

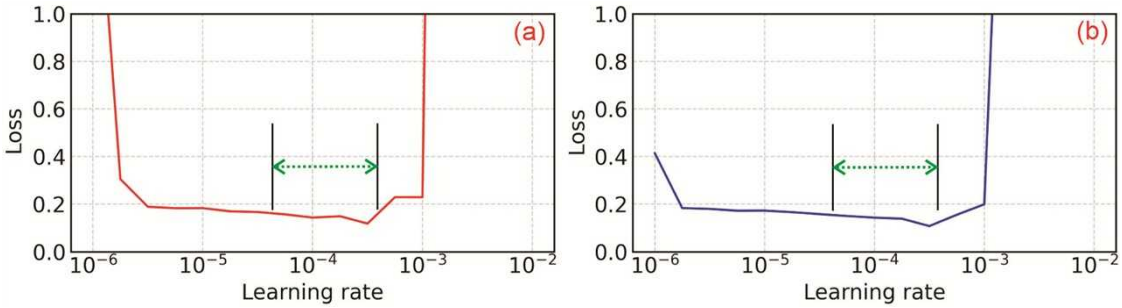


Figure 6. Learning rate search showing the metric curves of minimized loss function vs. learning rates ((**a**) training and (**b**) validation; the green double arrows display the optimal range of the learning rate).

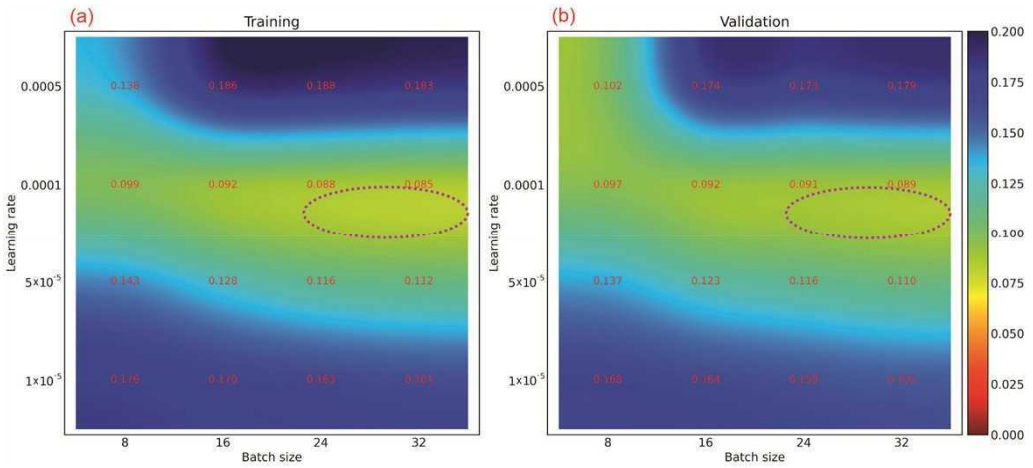


Figure 7. The grid search layout results showing the distribution contour and minimized metric values, where (**a**) model training and (**b**) model validation.

2.2.5. Inversion Process

The process of DL-based 3D inversion is illustrated in Figure 8. It includes the components of deep learning network and convolutional neural network architecture of MT 3D inversion, which includes three main stages: dataset preparation, model training and validation, and prediction. The process is as follows: (1) The random resistivity models are generated by 3D random seeds and by forward modelling the surface response with ModEM, with data engineering to exaggerate the datasets and data standardizing normalization pre-processing to ensure uniformity of the input feature values, and the data are split into training and validation groups in a 80/20 ratio. (2) A neural network model is created, and the initial network parameters and training parameters are set, for example, filter number, kernel size, a batch size of 32, a learning rate of 0.00005, a kernel size of $3 \times 3 \times 3$, etc. (3) The weight of the neural network is calculated based on the loss function by using the Adam algorithm, and the network optimizes the end-to-end mapping relationship by batch iteration. (4) The parameters are reset, and a grid search for better predicted results is performed, selecting the optimal parameters with a minimum verification metric and saving the final weight values. (5) The results are predicted with synthetic test datasets and comparison with the true models, and the resistivity distribution model is reconstructed for interpretation. Model training/validation took about 35 h of computation time with one Nvidia Quadro-RTX 5000 GPU. Compared with the ModEM iterative inversion, the training period took a similar amount of time, but the model prediction step was instantaneous. More GPUs can speed up the model training process. Python and related libraries (e.g., Scikit-learn, Tensorflow and Keras), were used to conduct this 3D DL inversion procedure [28,29]. MTpy and ModEM open-source packages were selected to create resistivity models and generate forward response datasets [26,30].

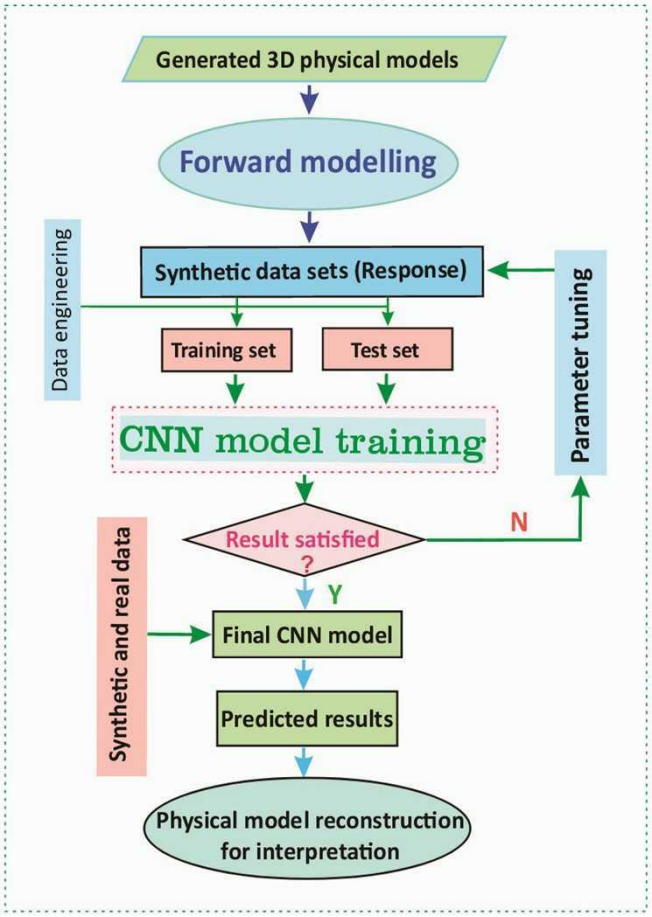


Figure 8. The schematic workflow of DL-based 3D inversion, including dataset preparation, parameter tuning and model training/validation/testing.

3. Results

3.1. Synthetic Model

In order to test the reliability and quantitative functionality of the proposed inversion framework, three different styles of synthetic models were created to verify the trained model. Three-dimensional MT modelling processing (ModEM) was used to compute the impedance and the tipper. The same grid parameters were used for synthetic and forward modelling, which included 49 stations (as in Figure 4). A cut-off was used to disregard the background resistivity value to make the comparison more intuitive. The first model was a dipping dike as a single anomaly body (Figure 9a), where the resistivity of the surrounding background was 400.0 Ωm and the resistivity of the anomaly was 30.0 Ωm. For forward modelling, the same station geometric distribution and frequency range were used. Figure 9b illustrates the predicted result with a resistivity cut-off of 30.0 Ωm. The predicted anomalous center reconstructed the true model, and the depth of the anomalies showed a similar range as the true model.

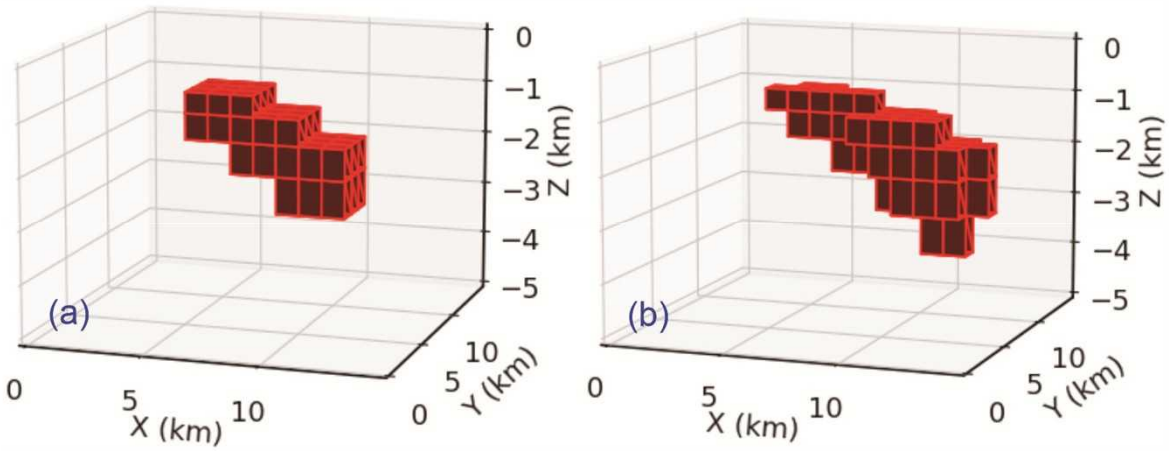


Figure 9. Three-dimensional view of single slipped dike synthetic model ((**a**) true model; (**b**) predicted model).

For verifying the capability of the trained model to identify conductors and resistors, the second model was a T-shaped anomaly with different anomalous resistivity. The resistivity of the conductor and resistor of the T-shaped anomaly were 30.0 Ωm and 795.0 Ωm, respectively, with a background resistivity of 400.0 Ωm (Figure 10a,b). For comparison, the cut-offs of 30.0 Ωm and 795.0 Ωm were used to display the anomalies clearly. We can see that the predicted results (Figure 10c,d) generally reconstruct the shape of anomalies. Although the model shape and anomalous resistivity affected the predicted results, which led to some false structures at the bottom and side boundaries, the trained inversion model could produce a reasonable resistivity distribution image of the subsurface.

The third model comprised two anomalous bodies. Two different resistivity distribution styles were created (Figure 11) for checking the potential interactive effect between two bodies. The first one (a, b) displayed double conductor anomalies (30.0 Ωm) and inversion result with the DL model; the second one (c, d) was the model with one conductor (30.0 Ωm) and one resistor (795.0 Ωm). A cut-off (less than 30.0 Ωm and greater than 795.0 Ωm) was used to disregard the background value of the model and display the resistivity anomalies for easier comparison. From the predicted results on the right side of the figure, one can see that both anomalies can be created in a general overview and that the depth of the two bodies corresponds to the true model. Although the resistivity values in some areas are lower than the true model, the cut-off anomalous values of the prediction can display the trend and central position and the edges of the anomalies. In general, the output can retrieve real values with an acceptable level of accuracy. For a qualitative comparison, the calculated RMSE values were 0.046 and 0.0595, which verifies that the predicted model by

the neural network is reliable and can reconstruct true structures accurately for different anomaly types.

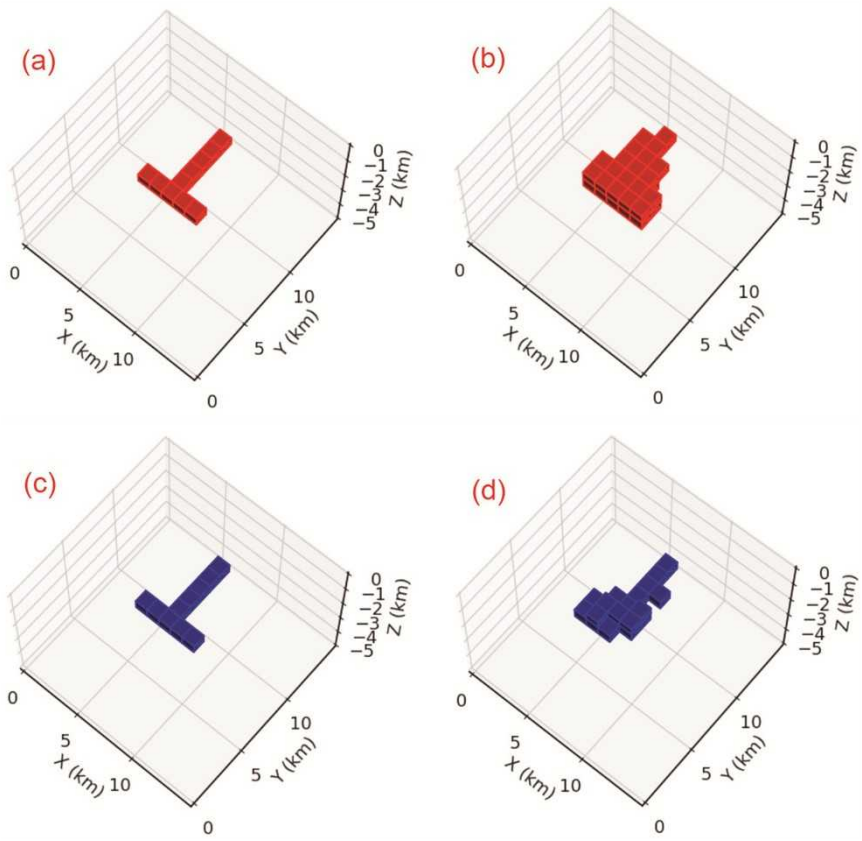


Figure 10. The model sensitivity testing with synthetic models including an anomaly with the same shape and different resistivity ((**a**,**b**) conductor anomaly model and prediction; (**c**,**d**) resistor anomaly model and prediction).

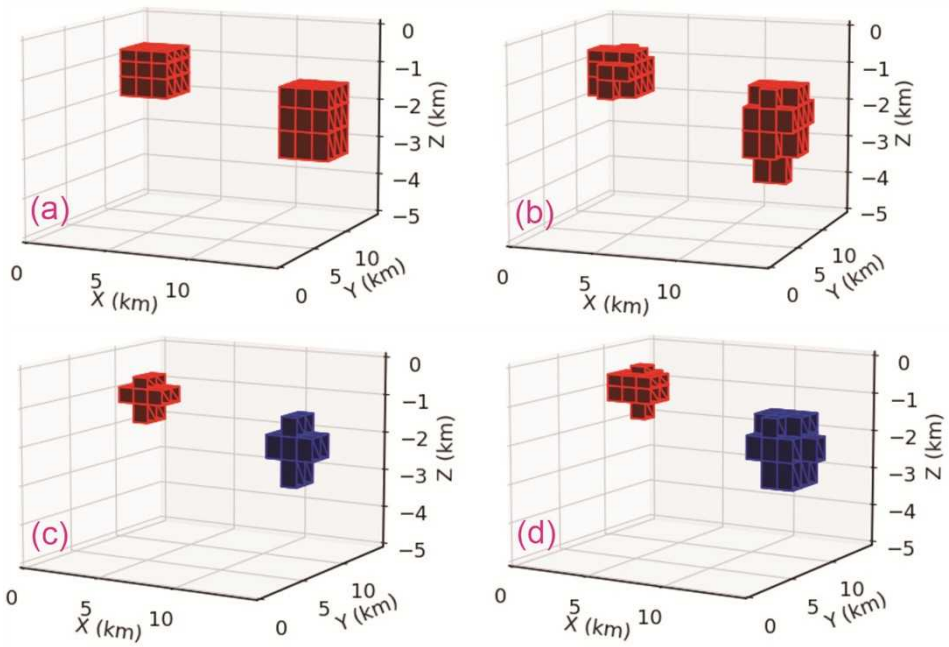


Figure 11. Synthetic models of two anomalies at different depths and the DL 3D inversion results ((**a**,**c**) true models; (**b**,**d**) predicted results).

During synthetic model testing (Figure 11), forward modelling was conducted to calculate the impedance from the true and predicted models. Figure 12 shows the response (real and imaginary parts) contours of real data and forwarded by the predicted model with all MT stations at frequencies of 3.5 Hz (a, c) and 0.0044 Hz (b, d). One can see that the impedance distribution values are slightly biased by lower values than the real data; however, it still effectively displays the trend of impedance variation. This indicates that the proposed DL method can yield inversion results with acceptable accuracy.

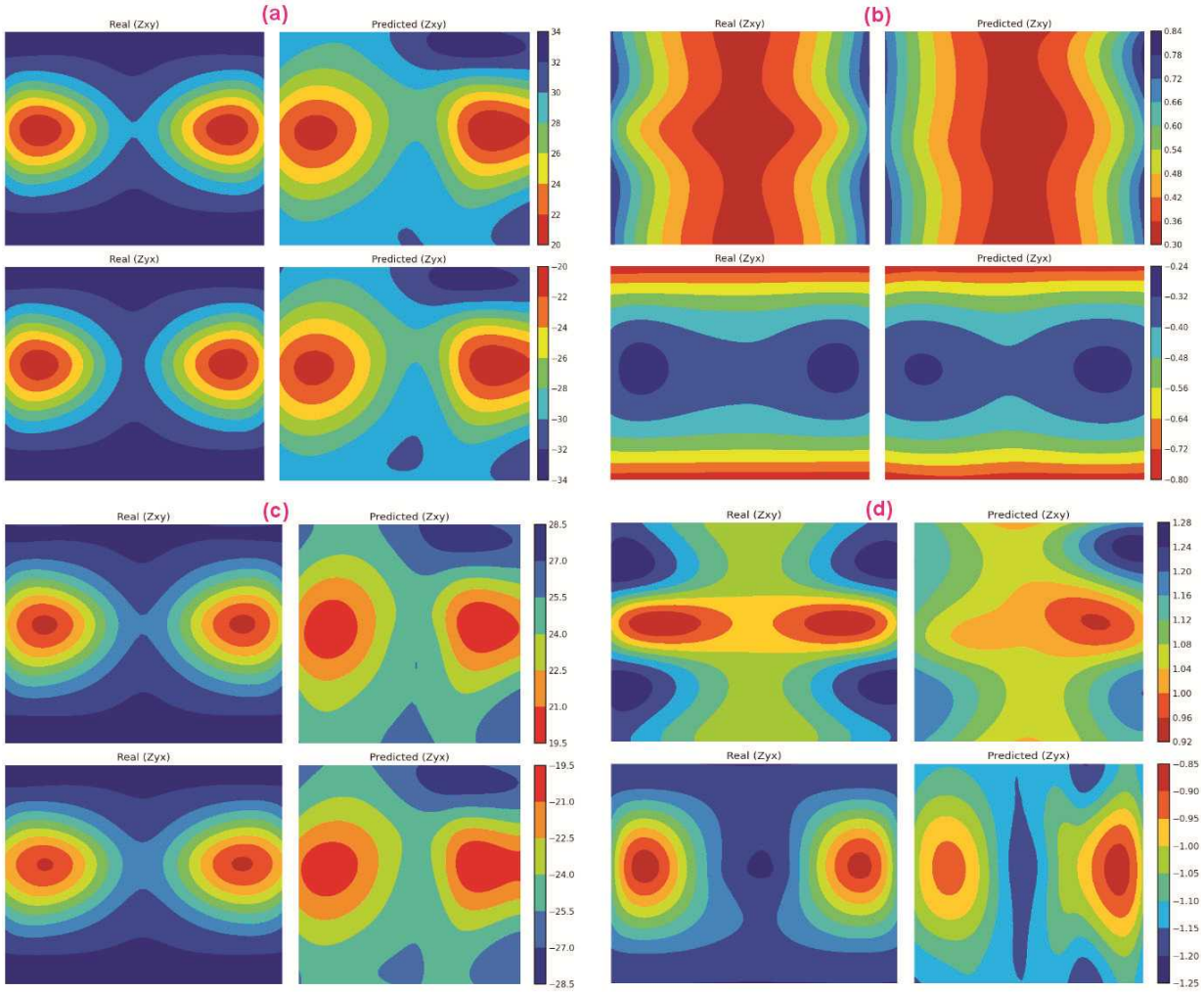


Figure 12. The comparison of simulated impedance at two different frequencies from the true model and the predicted results of the two-body synthetic model ((**a**,**b**) real part; (**c**,**d**) imaginary part; (**a**,**c**) 3.5 Hz; (**b**,**d**) 0.0044 Hz).

3.2. Noise Test

For real world geophysical data, the noise level is a significant factor in deciding if the inverted physical parameters and interpretation of subsurface features are meaningful. In this study, we used noise-free datasets to train and validate the CNN model. To check the stability and performance of the network model prediction in the presence of noise, input datasets with different noise levels were tested. One model with two anomalies at the same depth was used (Figure 13), and different levels (0%, 5% and 10%) of random white noise with a normal Gaussian distribution were added to the impedance tensor before applying it to the input layer.

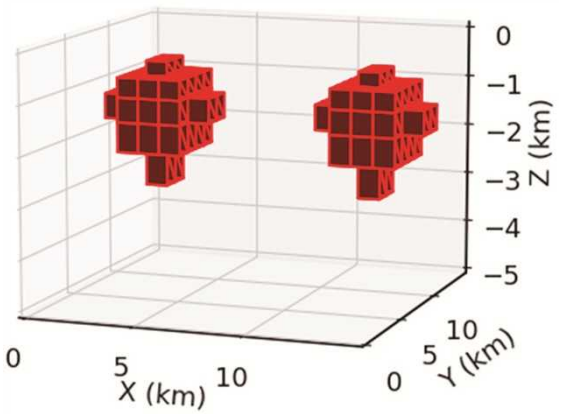


Figure 13. The synthetic model containing two anomalous conductors at the same depth.

Figure 14 displays the horizontal slice of DL inversion resistivities with the noised data. Notably, one can see that the predicted results all show the trend of the true model resistivity variations, which verifies that the effect of noise is minimal. Table 1 lists the noise effect on the final predicted output. The RMSE and correlation coefficient varied with different noise levels. Overall, the predicted results are acceptable. The network model could still predict the anomaly locations with 10% added noise. The RMSE was 0.04384, slightly higher than results with 0% noise, which demonstrates the limited effect of noise to the trained model prediction. Although the higher RMSE had lower R^2 with the increase in the noise level (Table 1), the trained network model could still produce stable results.

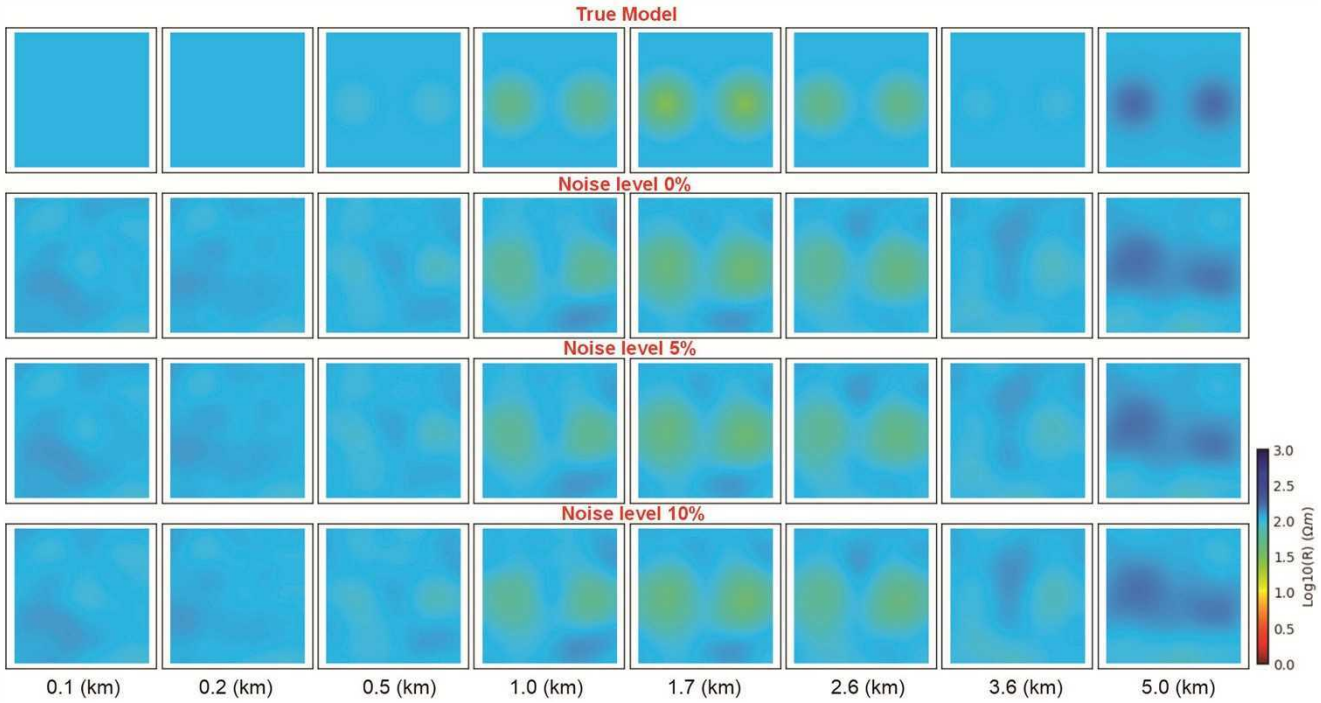


Figure 14. The horizontal slice of model and predicted results with different noise levels (0%, 5% and 10%).

Table 1. RMSE and R^2 of predicted model with different noise levels.

Noise Level	0%	5%	10%
RMSE	0.04200	0.04266	0.04384
R^2	0.89892	0.89593	0.89291

3.3. Real-Data Application

A field-data application of this 3D DL-based inversion algorithm was implemented to validate its reliability. MT data were acquired from the Mount Meager volcanic area, British Columbia, Canada, in 2019 and 2020; this area is part of a geothermal energy assessment project aimed at improving our understanding of the magmatic and hydrothermal conduit system beneath Mount Meager (see Figure 15) [2,3]. The purpose of the MT survey was to comprehend the basic geological structure of the surroundings and to locate the conductive zones that are related to either the magma chamber or upwelling conductive thermal fluids. The MT field data were acquired with a Phoenix MTU-5A system to record variations in the geomagnetic field and its induced electric field. The broadband magnetotelluric (BBMT) time-series data were processed into the frequency domain by using a robust estimation of the geomagnetic transfer functions [3,18].

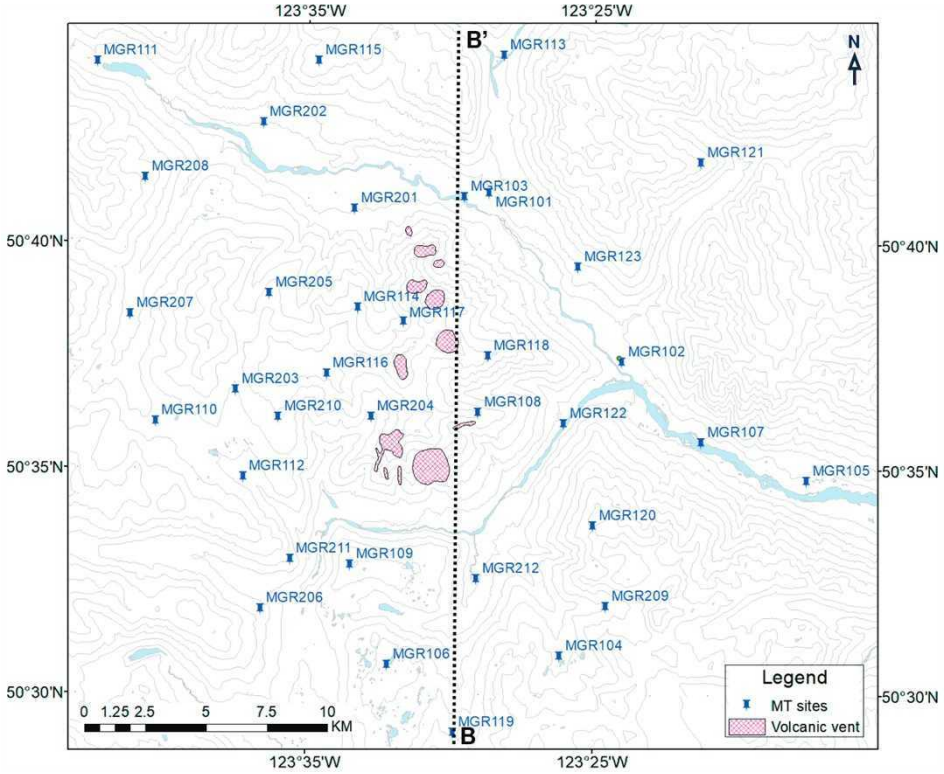


Figure 15. MT site distribution with cross section B-B′ location in Mount Meager study area, British Columbia, Canada.

The calculated impedance components and tipper data from 35 stations with a frequency range of 100–0.001 Hz were used in model testing. These stations are marked with blue pins in Figure 15. The 3D resistivity model in this area was 19 × 19 × 25 with a 2.0 km cell size horizontally. The vertical cell size of the top grid was 20 m, and the size increased layer by layer to a total depth of about 30 km. The same hyperparameters were used for model training and validation, and the model training process took about 40 h. Based on the 3D inversion results with the trained DL model, the resistivity model was sliced into profiles which crossed the main structure of the study area. Figure 16 shows horizontal slices at different depths, which reveal a resistivity anomaly (purple dashed-line square) in the depth range 5.0 to 8.0 km and display the trend of the conductor interpreted as a potential magma body. In general, the prediction of DL inversion is consistent with the ModEM inversion result in a previous study [3].

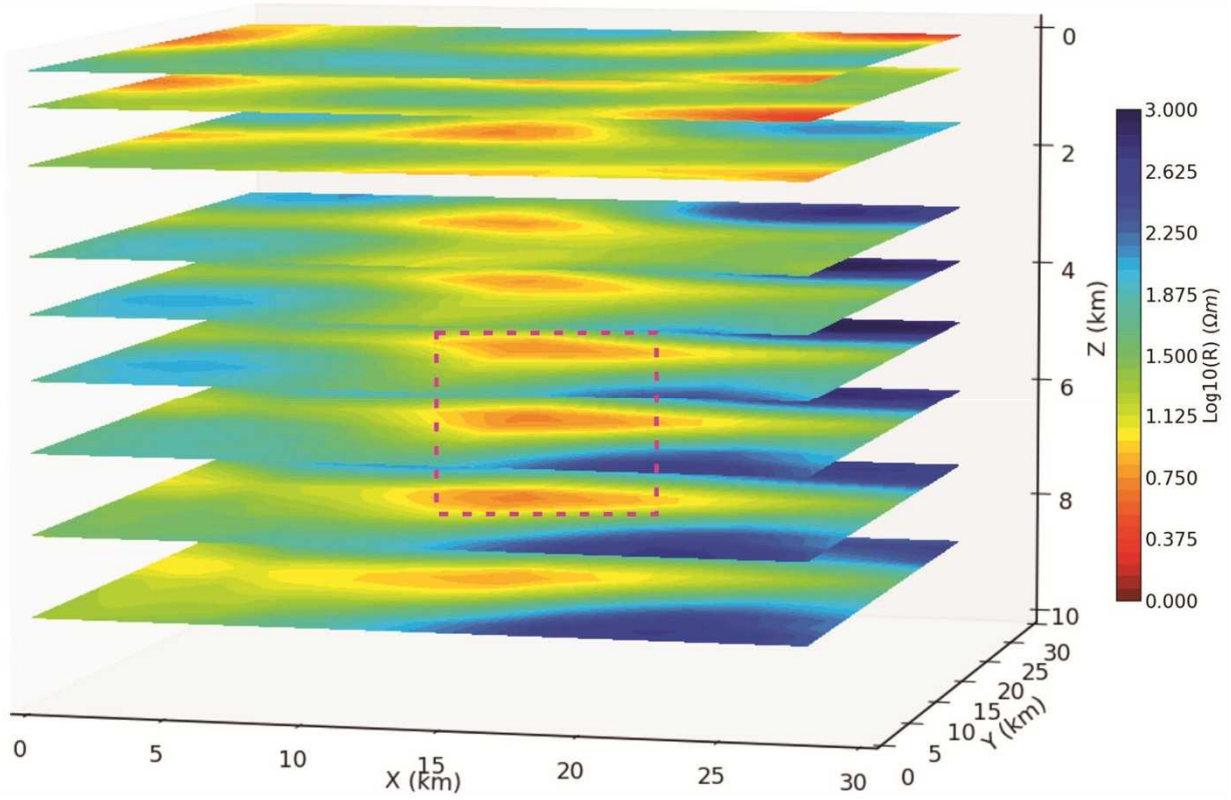


Figure 16. Horizontal sliced iso-resistivity maps at different depths between 0.5 and 10 km (the purple dashed line indicates the main resistivity anomaly).

For a further comparison with iterative inversion, the plots in Figure 17a,c,e display both the horizontal profiles (depths of 6.0 and 7.0 km) and a vertical profile (along B-B′ in Figure 15) of the predicted 3D resistivity model. This inversion determines the high conductive zone below 5.0 km, which shows satisfactory results and is in good agreement with the 3D iterative inversion method (Figure 17b,d,f). The identified shallow conductive features with both inversion methods (Figure 17e,f) are interpreted as a hydrothermal system, where the deep conductive zone is interpreted as a magma body [31]. Despite a larger cell size being used to decrease computation time in the DL model and forward modelling, without topography, the predicted results still reflect the position and trend of the conductive body. Due to the effect of data noise and topography, one can see a shift in the anomaly center and an artifact in the near-surface area by making a comparison with the results of the traditional iterative inversion. Overall, the DL inversion results show the location of the conductivity anomalies, and the retrieved resistivity distribution represents an actual case for geological interpretation, which displays similar interpretated structures at Mount Meager [2]. Because this study is focused on methodological development and experience and some real MT data in the Meager Creek area are not available, further detailed geological interpretation and analysis cannot be provided here.

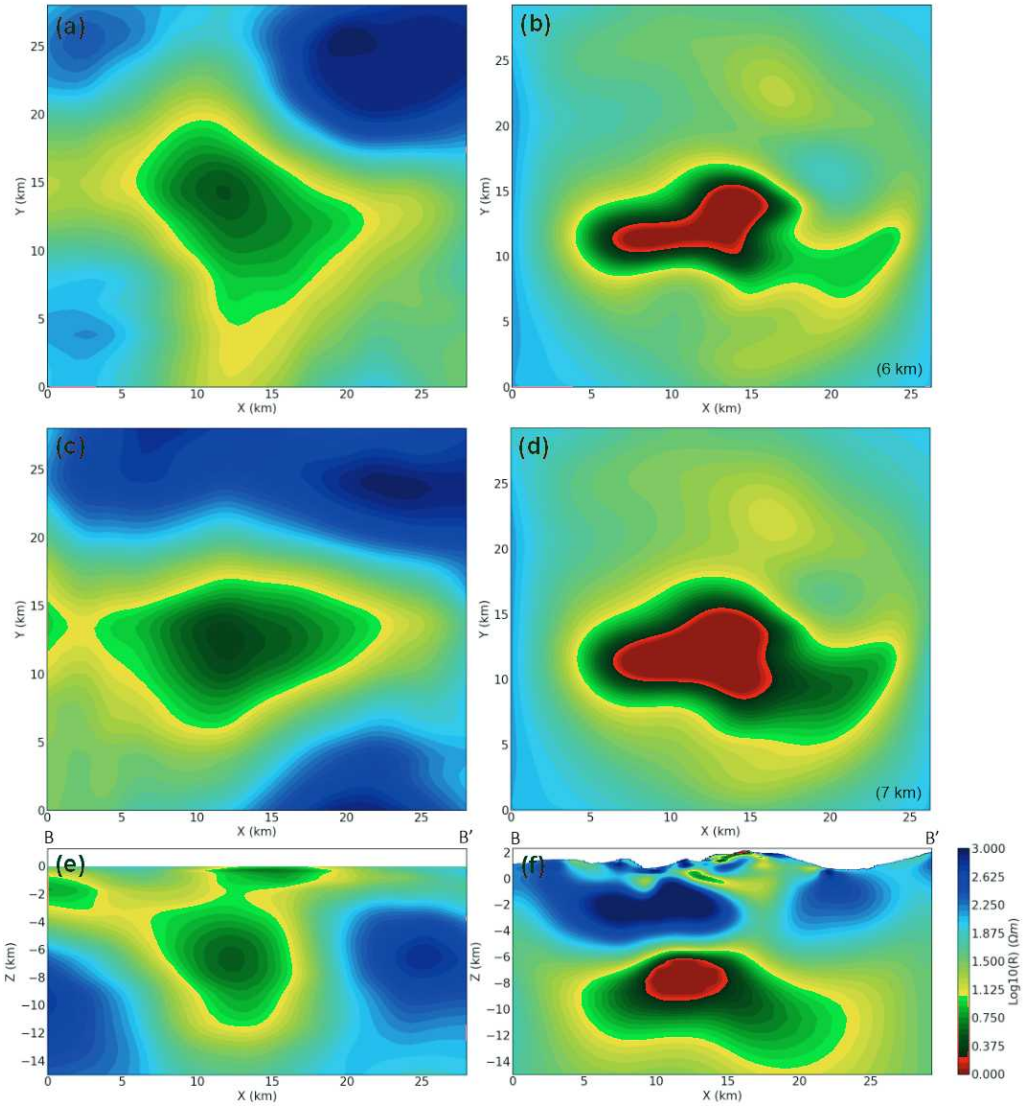


Figure 17. Slices of the inversion results with DL-based inversion and ModEM iterative inversion. (**a**–**d**) Horizontal slices at depths of 6.0 and 7.0 km; (**e**,**f**) vertical slices along profile B-B′ in Figure 15; (**a**,**c**,**e**) 3D DL-based inversion; (**b**,**d**,**f**) ModEM iterative inversion.

4. Discussion

The computation time of the MT 3D CNN inversion workflow is mainly attributed to generating the synthetic datasets and training/validating the model weight parameters. Compared with traditional inversion methods, this algorithm needs data pairs for model training, which is costly in terms of time. However, the prediction period is short once a neural network model is created. In general, data generation with random walk improves the generalized coverage of the trained model, so using relatively fewer datasets can still produce a stable model, which reduces the computation time for a 3D inversion problem. Only one GPU is available for data preparation and model training in this study. Multiple GPUs will speed up dataset generation and model training, which will make forward modelling with a fine-mesh grid possible.

The metric contour map from the grid search with two parameters displays a minimum area for choosing the optimized parameters. The applications of synthetic models and real field data demonstrated the accountability of calculating the resistivity distribution, displaying the advantage of this method. Noise testing further verified the reliability of the proposed method (Figure 14). As a data-driven mathematical method, the overall

test results of this study demonstrate that the CNN architecture can be used to simulate physical laws and simplify the MT inversion procedure.

Despite the advantages of deep learning in geophysical MT inversion, some limitations and challenges need to be noted.

Firstly, there is the biased coverage of the generated training datasets. Even when a random walk generator was used for creating training datasets, the number of resistivity models and their simulated response was limited by the computational capacity. This limits the minimum cell size and the total number of mesh grids, which leads to potential inaccuracies in model training and prediction. In this study, we could only test a mesh grid of $19 \times 19 \times 25$. Generating more diversified training datasets by combining a prior-information-based generator will improve the model prediction, but this method will reduce the generalization of the trained neural network.

The second limitation is the complexity of the neural network model. Although U-Net has been popularly used in image segmentation, model complexity is still not enough in physical law simulation. The available datasets and computational power, e.g., #GPUs, need be considered when designing a complex neural network model. Additional investigation by testing a deeper complex neural network framework is necessary.

The third limitation is stability with topographic real noised data. For real data, adding topography to forward modelling is operationally possible. The computational capacity limits the mesh gridding of the topography. With the current hardware capacity, adding the topography effect to the simulation might lead to simulated response bias and model prediction inaccuracies. Obtaining a better resistivity range based on different sources, e.g., core analysis and well logs, will make the predicted result more comparable with iterative inversion.

These limitations mean that DL-based inversion lacks the capability to recover the same level of detail as ModEM inversion. Further research on optimizing this proposed inversion algorithm will enhance the reliability and performance of the convolutional neural network application in MT data inversion. Increasing model generalization and enhancing the noise tolerance capability of the model are crucial to ensuring the inversion model's applicability.

5. Conclusions

Geophysical inversion is a significant and challenging step in deriving an interpretable physical model. The main purpose of this study is to improve and verify the performance of DL-based 3D inversion for MT data. The optimal parameters of a neural network framework were investigated to establish connections between MT impedance data and subsurface resistivity properties. The 3D U-Net framework displays its advantage in simulating physical laws. By using the random walk technique for data generation, we can create relative fewer datasets for model training and validation, which decreases computational cost and maintains the stability of the network model. By comparing the effect of model training parameters, the best optimal neural network parameters can be selected via parameter tuning for optimizing the weight parameters. This experience demonstrated that parameter tuning is an essential step to obtain a set of optimized parameters for model training. Lastly, the practical application of the 3D inversion method for MT synthetic and real data verified its reliability and potential. In future research, the topographic effect needs be considered with a smaller-cell-size model, and parameter tuning will be performed with a higher-dimensional search of multiple network parameters. This will produce integrated hyperparameters to improve the prediction accuracy of the trained network model.

Author Contributions: X.L.: methodology, software and original draft preparation; J.A.C.: resources, data curation, validation, and review and editing; V.T.: data curation, validation, and review and editing; S.E.G.: resources, review and editing, and project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Garibaldi Geothermal Energy Project (funded by NRCan and Geoscience BC) under Geoscience for New Energy Supply (GNES/GeoEnergy) and the Critical Mineral Geoscience Data (CMGD) program of Natural Resources Canada. This research received no external funding.

Data Availability Statement: The authors have not been granted permission to share the original MT data from the Mount Meager study area.

Acknowledgments: The authors are grateful to Martyn Unsworth and Cedar Hanneson from University of Alberta for providing the MT field data and ModEM output file and to Research Scientist Rebecca Swinscoe at Geological Survey of Canada for reviewing and discussions. The authors are grateful to the journal editors and reviewers for their rigorous reviews, with helpful suggestions and comments. The outcome represents a contribution of the GeoEnergy program. The NRCan Contribution number is 20230357.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Craven, J.A.; Farquharson, C.G.; Mackie, R.L.; Siripunvaraporn, W.; Tuncer, V.; Unsworth, M.J. A comparison of two- and three-dimensional modelling of audio-magnetotelluric data collected at the world's richest uranium mine, Saskatchewan, Canada. In Proceedings of the 18th International Workshop on Electromagnetic Induction in the Earth, El Vendrell, Spain, 17–23 September 2006.
2. Grasby, S.E.; Ansari, S.M.; Bryant, R.; Calahorrano-DiPatre, A.; Chen, Z.; Craven, J.A.; Dettmer, J.; Gilbert, H.; Hanneson, C.; Harris, M.; et al. Garibaldi Geothermal Energy Project Mount Meager 2019—Field Report. Geoscience BC Report 2020-09; Natural Resources Canada/CMSS/Information Management: Ottawa, ON, Canada, 2020; p. 153.
3. Hanneson, C.; Unsworth, M.J. Magnetotelluric imaging of the magmatic and geothermal systems beneath Mount Meager, southwestern Canada. *Can. J. Earth Sci.* **2023**. *e-First*. [CrossRef]
4. Zhdanov, M.S. *Geophysical Inverse Theory and Regularization Problems*; Elsevier Science B.V.: Amsterdam, The Netherlands, 2002; pp. 214–229.
5. Kim, Y.; Nakata, N. Geophysical inversion versus machine learning in inverse problems. *Lead. Edge* **2018**, *37*, 894–901. [CrossRef]
6. Russell, B. Machine learning and geophysical inversion; a numerical study. *Lead. Edge* **2019**, *38*, 512–519. [CrossRef]
7. Das, V.; Pollack, A.; Wollner, U.; Mukerji, T. Convolutional neural network for seismic impedance inversion. *Geophysics* **2019**, *84*, R869–R880. [CrossRef]
8. Puzyrev, V. Deep learning electromagnetic inversion with convolutional neural networks. *Geophys. J. Int.* **2019**, *218*, 817–832. [CrossRef]
9. Moghadas, D. One-dimensional deep learning inversion of electromagnetic induction data using convolutional neural network. *Geophys. J. Int.* **2020**, *222*, 247–259. [CrossRef]
10. Guo, R.; Li, M.; Yang, F.; Xu, S.; Abubakar, A. Application of supervised descent method for 2D magnetotelluric data inversion. *Geophysics* **2020**, *85*, WA53–WA65. [CrossRef]
11. Hu, Z.; Liu, S.; Hu, X.; Fu, L.; Qu, J.; Wang, H.; Chen, Q. Inversion of magnetic data using deep neural networks. *Phys. Earth Planet. Inter.* **2021**, *311*, 106653. [CrossRef]
12. Wang, Y.F.; Zhang, Y.J.; Fu, L.H.; Li, H.W. Three-dimensional gravity inversion based on 3D U-Net. *Appl. Geophys.* **2021**, *18*, 451–460.
13. Liu, X.; Craven, J.A.; Tschirhart, V. Retrieval of Subsurface Resistivity from Magnetotelluric Data Using a Deep-Learning-Based Inversion Technique. *Minerals* **2023**, *13*, 461. [CrossRef]
14. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *arXiv* **2014**, arXiv:1411.4038 https://arxiv.org/abs/1411.4038.
15. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9351, pp. 234–241.
16. Chen, Z.; Liu, X.; Yang, J.; Little, E.; Zhou, Y. Deep learning-based method for SEM image segmentation in mineral characterization, an example from Duvernay Shale samples in Western Canada Sedimentary Basin. *Comput. Geosci.* **2020**, *138*, 104450. [CrossRef]
17. Ward, S.H.; Hohmann, G.W. Electromagnetic Theory for Geophysical Applications. *Electromagn. Methods Appl. Geophys.* **1987**, *1*, 201.
18. Egbert, G.D.; Booker, J.R. Robust estimation of geomagnetic transfer functions. *Geophys. J. R. Astron. Soc.* **1986**, *87*, 173–194. [CrossRef]
19. Mackie, R.L.; Madden, T.R. Three-dimensional magnetotelluric inversion using conjugate gradients. *Geophys. J. Int.* **1993**, *115*, 215–219. [CrossRef]
20. Farquharson, C.G.; Craven, J.A. Three-dimensional inversion of magnetotelluric data for mineral exploration: An example from the McArthur River uranium deposit, Saskatchewan, Canada. *J. Appl. Geophys.* **2009**, *68*, 450–458. [CrossRef]

21. Egbert, G.D.; Kelbert, A. Computational recipes for electromagnetic inverse problems. *Geophys. J. Int.* **2012**, *189*, 251–267. [CrossRef]
22. Constable, S.C.; Parker, R.L.; Constable, C.G. Occam's inversion: A practical algorithm for generating smooth models from electromagnetic sounding data. *Geophysics* **1987**, *52*, 289–300. [CrossRef]
23. Çiçek, Ö.; Abdulkadir, A.; Lienkamp, S.S.; Brox, T.; Ronneberger, O. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. *arXiv* **2016**. [CrossRef]
24. Hochreiter, S. The Vanishing Gradient Problem during Learning Recurrent Neural Nets and Problem Solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **1998**, *6*, 107–116. [CrossRef]
25. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, 20 September 2018*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11045, pp. 3–11. [CrossRef]
26. Krieger, L.; Peacock, J.R. MTpy: A Python toolbox for magnetotellurics. *Comput. Geosci.* **2014**, *72*, 167–175. [CrossRef]
27. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
28. Keras (Webpage). 2015. Available online: https://keras.io (accessed on 1 July 2022).
29. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; Volume 16, pp. 265–283.
30. Kelbert, A.; Meqbel, N.; Egbert, G.D.; Tandon, K. ModEM: A modular system for inversion of electromagnetic geophysical data. *Comput. Geosci.* **2014**, *66*, 40–53. [CrossRef]
31. Jones, A.G.; Dumas, I. Electromagnetic images of a volcanic zone. *Phys. Earth Planet. Inter.* **1993**, *81*, 289–314. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

mathematics

MDPI

Article

A Domain Generation Diagnosis Framework for Unseen Conditions Based on Adaptive Feature Fusion and Augmentation

Tong Zhang [1], Haowen Chen [1], Xianqun Mao [1,*], Xin Zhu [1] and Lefei Xu [2]

[1] Marine Design and Research Institute of China, China State Ship-Building Corporation Limited, Shanghai 200011, China; tonguezhang@163.com (T.Z.); asdchehaowen@126.com (H.C.); zx1987_china@126.com (X.Z.)
[2] School of Traffic & Transportation Engineering, Central South University, Changsha 410004, China; dawnsky@csu.edu.cn
* Correspondence: maooxxqq@163.com

Abstract: Emerging deep learning-based fault diagnosis methods have advanced in the current industrial scenarios of various working conditions. However, the prerequisite of obtaining target data in advance limits the application of these models to practical engineering scenarios. To address the challenge of fault diagnosis under unseen working conditions, a domain generation framework for unseen conditions fault diagnosis is proposed, which consists of an Adaptive Feature Fusion Domain Generation Network (AFFN) and a Mix-up Augmentation Method (MAM) for both the data and domain spaces. AFFN is utilized to fuse domain-invariant and domain-specific representations to improve the model's generalization performance. MAM enhances the model's exploration ability for unseen domain boundaries. The diagnostic framework with AFFN and MAM can effectively learn more discriminative features from multiple source domains to perform different generalization tasks for unseen working loads and machines. The feasibility of the proposed unseen conditions diagnostic framework is validated on the SDUST and PU datasets and achieved peak diagnostic accuracies of 94.15% and 93.27%, respectively.

Keywords: intelligence fault diagnosis; domain generation; unseen working conditions; feature fusion

MSC: 68T07

1. Introduction

As industry progresses towards informatization and intelligence, advanced intelligent fault diagnosis (IFD) models have garnered more attention [1]. Currently, various IFD models have demonstrated satisfactory performance, helping to reduce unplanned downtime and enable predictive maintenance [2]. These models are highly appreciated for their autonomy and high accuracy, which greatly enhance the efficiency of equipment maintenance and management [3].

However, there can be limitations on the use of intelligent methods for practical industrial applications. In order to achieve a high accuracy of fault diagnosis, these traditional IFD methods often require a large number of labeled data which have the same distribution as the testing data [4,5]. Owing to variations in operational conditions and interference from environmental noise, rotating machines frequently operate under diverse working scenarios, significantly altering the distribution of collected data. Consequently, when confronted with practical industrial issues, there is often an insufficiency of labeled data to fulfill the model's training requirements [6]. Furthermore, given the high cost of human labor, it is impractical to gather a vast amount of fault samples

and generate corresponding ground-truth labels under all potential working conditions for model training [7]. Hence, cross-domain diagnosis poses a significant challenge in developing reliable and precise IFD methods.

Recently, significant efforts have been invested in developing domain adaptation (DA) models that strive to overcome the domain shift issues stemming from diverse conditions and foster the successful deployment of IFD models with limited labeled data. The central concept of DA methodologies revolves around harnessing the discrepancy between the training source domain and the testing target domain by aligning the distributions of fault features in a higher-dimensional space and extracting domain-invariant features for diagnostic purposes [8]. The differences between traditional data-driven fault diagnosis and cross-domain intelligent fault diagnosis are shown in Figure 1a,b [9]. These models aim to generalize the knowledge learned from the limited data collected from a single machine (referred to as the source domain) to enable cross-domain fault diagnosis on varying machines or the same machine operating under differing conditions (designated as the target domain). Lu et al. [10] first applied a model-based learning technique to solve DA issues for fault diagnosis under different working conditions. Shao et al. [11] proposed an adversarial domain adaption method based on deep transfer learning for rolling elements bearing insufficient labeled data issues. Xiong et al. [12] utilized a Wasserstein gradient-penalty generative adversarial network with a deep auto-encoder for intelligent fault diagnosis. Zhao and Shen [13] implemented a local class cluster module to explore the domain-invariant representation space and obtain discriminative representation structures to enhance the model's generation ability. Xu et al. [14] used Support Vector Data Description (SVDD) as the unknown fault identification method, and optimized the relevant parameters using a Particle Swarm Optimization (PSO) algorithm. Liao et al. [8] constructed a deep semi-supervised domain generalization network (DSDGN) which employed causal learning techniques to acquire causal invariant fault information across different machines. Despite the potential of DA-based IFD models, there persist obstacles in their application to contemporary industrial processes. The core purpose of these models is to minimize the divergence between training and testing data through techniques such as distance metrics and subspace alignment. It is evident that obtaining adequate data from the target domain is crucial for guiding the distribution adaptation process and enabling effective knowledge transfer during model training [15]. However, in many industrial scenarios, acquiring data from target machines or operating conditions prior to their occurrence is not feasible.

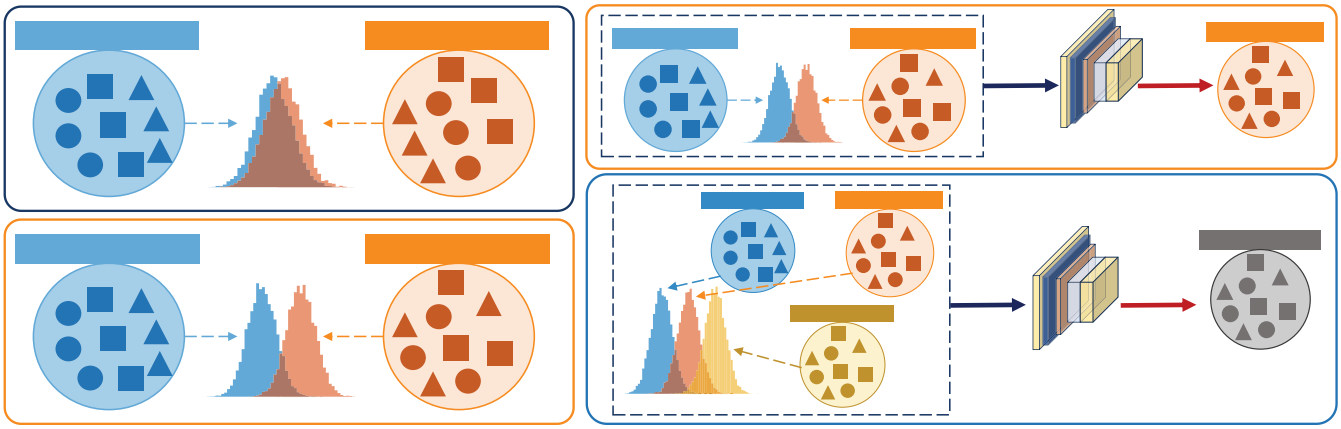


Figure 1. Illustration of (**a**) traditional data-driven fault diagnosis and (**b**) cross-domain intelligent fault diagnosis. The values inside the symbols denote the class label for each sample. (**c**) Domain adaption problem definition and (**d**) domain generation problem definition.

Domain generation (DG) is a method of transfer learning which endeavors to generalize a trained model, leveraging a diverse array of source datasets, to accommodate unseen target domains. It focuses on extracting key domain-invariant feature representa-

tions from specific source domains that are both discriminative across various classes and resilient to domain shifts, thereby enhancing the robustness and adaptability of data-driven approaches to varying conditions [16]. Domain generation-based models can be categorized as data manipulation [17], representation learning [18], and learning strategies [19]. Li et al. [20] pioneeringly integrated domain generalization into the realm of intelligent fault diagnosis, leveraging adversarial training and metric learning techniques to extract generalized features that transcend specific domains. In a parallel effort, Han, et al. [21] advocated minimizing the triplet loss as a means to foster intra-class cohesion and inter-class discernibility, thereby bolstering the generalization prowess of fault diagnosis models. Chen et al. [22] devised an innovative domain-regressive framework that centered around learning domain-agnostic representations, tailored specifically for rotating machinery fault diagnosis. Zheng et al. [23] innovatively fused prior diagnostic knowledge with a deep domain generalization network, augmenting it with an instance-based discriminative loss. This integration yielded a robust framework capable of performing cross-domain fault diagnosis across disparate bearing datasets, showcasing remarkable adaptability and generalization. Although these DG methods do not require prior knowledge of the target domain, one significant limitation persists which is that DG methods require a massive and varied collection of samples from multiple source domains.

In practical industrial applications, acquiring useful samples from varying working conditions can be challenging, often restricting utilization to data sampled under a single working condition. Meanwhile, the absence of target data during the training phase inherently elevates the likelihood of overfitting to the source data, posing a significant challenge. Consequently, DG-powered IFD models ought to prioritize strategies that enhance the integration of data acquired under diverse operational conditions into a cohesive, unified representation. This approach aims to mitigate the risk of overfitting and enhance the models' ability to generalize across varying scenarios.

To this end, we proposed a novel domain generation framework for unseen condition fault diagnosis which comprises an Adaptive Feature Fusion Domain Generation Network (AFFN) and a Mix-Up Augmentation Method (MAM). The key of the AFFN and MAM is to learn both domain-specific representation and domain-invariant representation and fuse them dynamically in a unified deep neural network. The main contributions of this study are summarized as follows:

1. A domain generation-based fault diagnosis framework for unseen conditions is proposed to complement the online health management operations of industrial applications for robust and generalized fault recognition.
2. A novel domain generation network, the Adaptive Feature Fusion Domain Generation Network (AFFN), is proposed, which can learn both domain-invariant and domain-specific deep representations to enhance the generalization capability of the framework.
3. A well-designed data Mix-up Augmentation Method (MAM) is integrated into the framework, which fully utilizes the existing data to extend the model's exploration and learning boundaries to improve the fit of the unseen domain.

The organization of the remainder of this paper is outlined as follows: Section 2 provides an introduction to pertinent techniques and background knowledge. Section 3 delves into the structure and specifics of the proposed diagnostic framework for unseen conditions. Section 4 presents the experimental validation and real-world application in detail. Finally, Section 5 brings this paper to a conclusion.

2. Preliminaries

2.1. Domain Generalization Problem Definition

The unseen domain diagnostic targeted to be solved in this paper can be regarded as a cross-domain classification; with this in mind, the utilization of DG has emerged as a prominent research trend. As depicted in Figure 1c,d, the training of the DG diagnostic model utilizes multiple source domain datasets and involves subsequently applying it directly to target diagnostic tasks. In research manuscripts detailing the deposition of

extensive datasets into publicly accessible databases, it is imperative to clearly indicate the repository where the data reside and furnish the corresponding accession numbers. If, at the point of submission, the accession numbers are yet to be secured, authors should make a note of their intention to provide these details during the review process. Ultimately, it is mandatory that these numbers are made available prior to the manuscript's publication, ensuring transparency and accessibility for the scientific community.

Let $x = \{x_1, x_2, \ldots, x_n\}$ be a set of observations from the feature space $\mathcal{X}$ and $y = \{y_1, y_2, \ldots, y_n\}$ be a set of n corresponding class labels from the output space $\mathcal{Y}$, with P_{XY} denoting the joint distributions. Herein, the source component is denoted as "$\mathbb{S}$" and the target component is donated as "$\mathbb{T}$". The M source domain is defined as $\mathcal{D}^{\mathbb{S}} = \left\{\mathcal{D}_{\mathrm{m}}^{\mathbb{S}}\right\}_{m=1}^{M}$, where $\mathcal{D}_{\mathrm{m}}^{\mathbb{S}} = \left\{\left(x_i^{\mathbb{S},\mathrm{m}}, y_i^{\mathbb{S},\mathrm{m}}\right)\right\}_{i=1}^{N_{s,m}}$ denotes the m^{th} domain and the labeled observations in all source domains are sufficiently available for model training. The joint distribution of the source domain is different, which is $P_{XY}^{m_1} \neq P_{XY}^{m_2}$ [6]. Similarly, the unseen and unlabeled target domain in the training stage is denoted as $\mathcal{D}^{\mathbb{T}} = \left\{\mathcal{D}_{\mathrm{m}}^{\mathbb{T}}\right\}_{m=1}^{M}$; apparently, the joint distribution of the target domain is different from any other source domain, namely $P_{XY}^{\mathbb{T}} \neq P_{XY}^{\mathbb{S},\mathrm{m}}$. Hence, the goal of domain generalization is to learn a robust and generalizable predictive function $\langle : \mathcal{X} \to \mathcal{Y}$ from M training source domains to achieve a minimum prediction error on an unseen test domain $\mathcal{D}^{\mathbb{T},test}$(i.e., $\mathcal{D}^{\mathbb{T},test}$ cannot be accessed in training).

$$\min_{h} \mathbb{E}_{(x,y)\in\mathcal{D}^{\mathbb{T},test}}[\updownarrow(\langle(\mathcal{X}), \mathcal{Y})] \tag{1}$$

where $\mathbb{E}$ is the expectation and $\updownarrow(\cdot,\cdot)$ is the loss function.

2.2. Data Augmentation

It is widely recognized that by increasing the variety and volume of training data, the generalization performance of intelligent models can be enhanced. When dealing with a finite set of training data, data augmentation stands as one of the most prevalent and straightforward methods to enrich the diversity of the training dataset.

When carrying out classification tasks, it is common practice to define the neighborhood or vicinity of a sample as the set encompassing its horizontal reflections or linear transformations. Notably, data augmentation has consistently demonstrated its ability to enhance the generalization capabilities of models. Zhang et al. [24] introduced a mix-up learning methodology that involves combining two observations in a linear fashion, smoothing the transition between decision boundaries. This approach involves training a neural network on convex combinations of pairs of examples and their corresponding labels.

$$\mu\left(\widetilde{x}, \widetilde{y}\right) = \frac{1}{n}\sum_{j}^{n} \mathbb{E}_{\lambda}\left[\delta\left(\widetilde{x} = \varphi(x_i, x_j), \widetilde{y} = \varphi\left(y_i, y_j\right)\right)\right] \tag{2}$$

where $\varphi(a,b) = \lambda a + (1-\lambda)b$, and $\lambda \sim Beta(\alpha,\alpha)$ for $\alpha \in (0,\infty)$. $\delta\left(\widetilde{x}, \widetilde{y}\right)$ is a Dirac mass centered at $\left(\widetilde{x}, \widetilde{y}\right)$. The mix-up hyper-parameter α controls the strength of interpolation between feature–target pairs. Also, the mix-up reduces the memorization of corrupt labels, increases the robustness to unseen examples, and stabilizes the training of generalization networks.

3. Methods

3.1. Unseen Conditions Diagnostic Framework

The proposed unseen conditions diagnostic framework is as depicted in Figure 2, and mainly consists of three parts: data augmentation, adaptive feature network training, and online diagnostic application. The utilized MAM method is for data and feature augmentation, including feature augmentation, label fusion, and domain mix-up. The

constructed adaptive feature network aims to is to learn both domain-specific representation and domain-invariant representation and fuse them.

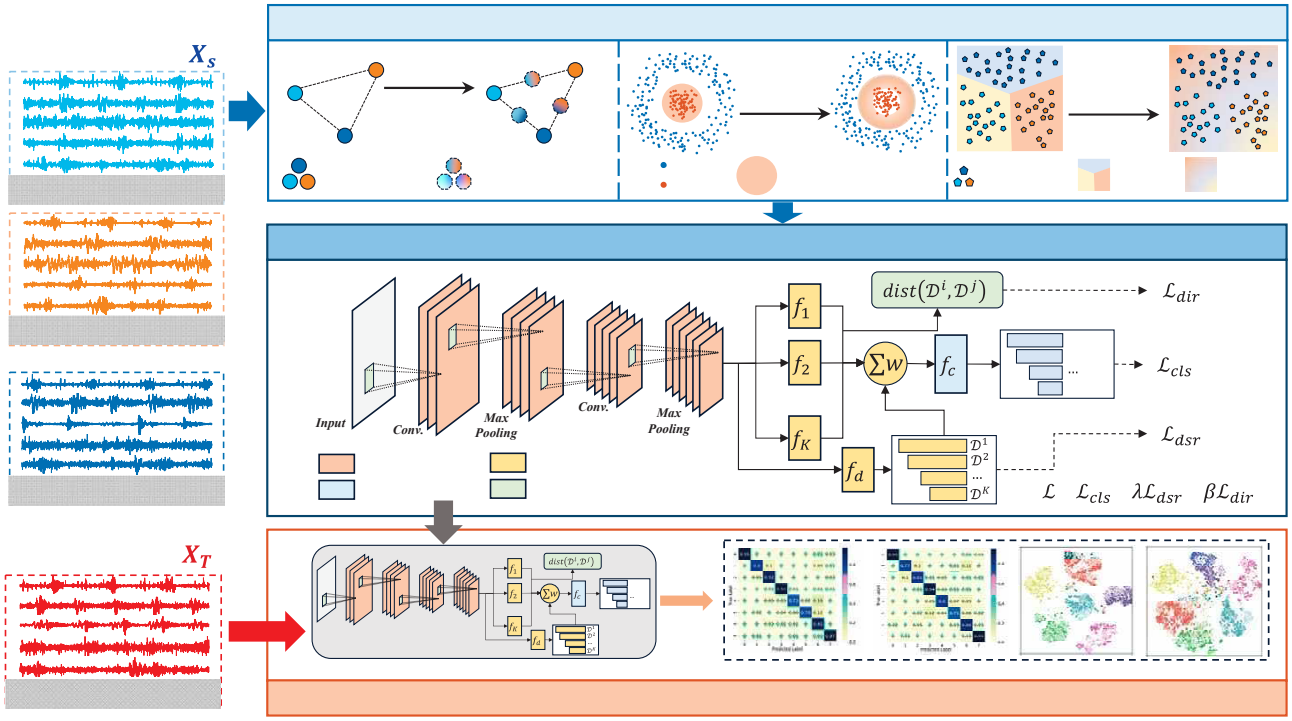


Figure 2. Structure of the proposed unseen conditions diagnostic framework.

3.1.1. Data Augmentation and Fusion

The mix-up method was proposed by Zhang et al. [24] to generate a new feature–label pair based on the weighted sum of features and the corresponding one-hot label vector. However, the origin proposed data augmentation method only takes the data level into consideration, resulting in limited volume and features. In this paper, the mix-up method has been extended to label the space and domain space. Thus, the lack of information on different fault classes and domains is compensated for and the diversity of the category and domain space is increased.

As defined previously, 0 x_i is the observations from the feature space $\mathcal{X}$, y_i is the corresponding fault classes in the label domain $\mathcal{Y}$, and d_i is the domain labels in $\mathcal{D}$. For all the samples in the source domain (i.e., $\mathcal{D} = \{\mathcal{D}_{\mathrm{m}}^{\mathbb{S}}\}_{m=1}^{M}$), data augmentation can be realized by the following equations:

$$\begin{cases} x_i^g = \lambda x_j + (1-\lambda)x_k \\ y_i^g = \lambda y_j + (1-\lambda)y_k \\ d_i^g = \lambda d_j + (1-\lambda)d_k \end{cases} \tag{3}$$

where $\lambda \sim Beta(\alpha, \alpha)$ for $\alpha \in (0, \infty)$, and the original x_j and x_k are randomly selected form the overall set of source domains; moreover, the class label y_i^g and domain label d_i^g are no longer one-hot vectors. The normalized distribution of $Beta(\alpha, \alpha)$ is generally expressed as

$$Beta(\lambda \mid a, b) = \frac{\Gamma(2\alpha)}{2\Gamma(\alpha)}[\lambda(1-\lambda)]^{\alpha-1} \tag{4}$$

In the domain and label fusion process, the empirical distribution is utilized to measure augmentation quality, which can be expressed as

$$P_\delta(x, y, d) = \frac{1}{n}\sum_{i=1}^{n}\delta(x = x_i, y = y_i, d = d_i) \tag{5}$$

where $\delta(x = x_i, y = y_i, d = d_i)$ is the Dirac mass centered at (x_i, y_i, d_i), and n is the number of all augmented samples. Hence, in order to achieve the goal of label fusion and domain mix-up, the expectation of the extended generalization can be concluded as

$$\begin{cases} min\ E_\lambda\left[\delta\left(x_i^g = \varphi(x_j, x_k, \lambda), y_i^g = \varphi(y_j, y_k, \lambda)\right)\right] \\ max\ E_\lambda\left[\delta\left(x_i^g = \varphi(x_j, x_k, \lambda), d_i^g = \varphi(d_j, d_k, \lambda)\right)\right] \end{cases} \tag{6}$$

where $\varphi(a, b, \lambda) = \lambda a + (1 - \lambda)b$, and $\lambda \sim Beta(\alpha, \alpha)$; the equation above indicates that, at the minimum discrepancy of augmented features for fault features, fault labels can be well fitted, and at the maximum of domain discrepancy for unseen domain exploration, domain boundaries can be fused and extended.

3.1.2. Adaptive Feature Fusion Network

The Adaptive Feature Fusion Network (AFFN) excels in its capacity to concurrently harness both domain-invariant and domain-specific features, marking a pivotal advancement. In contrast, the domain-specific representation learning module delves into capturing the unique nuances and distinguishing attributes of each source domain, thereby augmenting the network's generalization prowess. Conceptually, the AFFN is meticulously crafted with four distinct yet interconnected modules: the feature extraction module (embodied by orange blocks), the fault classification module (represented by blue blocks), the domain-specific representation learning module (highlighted in yellow), and the domain-invariant learning module (depicted in green), as elegantly illustrated in Figure 2.

The learning objective of AFFN can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_{cls} + \beta\mathcal{L}_{dsr} + \gamma\mathcal{L}_{dir} \tag{7}$$

where $\mathcal{L}_{cls}$ is the classification loss, $\mathcal{L}_{dsr}$ is the loss of domain-specific representation learning, and $\mathcal{L}_{dir}$ is the loss of domain-invariant representation learning. β and γ are the tradeoff hyperparameters.

(1) The feature extraction module serves the purpose of distilling features from the row sensor data. Two CNN layers accompanied by max-pooling operations are employed to extract these features. Notably, these layers are designed to be shared across all training domains, thus minimizing the number of parameters. For classification, the cross-entropy is taken as the classification loss:

$$\mathcal{L}_{cls} = -\frac{1}{N}\sum_{i=1}^{N} y_i log P(y_i \mid x_i) \tag{8}$$

where $N = \sum_{k=1}^{K}, n_k$ is the number of samples from all the training domains.

(2) The domain-specific representation learning module specializes in acquiring features that are tailored to each domain; hence, it is not shared and remains unique to each domain. Following feature extraction, K fully connected (FC) layers are implemented for each domain. Additionally, a weighting function has been devised to integrate the specific information of each domain. Given new test data x, their feature z is formulated as

$$z = \sum_{k=1}^{K} w_k f_k(f_e(x)) \tag{9}$$

where w_k ($w_k > 0$ and $\sum_{k=1}^{K} w_k = 1$) is the weight on domain $\mathcal{D}^k$, indicating the similarity between the data in domain $\mathcal{D}^k$ and the input data under unseen conditions. f_k is the feature learning function of domain $\mathcal{D}^k$ and f_e is the shared feature extraction function.

The combination of f_k and f_e guarantees that the initial layers tend to acquire general features, whereas the later layers gravitate toward domain-tailored features. During the

training phase, the domain label $d_k \equiv k$ for each sample is known beforehand, thus allowing for the computation of the domain-specific loss for each domain k to be concluded as

$$\mathcal{L}_{dsr}^{k} = \frac{1}{n_k} \sum_{i=1}^{n_k} \updownarrow (f_d(f_e(x_i)), d_k) \tag{10}$$

Then, the total domain-specific loss can be obtained by averaging the losses on all domains:

$$\mathcal{L}_{dsr} = \frac{1}{K \times n_k} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \updownarrow (f_d(f_e(x_i)), d_k) \tag{11}$$

3.2. Training and Application

During training, the data augmentation and fusion method enrich the data, label, and domain spaces. The feature extractor captures lower-level features and feeds them to domain-specific branches. The domain classifier uses these features to fuse domain-specific characteristics and learns domain-invariant representations. These fused features then enable activity classification.

For application, the model parameters are fixed, and domain-specific and invariant representations are derived for an unseen target domain. Without knowing the target domain's label, the domain classifier's output indicates the similarity between the target and source domains. The label loss in the source domain and the domain fusion loss are calculated as in every epoch. The optimizer Adam is utilized to optimize the parameters of the feature extractor and, as part of the parameters of domain classifiers, the learning rate is set as 0.001. The explicit steps of adversarial transfer learning are shown in Table 1. The Learning Rate Schedule is set as decreasing by a factor of 0.1 every 20 epochs. The Adam optimizer is used in the neural network optimization process. The Adam optimizer combines the advantages of Adagrad's goodness in dealing with sparse gradients and RMSprop's goodness in dealing with non-smooth objectives, and is able to automatically adjust the learning rate, converge faster, and perform better in complex networks. Considering the amount of data in each training batch as a proportion of the overall amount of data, and the savings in training time, the batch size is set as 128 and the training epochs is set as 100.

The complete learning process of the whole framework for unseen faults is summarized in Algorithm 1.

Algorithm 1: MAM-AFFN for unseen fault diagnosis.

Input: K source domains $\mathcal{D}_1^{\mathbb{S}}, \ldots, \mathcal{D}_K^{\mathbb{S}}$ for training, training steps $\sqcup$ and β, γ.
Output: Classification results on test domain $\mathcal{D}^{\mathbb{T}}$.

Data, label, and domain $\left(\mathcal{X}, \mathcal{Y}, \mathcal{D} \middle| \left\{\mathcal{D}_1^{\mathbb{S}}, \ldots, \mathcal{D}_K^{\mathbb{S}}\right\}\right)$ augmented by MAM;
Randomly initialize the model parameters θ, $i = 0$;
while $i < \sqcup$ **do**
1. Sample a batch $\mathcal{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_K\}$ from K domains;
2. Extract the lower-level features $f_e(x)$ by the feature extractor;
3. Extract the domain-specific features $f_k(f_e(x))$ by K domain-specific fully connected layers;
4. Calculate the domain-specific loss $\mathcal{L}_{dsr}$ and output the weight for each source branch;
5. Calculate the domain-invariant loss $\mathcal{L}_{dir}$.
6. Fuse the domain-specific weight according to Equation (9).
7. Calculate the total loss of AFFN according to Equation (7).
8. Update the model parameter θ, using $AdamW$.

end while
Application on the target data of unseen domain $\mathcal{D}^{\mathbb{T}}$.
return Classification results of unseen domain $\mathcal{D}^{\mathbb{T}}$.

Table 1. Case 1: diagnostic cases from the SDUST dataset.

Diagnostic Cases	Seen Domain Condition	Unseen Domain Condition
T1	1500 r/min, 2000 r/min, 2500 r/min	1000 r/min
T2	1000 r/min, 2000 r/min, 2500 r/min	1500 r/min
T3	1000 r/min, 1500 r/min, 2500 r/min	2000 r/min
T4	1000 r/min, 1500 r/min, 2000 r/min	2500 r/min

4. Experiments

Without compromising on generality, we delve into the challenging yet crucial task of fault diagnosis for rotating bearings, where the fault characteristics exhibit periodicity and regularity. To assess the effectiveness of the general framework, we undertook two experimental case studies leveraging test rigs from SDUST [25] and the Paderborn University bearing dataset (PU dataset) [26] for bearing fault diagnosis, as shown in Figure 3. Within these datasets, we designated a specific working condition or machine as the unseen target domain, serving as the benchmark for evaluating the model's performance. The normalized method was utilized for data preprocessing, which keeps the data size within the same interval to ensure effective convergence of the model. The number of each dataset of training categories was controlled as 150 for data balancing. Meanwhile, the data pertaining to the remaining conditions were utilized for model training. The first scenario involved a constant domain shift where the rig operated at varying but constant speeds, whereas the second scenario scrutinized a constant domain shift where the rig functioned under numerous conditions with varying rotational speeds and loads. These experiments validated the efficacy of the MAM-AFFN method, which employs domain generalization principles, for demanding industrial applications.

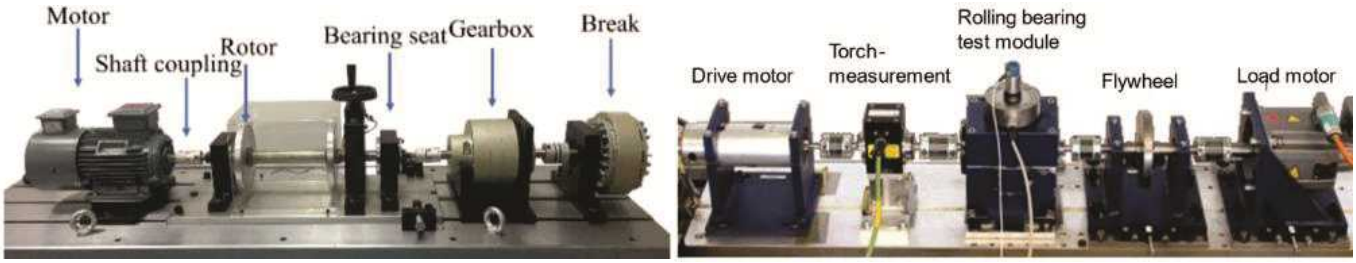


Figure 3. Illustration of (**a**) the SDUST experimental platform and (**b**) the PU bearing test-rig.

4.1. Experiment Setting

4.1.1. Description of SDUST Dataset and Case 1

The SDUST dataset comprised a motor, a shaft coupling, a rotor, a testing bearing, a gearbox, and a brake. Specifically, the bearing type used was N205EU (manufactured by SKF Co., Goteborg, Sweden), and data were gathered under four distinct health conditions: normal (NOR), inner ring fault (I), rolling element fault (B), and outer ring fault (O). Additionally, four varying working conditions were examined at speeds of 1000, 1500, 2000, and 2500 rpm. The experimental setup included four diagnostic scenarios for unseen working conditions across domains, namely T1000, T1500, T2000, and T2500, as shown in Table 1.

4.1.2. Description of PU Dataset and Case 2

The PU dataset's test rig was a comprehensive assembly, primarily encompassing a motor, a torque measurement shaft, a dedicated bearing test module, a flywheel, and a load motor. This setup facilitates the evaluation of seven distinct health conditions, including the baseline normal (N) state, three levels of inner-race faults (IF1, IF2, IF3) that progressively escalate in severity, two grades of outer-race faults (OF1, OF2) showcasing varying degrees

of damage, and a complex compound fault (CF) scenario, which integrates both inner and outer race faults (IF and OF).

Bearings exhibiting genuine damage were sourced from an accelerated lifetime testing procedure, ensuring the authenticity of fault conditions. Vibration signals were meticulously captured under four diverse operational scenarios, each characterized by specific rotational frequencies (Hz), load torques (Nm), and radial forces (N). The data acquisition was performed at a high sampling frequency of 64 kHz, ensuring precision and accuracy in capturing the dynamic behavior of the bearings. These operational scenarios collectively defined four distinct domains: P1, P2, P3, and P4, leading to four diagnosis cases, as outlined in Table 2. Each category in unseen working conditions comprises 2000 samples.

Table 2. Case 2: diagnostic cases from the PU dataset.

Domain	Rotational Frequency	Load	Cases	Seen Domain	Unseen Domain
p1	25 Hz	1000 N	R1	p2, p3, p4	p1
p2	15 Hz	400 N	R2	p1, p3, p4	p2
p3	15 Hz	1000 N	R3	p1, p2, p4	p3
p4	25 Hz	400 N	R4	p1, p2, p3	p4

4.1.3. Optimization and Compared Methods Description

The optimization methodology was tailored to maximize performance on the validation set during training. For deep generalization (DG) tasks, we employed a 10-fold cross-validation approach, dividing the data into 70% training and 30% validation subsets randomly. Balancing computational efficiency and resource utilization, we adopted a batch size of 128. To combat overfitting, we incorporated an early stopping criterion with a patience threshold of 10 epochs. Our training routine limited the maximum number of epochs to 50, leveraging Stochastic Gradient Descent (SGD) with an initial learning rate (LR) of 0.001. A dropout rate of 0.5 was employed for regularization. Furthermore, to guarantee model convergence, we employed an exponentially decaying learning rate schedule, commencing at 0.001 and adaptively adjusting as training progressed.

$$lr \;=\; initial_lr \;\cdot (\varepsilon \cdot epoch + 1)^{\gamma} \tag{12}$$

where *initial_lr* is the initial learning rate, epoch is the completed training epochs. ε and γ are two parameters that control the rate of LR's change.

For comparative analysis, several cutting-edge techniques for domain generalization (DG) were evaluated, encompassing MLDG [27], ANDMask [25], and CORAL [16]. Additionally, drawing inspiration from Matsuura and Harada [28], we devised a deep model incorporating adversarial and entropy losses, referred to as DAE, and another deep model without generalization capabilities, labeled DeepNG. The integration of Mixup and DAE within an MDAE model aimed to demonstrate the effectiveness of the domain-based discrepancy metric module.

4.2. Experimental Results

4.2.1. Performance Comparisons

The experimental results of the proposed method and the comparison methods under eight tasks in Case 1 (SDUST dataset) and Case 2 (PU dataset) are shown in Figure 4. The details are further listed in Table 3.

A few conclusions that can be drawn from the analysis are as follows: Firstly, the baseline DeepNG approach achieved significantly lower average test accuracies of 76.46% and 80.63% in the two scenarios, respectively. This suggests that there was interference among the data distributions during training, which negatively impacted the model's generalization capabilities. Moreover, DG methods employing MLDG, ANDMask, and CORAL demonstrated improvements over the basic DAE and MADAE models. Specifically, in Case 1, they achieved average accuracy gains of 1.57%, 12.06%, and 10.05%, respectively,

while in Case 2, the gains were 4.04%, 5.27%, and 3.49%. These methods were designed to mitigate distributional discrepancies between source domains and learn representations that are invariant to domain shifts. Finally, the proposed method exhibited the best diagnostic performance in nearly all DG tasks, achieving peak average accuracies of 94.15% and 93.27% in both experimental cases.

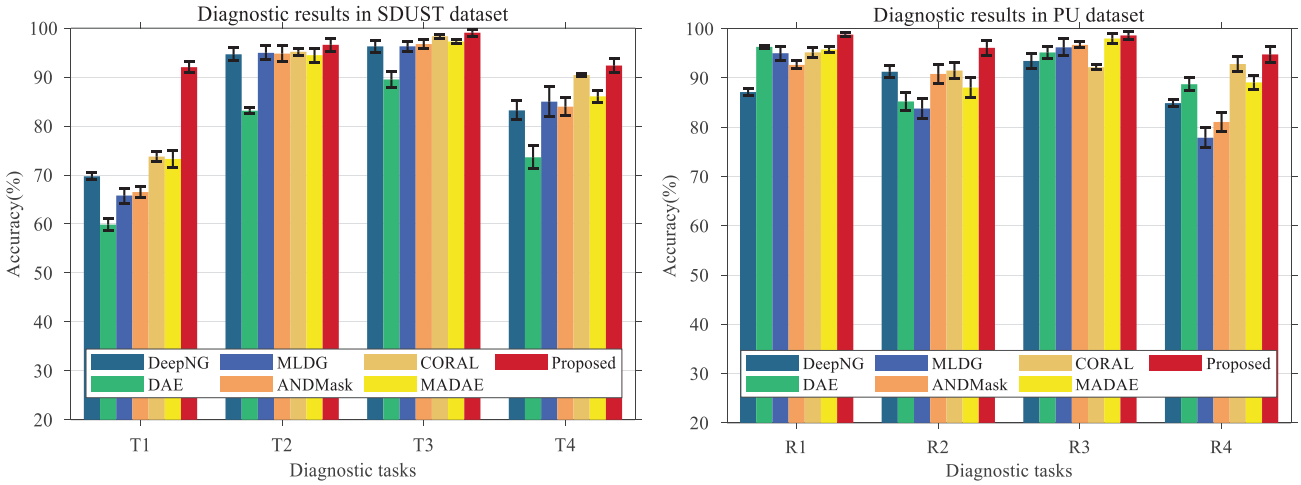


Figure 4. Classification accuracy of the different diagnostic tasks of Case 1 and Case 2.

Table 3. Diagnostic accuracy (%) of Case 1 and Case 2.

	Tasks	DeepNG	DAE	MLDG	ANDMask	CORAL	MADAE	Proposed
Case1	T1	59.73 ± 1.43	59.85 ± 2.48	65.8 ± 3.05	66.51 ± 2.2	73.75 ± 1.91	73.27 ± 3.64	93.03 ± 2.17
	T2	84.65 ± 2.72	83.14 ± 1.28	94.95 ± 2.9	94.81 ± 3.27	95.19 ± 1.52	94.48 ± 2.86	95.6 ± 2.64
	T3	86.28 ± 2.47	89.51 ± 3.21	96.28 ± 1.98	96.76 ± 1.8	98.35 ± 0.84	97.27 ± 0.74	98.06 ± 1.42
	T4	83.19 ± 3.89	83.62 ± 4.67	84.98 ± 6.26	83.96 ± 3.6	90.42 ± 0.76	86.06 ± 2.49	91.37 ± 2.94
	Average	**76.46 ± 2.63**	**78.03 ± 4.66**	**88.52 ± 3.55**	**86.51 ± 2.72**	**88.42 ± 1.26**	**88.77 ± 2.93**	**94.15 ± 2.54**
Case2	R1	75.92 ± 1.38	92.75 ± 0.69	90.26 ± 2.87	85.86 ± 1.59	90.61 ± 2.17	91.73 ± 1.26	97.6 ± 0.92
	R2	83.26 ± 2.42	72.64 ± 3.57	70.19 ± 4.15	82.47 ± 3.73	83.74 ± 3.33	77.54 ± 3.89	92.36 ± 3.02
	R3	87.33 ± 3.07	90.59 ± 2.26	92.57 ± 3.42	93.52 ± 1.24	84.98 ± 0.94	96.02 ± 1.96	97.29 ± 1.55
	R4	72.04 ± 1.53	78.73 ± 2.59	60.61 ± 4.11	65.73 ± 3.97	86.17 ± 3.12	79.39 ± 2.78	89.81 ± 3.31
	Average	**80.63 ± 2.10**	**84.67 ± 2.27**	**79.41 ± 3.64**	**85.89 ± 2.63**	**87.38 ± 2.39**	**87.17± 2.47**	**93.27 ± 2.21**

Furthermore, the model demonstrated superior stability compared to other comparison methods in most tasks. Figure 4 illustrates the classification accuracy of various diagnostic tasks for Case 1 and Case 2, providing a visual aid for comparing the diagnostic results. Figure 5 illustrates the confusion matrix of proposed methods in four diagnostic tasks of Case 1. A plausible explanation for this finding is that the data distribution without loads differs significantly from that with loads, thereby exacerbating the issue of distribution shift. Furthermore, the proposed MAM model surpasses other models in terms of both average accuracy across all tasks. In essence, the empirical findings validate the viability and efficacy of the proposed model.

To mitigate the potential bias stemming from the unique characteristics of training samples, the final analysis presents the mean and variance of five independent experiments. As evident from the graphical representation, our proposed method consistently outperforms others in diagnosing the seven distinct bearing failure scenarios, underscoring its effectiveness and robustness.

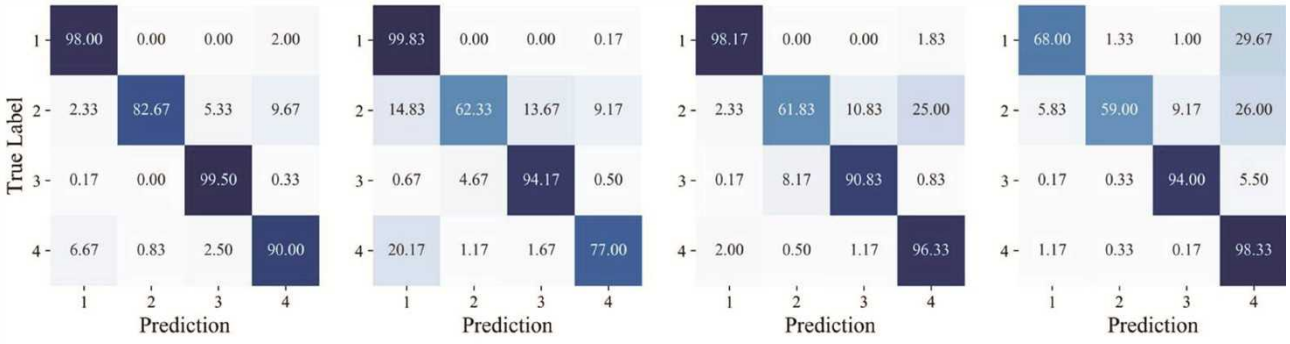


Figure 5. The diagnostic accuracies under four unseen working conditions in Case 1 utilizing the proposed method.

4.2.2. Feature Visualization and Analysis

For an intuitive and detailed comparison, the t-SNE technique [29,30] was adopted on one trial of Task T1. Four key methodologies—DeepNG, DAE, ANDMask, and CORAL—were selected for a visual assessment of their feature representations. As depicted in Figure 6a–e, the visualizations reveal distinct insights. DeepNG (Figure 6a) exhibits weak inter-class separation, indicating a potential for misdiagnosis due to category confusion. Both DAE (Figure 6b) and ANDMask (Figure 6c) show improved inter-class separation but struggle with intra-class clustering, evidenced by fragmented clusters within the same class across domains. This can lead to misclassification near decision boundaries. In contrast, our proposed approach exhibits remarkable harmony in feature distribution, significantly enhancing both inter- and intra-class coherence. This improvement can be attributed to two possibilities: significant variations between Task T1 and other conditions, or the subtle characteristics of I04 and R2 samples in T1, making them prone to misclassification.

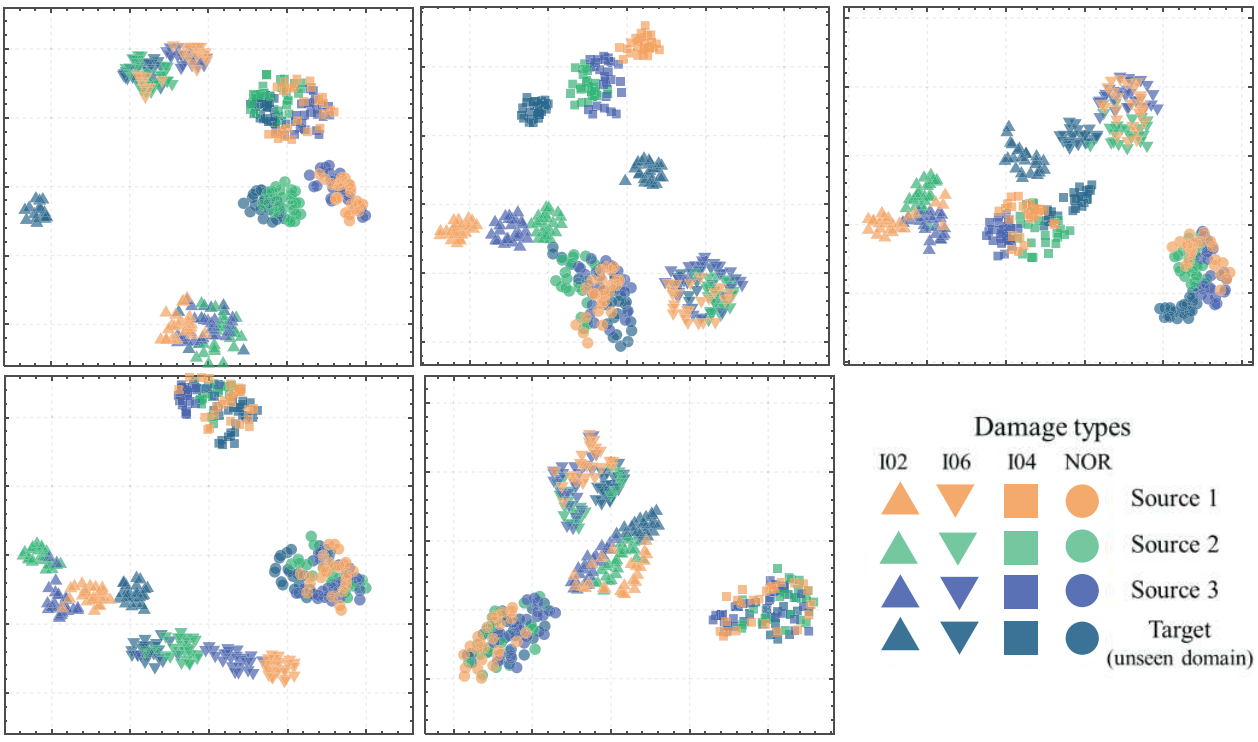


Figure 6. Visualization of the t-SNE embeddings under unseen target working conditions: (**a**) DeepNG, (**b**) DAE, (**c**) ANDMask, (**d**) CORAL, (**e**) the proposed method.

Drawing from the aforementioned analysis, we propose integrating fault mechanism and signal processing expertise as a promising avenue to further refine our approach, a direction we aim to explore in our forthcoming endeavors. Notably, our current method has demonstrated its prowess in extracting domain-invariant feature representations across diverse distributions, elucidating its exceptional generalization capabilities from Table 1.

4.2.3. Hyperparameter Sensitivity Analysis

The performance of the proposed MAM-AFFN model depends on the hyperparameters β and γ in the combined loss function as shown in Equation (7). These parameters strike a delicate balance between domain-specific and domain-invariant modules, pivotal for maintaining diagnostic accuracy amidst intricate engineering scenarios. Given the substantial data volume, we resorted to the grid search approach for parameter optimization to search β and γ from {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5,10,50} and {0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500}, respectively; the values of γ selected were ten times those of β due to domain-invariant representation being more ambiguous and difficult to access. The weight coefficients were chosen from the 100 possible combinations of the two sets, utilizing the accuracy index as the benchmark to identify the optimal pairing. We provide a detailed account of this process using the SDUST dataset as an example.

Figure 7 illustrates the MAM-AFFN's accuracy across varying hyperparameter combinations. Notably, the accuracy experiences significant fluctuations with alterations in γ, while β exerts a more subtle and consistent effect. This outcome suggests that the domain-based discrepancy metric module associated with γ holds a pivotal role in the framework, outweighing the influence of the module corresponding to β. Therefore, we adopted the optimal combination (β, γ) = (5, 300) in this experiment for the relatively best results.

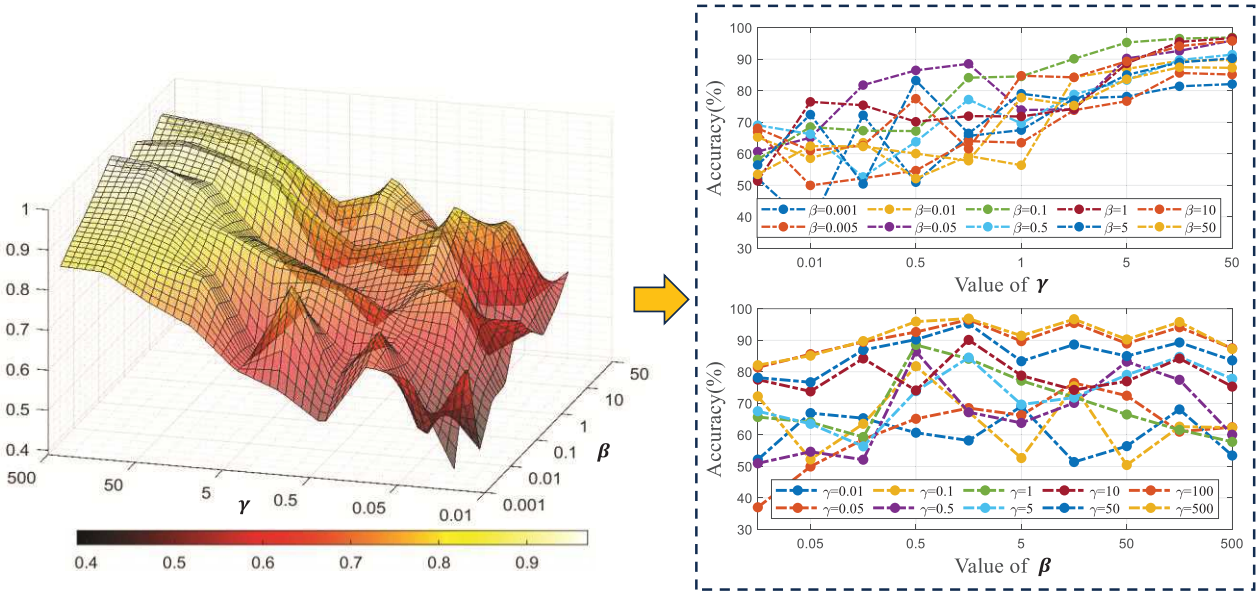


Figure 7. Diagnosis accuracy obtained using different values of the hyperparameters γ and β.

The hyperparameter α in the data augmentation module regulates the extent of interpolation between original feature vector pairs. The coefficient λ, derived from the Beta (α, α) distribution, as shown in Equation (4), remains constant for each minibatch, after which the mix-up method is implemented on the shuffled minibatch. Consequently, we explored the influence of gradually varying α from 0.1 to 1 on cross-domain diagnostic performance. For comparison, we also considered fixed λ values ranging from 0.1 to 1. Figure 8 depicts the findings on the SDUST dataset, revealing a superior average accuracy achieved by the ten value settings in the fixed α group compared to the fixed λ group.

This outcome is attributed to the fact that diverse λ values sampled from the Beta (α, α) distribution with a constant α enhance the variability of the original data, subsequently enhancing the model's generalization capabilities. As a result, α = 0.7, which yielded the highest accuracy, was selected for this study.

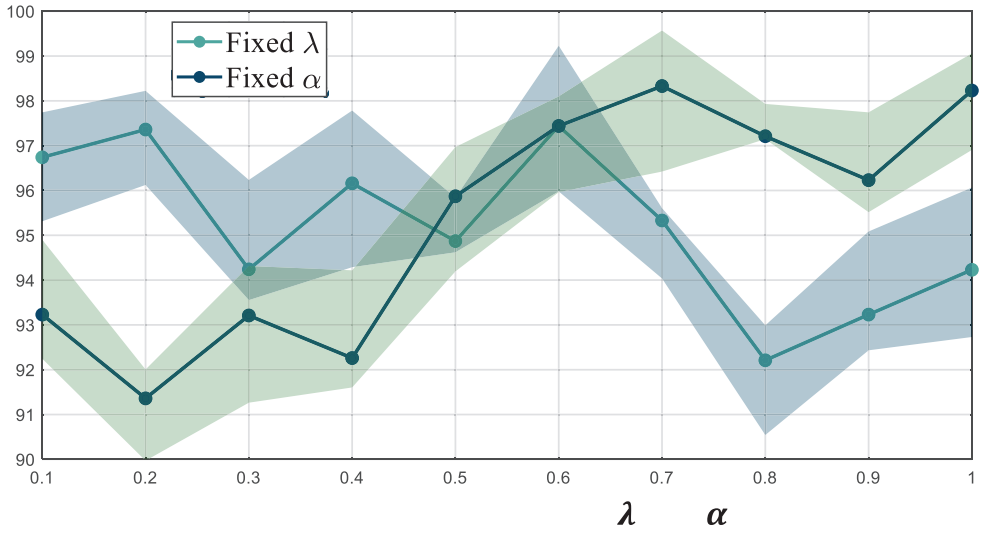


Figure 8. The strength of interpolation between data pairs significantly impacts the diagnosis performance.

5. Conclusions

In this paper, a new DG model, MAM-AFFN, is proposed to solve the problem of poor domain generalization ability under unseen working conditions in extreme real industrial applications. A domain generation framework is introduced for fault diagnosis in unseen conditions, encompassing an Adaptive Feature Fusion Domain Generation Network (AFFN) and a Mix-up Augmentation Method (MAM) that applies to both data and domain spaces. The framework incorporates thoughtfully designed modules for data augmentation, domain discriminators, and domain-based discrepancy metrics, resulting in a well-integrated domain generation diagnosis application for unseen domain conditions. The feasibility of MAM-AFFN was successfully demonstrated through a range of fault diagnosis generalization tasks conducted on two publicly available datasets as well as a practical real-world scenario. Based on the comprehensive experimental results, several significant conclusions can be inferred:

(1) A thoughtfully crafted data Mix-up Augmentation Method (MAM) is integrated within the framework, effectively leveraging the available data to broaden the model's exploration and learning horizons, ultimately enhancing its adaptability to unseen domains.
(2) The comparison results with the six other methods prove the superiority of MAM-AFFN for extracting both domain-invariant and domain-specific deep representations to enhance the generalization capability of the unseen fault diagnosis.
(3) The experimental results in eight cases of bearing diagnosis experiments from two bearings public datasets demonstrate the reliability and generalization of the proposed method, as the proposed MAM-AFFN produces higher classification accuracy compared with the traditional model and some other state-of-the-art DG-based IFD methods.

In essence, the proposed MAM-AFFN model adeptly addresses domain shift challenges stemming from varying workloads and even disparate machines, accurately executing IFD tasks under unseen conditions. Furthermore, the model's flexibility allows operators to customize adjustable thresholds, achieving a practical balance between performance and computing resources for real-world applications.

Author Contributions: Conceptualization, T.Z. and L.X.; methodology, L.X.; software, X.Z. and X.M.; validation, T.Z. and L.X.; formal analysis, H.C.; investigation, X.M. and H.C.; data curation, X.M.; writing—original draft preparation, L.X. and X.M.; writing—review and editing, L.X.; visualization, T.Z. and L.X.; supervision, H.C.; project administration, T.Z.; funding acquisition, X.M. and T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Defense Basic Scientific Research Program grant number 2022601C009.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Tong Zhang, Haowen Chen, Xianqun Mao and Xin Zhu were employed by the China State Ship-Building Corporation Limited. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Yu, X.L.; Zhao, Z.; Zhang, X.; Zhang, Q.; Yilong, L.; Sun, C.; Chen, X. Deep-Learning-Based Open Set Fault Diagnosis by Extreme Value Theory. *IEEE Trans. Ind. Inform.* **2022**, *18*, 185–196. [CrossRef]
2. Lu, S.X.; Gao, Z.; Xu, Q.; Jiang, C.; Zhang, A.; Wang, X. Class-Imbalance Privacy-Preserving Federated Learning for Decentralized Fault Diagnosis with Biometric Authentication. *IEEE Trans. Ind. Inform.* **2022**, *18*, 9101–9111. [CrossRef]
3. Li, X.; Zhang, W.; Xu, N.X.; Ding, Q. Deep Learning-Based Machinery Fault Diagnostics with Domain Adaptation Across Sensors at Different Places. *IEEE Trans. Ind. Electron.* **2020**, *6*, 6785–6794. [CrossRef]
4. Li, Q.; Shen, C.; Chen, L.; Zhu, Z. Knowledge mapping-based adversarial domain adaptation: A novel fault diagnosis method with high generalizability under variable working conditions. *Mech. Syst. Signal Process.* **2021**, *147*, 107095. [CrossRef]
5. Fan, Z.H.; Xu, Q.; Jiang, C.; Ding, S.X. Weighted quantile discrepancy-based deep domain adaptation network for intelligent fault diagnosis. *Knowl.-Based Syst.* **2022**, *240*, 13. [CrossRef]
6. Wang, J.D.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; Yu, S.P. Generalizing to Unseen Domains: A Survey on Domain Generalization. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 8052–8072.
7. Xie, J.S.; Guo, Z.; Wang, T.; Jinsong, Y. A diagnostic framework with a novel simulation data augmentation method for rail damages based on transfer learning. *Struct. Health Monit. Int. J.* **2023**, *22*, 3437–3450. [CrossRef]
8. Liao, Y.X.; Huang, R.; Jipu, L.; Chen, Z.; Li, W. Deep Semisupervised Domain Generalization Network for Rotary Machinery Fault Diagnosis Under Variable Speed. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 8064–8075.
9. Zhao, C.; Zio, E.; Shen, W.M. Domain generalization for cross-domain fault diagnosis: An application-oriented perspective and a benchmark study. *Reliab. Eng. Syst. Saf.* **2024**, *245*, 18. [CrossRef]
10. Lu, W.N.; Liang, B.; Cheng, Y.; Meng, D.; Yang, J.; Zhang, T. Deep Model Based Domain Adaptation for Fault Diagnosis. *IEEE Trans. Ind. Electron.* **2017**, *64*, 2296–2305. [CrossRef]
11. Shao, J.J.; Huang, Z.W.; Zhu, J.M. Transfer Learning Method Based on Adversarial Domain Adaption for Bearing Fault Diagnosis. *IEEE Access* **2020**, *8*, 119421–119430. [CrossRef]
12. Xiong, X.; Hongkai, J.; Xingqiu, L.; Niu, M. A Wasserstein gradient-penalty generative adversarial network with deep auto-encoder for bearing intelligent fault diagnosis. *Meas. Sci. Technol.* **2020**, *31*, 26. [CrossRef]
13. Zhao, C.; Shen, W.M. Adaptive open set domain generalization network: Learning to diagnose unknown faults under unknown working conditions. *Reliab. Eng. Syst. Saf.* **2022**, *226*, 12. [CrossRef]
14. Xu, E.R.; Li, Y.; Peng, L.; Mingshun, Y.; Liu, Y. An unknown fault identification method based on PSO-SVDD in the IoT environment. *Alex. Eng. J.* **2021**, *60*, 4047–4056. [CrossRef]
15. Fan, Z.H.; Xu, Q.; Jiang, C.; Ding, S.X. Deep Mixed Domain Generalization Network for Intelligent Fault Diagnosis Under Unseen Conditions. *IEEE Trans. Ind. Electron.* **2024**, *71*, 965–974. [CrossRef]
16. Zhao, C.; Shen, W.M. Adversarial Mutual Information-Guided Single Domain Generalization Network for Intelligent Fault Diagnosis. *IEEE Trans. Ind. Inform.* **2023**, *19*, 2909–2918. [CrossRef]
17. Peng, X.; Qiao, F.C.; Zhao, L. Out-of-Domain Generalization from a Single Source: An Uncertainty Quantification Approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 1775–1787. [CrossRef] [PubMed]
18. Zhao, Q.; Yu, W.T.; Ji, T.Y. Style Elimination and Information Restitution for generalizable person re-identification. *J. Vis. Commun. Image Represent.* **2024**, *98*, 11. [CrossRef]
19. Zhao, Y.Q.; Cheung, N.M. FS-BAN: Born-Again Networks for Domain Generalization Few-Shot Classification. *IEEE Trans. Image Process.* **2023**, *32*, 2252–2266. [CrossRef]
20. Guo, Z.; Wang, T.; Xie, J.; Yang, Y.; Peng, Q. A Deep Transfer Learning-Based Open Scenario Diagnostic Framework for Rail Damage Using Ultrasound Guided Waves. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–17. [CrossRef]
21. Han, T.; Li, Y.F.; Qian, M. A Hybrid Generalization Network for Intelligent Fault Diagnosis of Rotating Machinery Under Unseen Working Conditions. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3520011. [CrossRef]

22. Chen, L.; Li, Q.; Shen, C.; Zhu, J.; Wang, D.; Xia, M. Adversarial Domain-Invariant Generalization: A Generic Domain-Regressive Framework for Bearing Fault Diagnosis Under Unseen Conditions. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1790–1800. [CrossRef]
23. Zheng, H.L.; Yang, Y.; Xu, M. Deep Domain Generalization Combining *a priori* Diagnosis Knowledge Toward Cross-Domain Fault Diagnosis of Rolling Bearing. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 11. [CrossRef]
24. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez Paz, D. Mixup Beyond Empirical Risk Minimization. *arXiv* **2017**, arXiv:1710.09412. [CrossRef]
25. Jia, S.X.; Wang, J.; Han, B.; Zhang, G.; Wang, X.; He, J. A Novel Transfer Learning Method for Fault Diagnosis Using Maximum Classifier Discrepancy with Marginal Probability Distribution Adaptation. *IEEE Access* **2020**, *8*, 71475–71485. [CrossRef]
26. Li, X.; Zhang, W.; Ding, Q.; Sun, J.Q. Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. *J. Intell. Manuf.* **2020**, *31*, 433–452. [CrossRef]
27. Chen, K.Y.; Zhuang, D.; Chang, J.M. Discriminative adversarial domain generalization with meta-learning based cross-domain validation. *Neurocomputing* **2022**, *467*, 418–426. [CrossRef]
28. Matsuura, T.; Harada, T. Domain Generalization Using a Mixture of Multiple Latent Domains. *arxiv* **2019**, arXiv:1911.07661. [CrossRef]
29. Van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
30. Yang, B.; Wang, T.; Xie, J.; Yang, J. Deep Adversarial Hybrid Domain-Adaptation Network for Varying Working Conditions Fault Diagnosis of High-Speed Train Bogie. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–10. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

mathematics

MDPI

Article

Semi-Supervised Interior Decoration Style Classification with Contrastive Mutual Learning

Lichun Guo [1,*], Hao Zeng [1], Xun Shi [1], Qing Xu [1], Jinhui Shi [1], Kui Bai [2,3], Shuang Liang [4] and Wenlong Hang [2,3]

[1] College of Art and Design, Nanjing Audit University Jinshen College, Nanjing 210023, China
[2] College of Computer and Information Engineering, Nanjing Tech University, Nanjing 211816, China; wlhang@njtech.edu.cn (W.H.)
[3] College of Artificial Intelligence, Nanjing Tech University, Nanjing 211816, China
[4] School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
* Correspondence: guolichun@naujsc.edu.cn

Abstract: Precisely identifying interior decoration styles holds substantial significance in directing interior decoration practices. Nevertheless, constructing accurate models for the automatic classification of interior decoration styles remains challenging due to the scarcity of expert annotations. To address this problem, we propose a novel pseudo-label-guided contrastive mutual learning framework (PCML) for semi-supervised interior decoration style classification by harnessing large amounts of unlabeled data. Specifically, PCML introduces two distinct subnetworks and selectively utilizes the diversified pseudo-labels generated by each for mutual supervision, thereby mitigating the issue of confirmation bias. For labeled images, the inconsistent pseudo-labels generated by the two subnetworks are employed to identify images that are prone to misclassification. We then devise an inconsistency-aware relearning (ICR) regularization model to perform a review training process. For unlabeled images, we introduce a class-aware contrastive learning (CCL) regularization to learn their discriminative feature representations using the corresponding pseudo-labels. Since the use of distinct subnetworks reduces the risk of both models producing identical erroneous pseudo-labels, CCL can reduce the possibility of noise data sampling to enhance the effectiveness of contrastive learning. The performance of PCML is evaluated on five interior decoration style image datasets. For the average AUC, accuracy, sensitivity, specificity, precision, and F1 scores, PCML obtains improvements of 1.67%, 1.72%, 3.65%, 1.0%, 4.61%, and 4.66% in comparison with the state-of-the-art method, demonstrating the superiority of our method.

Keywords: semi-supervised learning; contrastive learning; interior decoration style

MSC: 68T45; 68T07; 68T30

1. Introduction

With the progressive improvement of living standards, there is an increasing focus on the aesthetic and functional aspects of interior decoration. Accordingly, the precise identification of interior decoration styles is of great significance for the guidance and instruction of interior design practice [1–3]. To date, many supervised learning methods based on deep neural networks have achieved great performances in interior design practice, such as interior decoration style (e.g., four images from different types of interior decoration styles in Figure 1) recognition [4,5], interior decoration style colorization [6], and interior decoration style design [7]. The success of deep learning models in this field is predominantly attributed to the availability of large amounts of labeled interior decoration images. Nevertheless, it is difficult to obtain a large number of labeled interior decoration images due to the requirements of professional knowledge and time consumption on labeling. Semi-supervised learning paradigm enables models leverage both limited labeled data and extensive unlabeled data, which significantly reduces the dependence

on annotations and progressively emerges as the mainstream paradigm for the interior decoration style recognition.



Figure 1. Examples of four types of interior decoration styles: (**a**) country style, (**b**) Chinese style, (**c**) European style, and (**d**) simple style.

The semi-supervised learning (SSL) paradigm aims to explore and leverage the internal knowledge underlying unlabeled data to enhance the performance of the model. To date, consistency learning [8] and pseudo-labeling [9] are two mainstream techniques that can be used for exploiting unlabeled data. The consistency-learning-based SSL methods enforce the model to generate consistent predictions for the same input under small perturbations [10]. The pseudo-labeling-based SSL methods generate pseudo-labels for unlabeled data, which are subsequently integrated with labeled data to retrain the model [11]. Furthermore, some other SSL methods attempt to integrate both consistency learning and pseudo-labeling techniques to boost the learning performance [12–14].

Although SSL methods based on above techniques have achieved promising results, several challenges significantly affect their robustness and may result in the degradation of model performance. First of all, most of consistency-learning-based SSL methods are based on the self-ensemble framework, which usually takes two subnetworks with identical architecture and enforces their predictions to be consistent. For instance, the mean teacher (MT) framework [15] is a typical consistency-learning-based SSL method which has a teacher subnetwork and a student subnetwork. Generally, the MT method and its extensions possess the following three characteristics: (1) the student and teacher networks share the same architecture; (2) the parameters of the teacher network are updated as the exponential moving average of those of the student network; and (3) consistency learning regularization is employed to ensure consistent predictions between the student and teacher networks. It is evident that the shared architecture leads to the homogenization of subnetworks. The model parameters of teacher network is a weighted mixture of the historical states of the student network; that means that the predictions of the teacher network are constrained by those of the student network. In addition, the consistency learning regularization enforces the consensus predictions of student and teacher networks, further limiting the diversity of the predictions of them. In summary, the coupling issue constrains MT-like SSL methods from generating diverse predictions, making the models prone to trapping confirmation bias and meaning that it is difficult for them to self-correct.

The second challenge arises from the unreliable pseudo-labels. In practice, unlabeled images usually cover different equipment and environments, thereby increasing the risk of the model making unreliable predictions. Potentially inaccurate pseudo-labels make the SSL's training and labeling loop collapse and degrade the performance of SSL methods. Furthermore, the inaccurate pseudo-labels will adversely affect the learning of internal

correlations among unlabeled images. The supervised contrastive learning strategy [16] has been demonstrated to have a superior performance in the discriminative feature learning of images. The core principle of supervised contrastive learning is that representations of similar samples should be closely aligned, while representations of different types of samples should be distinct. For unlabeled images, the pseudo-labels are usually employed to determine their categories. For instance, the authors of [17] leveraged the spatial consistency of weakly augmented images to generate similar samples, while dissimilar samples were constructed using a straightforward cross-image and pseudo-label weighting heuristic. However, utilizing pseudo-labels to generate samples may not align with actual semantic categories, potentially resulting in noisy sampling in contrastive learning. Due to the unreliability of predictions of unlabeled images, the use of self-generated pseudo-labels disrupts model training, leading to a progressive deterioration of model performance.

To tackle the aforementioned challenges, we propose a novel pseudo-label-guided contrastive mutual learning (PCML) framework for semi-supervised interior decoration style classification. Specifically, PCML employs two subnetworks with different architectures, thus directing the model generate diverse predictions. For labeled images, the inconsistent pseudo-labels generated by the two subnetworks are employed to identify images that are prone to misclassification. We then devise an inconsistency-aware relearning (ICR) regularization to perform a review training for these images. For unlabeled images, we introduce a class-aware contrastive learning (CCL) regularization to learn their discriminative feature representations using the corresponding pseudo-labels. Since the use of distinct subnetworks reduces the risk of both models producing identical erroneous pseudo-labels, CCL can reduce the possibility of noise data sampling to enhance the effectiveness of contrastive learning. We introduce a weighting module to CCL that emphasizes learning from highly probable samples within the same category while reducing the impact of unreliable noisy samples. The synergistic learning among the mutual learning framework, ICR regularization, and CCL regularization during the training process enables each subnetwork to selectively incorporate the reliable knowledge imparted by the other subnetwork, thereby mitigating the issue of confirmation bias. The primary contributions of this work can be summarized as follows:

- We propose a novel PCML framework to facilitate semi-supervised interior decoration style classification by exploiting the diversified pseudo-labels generated by distinct subnetworks.
- PCML integrates two novel modules: ICR regularization to direct the subnetworks review the labeled imaged with inconsistent predictions, and CCL regularization to learn discriminative feature representations of unlabeled images.
- The synergistic learning among the distinct subnetworks, ICR regularization, and CCL regularization helps the model overcome confirmation bias. Extensive experimental results demonstrate the superiority of PCML.

2. Related Works

2.1. Semi-Supervised Learning

By leveraging a large amount of unlabeled data, SSL methods can significantly improve the model performance. Current SSL approaches mainly focus on the consistency learning paradigm, the pseudo-labeling paradigm, or a combination of both to effectively exploit the extensive unlabeled data. Consistency-learning-based SSL methods [8,10,12–15,18–21] encourage the model to generate consistent predictions for the same input under small perturbations, such as input perturbations [8], feature perturbations [19], or network perturbations [14]. For instance, the π model presented in [16] directly uses the network's predictions of the same input under stochastic augmentation and dropout perturbations as the consistency targets. Based on [16], MT model introduces an additional teacher network with the same architecture. In [22], a sample relation consistency regularization is proposed to be integrated into the MT framework, which enables the model to capture the additional internal correlation information between unlabeled data. In addition, the local and global

structural consistencies [21] were developed to jointly learn spatial and geometric structural information, thereby enhancing the generalization capability of the MT model. In [14], three types of perturbations, i.e., input data, network, and feature perturbations, were employed to enhance model training and further improve the generalization of consistency learning. Despite the advancements, consistency-learning-based SSL methods typically employ two subnetworks with identical architectures and encourage consensus in their predictions, making the models prone to falling into confirmation bias and making it so that they find it difficult to self-correct.

Pseudo-labeling-based SSL methods follow the self-training [23] and pseudo-labeling [24] techniques to generate the pseudo-labels for unlabeled data by leveraging the model trained on labeled data. Subsequently, these pseudo-labels are incorporated into the labeled dataset to retrain the model [9,24–27]. To filter out the low quality pseudo-labels, most current SSL methods simply employ a predefined confidence threshold (e.g., 0.95) to discard potentially unreliable pseudo-labels with low confidence [24]. However, it is difficult to determine a reasonable confidence threshold for filtering out all the unreliable pseudo-labels. To address this, an entropy-based module [28] was designed to enable the model to generate low-entropy predictions for unlabeled data. In addition, a curriculum pseudo-labeling method was developed to adjust thresholds for different classes, thereby filtering out unreliable pseudo-labels [29]. Another SSL method introduced an uncertainty-aware pseudo-label selection strategy [30], which accounts for the effects of inadequate network calibration.

Some SSL methods integrated consistency learning and pseudo-labeling techniques. For example, [8] selected partially reliable pseudo-labels and guided the student network to learn from these reliable targets. The authors of [31] developed a cycled pseudo-label scheme to promote mutual consistency for challenging unlabeled data, thereby minimizing uncertain predictions. Despite the notable success of these SSL methods, most of them utilize subnetworks with identical architectures, thereby resulting in the homogenization problem. In view of this, some studies employed distinct subnetworks to enhance the diversity between them. For instance, the authors of [32] proposed the use of a convolutional neural network and a transformer as subnetworks. In addition, the authors of [33] developed a mutual correction framework utilizing two structurally distinct subnets with independent parameter updates for semi-supervised learning.

2.2. Contrastive Learning

The contrastive learning technique significantly advanced self-supervised representation learning [34–37]. The fundamental concept of contrastive learning is to draw together an anchor and a "positive" sample within the embedding space while simultaneously pushing the anchor away from the "negative" samples. In SSL, contrastive learning can fully leverage unlabeled data to learn discriminative visual representations. For example, a pseudo group contrast method [38] was developed to automatically rectify incorrect pseudo-labels. Based on [38], a reliability-aware contrastive self-ensemble framework [39] was proposed to select in-distribution unlabeled data to exploit the reliable internal correlation information, thereby enhancing the robustness of the SSL method. In addition, contrastive learning [40] was used to model pairwise similarities according to their pseudo-labels, which is beneficial for better prediction and avoids being trapped in local minimum. In [41], a graph-based contrastive learning scheme was developed to regularize the structure of the embeddings by using pseudo-labels of unlabeled data. Furthermore, contrastive learning regularization [42] was also used to improve the classification performance of the consistency regularization by well-clustered features of unlabeled data. In this paper, we extend the unsupervised contrastive learning technique to supervised scenarios [16] by using the diverse predictions generated from two distinct subnetworks, thereby facilitating the learning from samples within the same category while reducing the impact of unreliable noisy samples.

3. Pseudo-Label-Guided Contrastive Mutual Learning Framework

In this section, we first briefly introduce our pseudo-label-guided contrastive mutual learning framework. Subsequently, we elaborate on the techniques of inconsistency-aware re-learning and category-aware contrastive learning to enhance the semi-supervised interior decoration style classification performance, respectively.

3.1. Architecture Overview

We firstly introduce the basic formulations of SSL for interior decoration style classification. We have a set of training dataset D, which contains N fully labeled images $D^l = \left\{\left(\mathbf{x}_i^l, \mathbf{y}_i^l\right)\right\}_{i=1}^{N}$ and M unlabeled images $D^u = \left\{\mathbf{x}_i^u\right\}_{i=N+1}^{N+M}$. In practice, we have $N \ll M$. $\mathbf{x}_i^l \in \mathcal{X}$ denotes the i-th training image in D^l and $\mathbf{y}_i^l \in \mathcal{Y}$=$\{1, 2, \cdots, C\}$ denotes the corresponding one-hot ground-truth label of $\mathbf{x}_i^l$.

In PCML, we employ two networks with different architectures, i.e., subnetwork A ($f_A(\cdot)$) and subnetwork B ($f_B(\cdot)$). We denote θ_A and θ_B as the model parameters of subnetworks A and B, respectively. The subnetwork f can be decomposed into a feature extractor $h(\cdot) : \mathcal{X} \to \mathcal{Z}$ and a classifier $g(\cdot) : \mathcal{Z} \to \mathcal{Y}$, parameterized by θ^h and θ^g, respectively. Here, $\mathcal{Z} \subset \mathbb{R}^Z$ represents the feature space of dimension. For subnetwork A, we have $f_A = g_A \circ h_A$ and $\theta_A = \left\{\theta_A^g, \theta_A^h\right\}$. For subnetwork B, we have $f_B = g_B \circ h_B$ with $\theta_B = \left\{\theta_B^g, \theta_B^h\right\}$. Our goal is to accurately predict the style of interior decoration images by using training dataset D.

The overall framework of the proposed PCML is shown in Figure 2. PCML employs subnetwork A and subnetwork B with different architectures to direct the model generate diverse predictions, thereby mitigating the homogenization problem. Utilizing the diverse predictions, PCML takes ICR regularization to conduct review training on labeled images that are prone to misprediction. For unlabeled images, PCML introduces CCL regularization to learn their discriminative feature representations using the corresponding pseudo-labels. Since the diverse predictions reduce the risk of both models producing identical erroneous pseudo-labels, CCL can reduce the possibility of noise data sampling to enhance the effectiveness of contrastive learning. Below, we introduce the main components of PCML, i.e., ICR regularization, and CCL regularization.

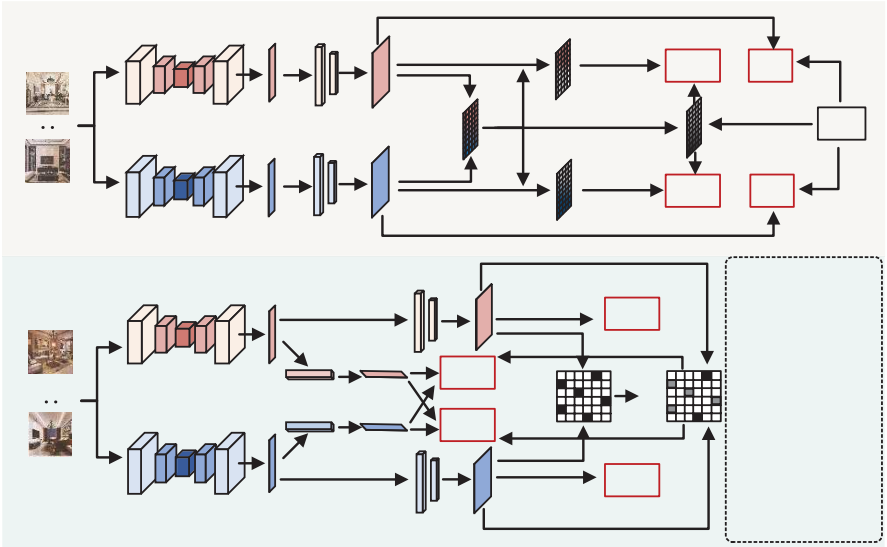


Figure 2. The architecture of PCML for medical image segmentation. A mini-batch input containing labeled data and unlabeled images are fed into subnetwork A and subnetwork B, respectively. The predictions $\mathbf{P}_A^l$ and $\mathbf{P}_B^l$ are used to calculate the ICR regularization and cross-entropy loss, respectively. The predictions $\mathbf{P}_A^u$ and $\mathbf{P}_B^u$ are used to calculated the CPS loss and the class-aware matrix, which is then used to guide the calculation of the CCL regularization.

3.2. Inconsistency-Aware Relearning

Assume that the input min-batch of data M contains N samples $\mathbf{x}^M$, comprising labeled image $\mathbf{x}^l$ randomly sampled from D^l and unlabeled image $\mathbf{x}^u$ randomly sampled from D^u. The input min-batch of images $\mathbf{x}^M$ are fed into feature extractors $h_A(\cdot)$ and $h_B(\cdot)$, respectively. Then, we can obtain the feature representations as follows:

$$\mathbf{z}_A^l, \mathbf{z}_A^u = h_A\left(\mathbf{x}^M, \theta_A^h\right), \tag{1}$$

$$\mathbf{z}_B^l, \mathbf{z}_B^u = h_B\left(\mathbf{x}^M, \theta_B^h\right). \tag{2}$$

For labeled images, the probability outputs of subnetworks A and B can be obtained by using softmax function $\sigma(\cdot)$, which can be formulated as follows:

$$\mathbf{p}_A^l = \sigma\left(g_A\left(\mathbf{z}_A^l, \theta_A^g\right)\right), \tag{3}$$

$$\mathbf{p}_B^l = \sigma\left(g_B\left(\mathbf{z}_B^l, \theta_B^g\right)\right). \tag{4}$$

Since $\mathbf{P}_A^l$ and $\mathbf{P}_B^l$ originate from two different subnetworks, their consistency suggests that the corresponding pseudo-labels are highly accurate. If the two subnetworks yield different predictions for the same image, it indicates that at least one of the predictions is incorrect. As a result, we use the inconsistent predictions as indicators to identify the images that are prone to misclassification. Consequently, the subnetworks need to relearn these images to fully leverage the knowledge within the limited labeled data.

To this end, we propose an ICR regularization approach to selectively leverage the images with inconsistent predictions to retrain the two subnetworks. Specifically, the predicted one-hot label of annotated images obtained from subnetwork A and subnetwork B can be expressed as follows:

$$\mathbf{y}_A^l = \textit{One-hot}\left(\mathbf{p}_A^l\right), \mathbf{y}_B^l = \textit{One-hot}\left(\mathbf{p}_B^l\right), \tag{5}$$

where $\textit{One-hot}(\cdot)$ transforms the predictions $\mathbf{p}_A^l$ and $\mathbf{p}_B^l$ to the responding hard labels $\mathbf{y}_A^l$ and $\mathbf{y}_B^l$. The indexes of images with inconsistent predictions can be expressed as follows:

$$\mathbf{IN} = \mathbb{I}\left(\mathbf{y}_A^l \neq \mathbf{y}_B^l\right). \tag{6}$$

Here, $\mathbb{I}(\cdot)$ is the indicator function. Thus, the predictions corresponding to these images obtained from subnetworks A and B can be identified using operation $F(\cdot)$:

$$\hat{\mathbf{P}}_A^l = F\left(\mathbf{IN}, \mathbf{P}_A^l\right), \hat{\mathbf{P}}_B^l = F\left(\mathbf{IN}, \mathbf{P}_B^l\right). \tag{7}$$

Similarly, the truth labels corresponding to these images can be expressed as follows:

$$\hat{\mathbf{y}}^l = F\left(\mathbf{IN}, \mathbf{y}^l\right). \tag{8}$$

To facilitate the relearning of samples prone to misprediction, we introduce the ICR regularization, which directs the two subnetworks to focus more on these images. Thus, the ICR regularization employed for subnetworks A and B can be expressed as follows:

$$L_{ICR}^A = MSE\left(\hat{\mathbf{P}}_A^l, \hat{\mathbf{y}}^l\right), L_{ICR}^B = MSE\left(\hat{\mathbf{P}}_B^l, \hat{\mathbf{y}}^l\right). \tag{9}$$

where $MSE(\cdot)$ denotes the mean squared error loss.

3.3. Class-Aware Contrastive Learning

In the SSL paradigm, supervised contrastive learning [40,41] is often employed to guide the model to focus on the pairwise similarities of unlabeled data based on their pseudo-labels, thereby facilitating discriminative feature learning. Although this is effective, it is difficult to use it to guarantee the reliability of self-generated pseudo-labels, which can adversely affect the performance of semi-supervised classification. In view of this, we propose a class-aware contrastive learning (CCL) scheme to learn discriminative feature representations of unlabeled images by selectively leveraging reliable pseudo-labels. Different from [43], we employ two subnetworks with distinct architectures which can help the model reduce the risk of producing identical erroneous pseudo-labels and dispose of confirmation bias. Hence, CCL can reduce the possibility of noise data sampling to enhance the effectiveness of contrastive learning.

Specifically, the objective of classical supervised contrastive learning can be mathematically expressed as follows:

$$L_{SCL} = -\sum_{i} \frac{1}{J+1} \sum_{p \in P(i)} \log \frac{\exp\left(\hat{\mathbf{z}}^u_{A,i} \cdot \hat{\mathbf{z}}^u_{B,p} \Big/ \tau\right)}{\sum_{j=1}^{N} \mathbb{I}_{i \neq j} \exp\left(\hat{\mathbf{z}}^u_{A,i} \cdot \hat{\mathbf{z}}^u_{B,j} \Big/ \tau\right)}. \tag{10}$$

Here, $\hat{\mathbf{z}}^u_{A,i}$ and $\hat{\mathbf{z}}^u_{B,j}$, $\forall i, j = 1, 2, \cdots, N$ represent the normalized embeddings output by the projection networks $p_A(\cdot)$ and $p_B(\cdot)$, respectively. Thus, we have $\hat{\mathbf{z}}^u_{A,i} = p_A\left(\mathbf{z}^u_{A,i}\right)$ and $\hat{\mathbf{z}}^u_{B,i} = p_B\left(\mathbf{z}^u_{B,i}\right)$. $P(i)$ is the set of indices of all positives according to their pseudo-labels. τ is a hyper-parameter for temperature scaling.

In practice, unlabeled images usually come different equipment and environments, thereby increasing the risk of the model making unreliable predictions. The positive samples identified by pseudo-labels in Equation (10) may be inaccurate due to their incorrect predictions. In view of this, we introduce CCL regularization for supervised contrastive learning among the reliable unlabeled images. Specifically, the probability outputs of unlabeled images from subnetwork A and subnetwork B can be expressed as follows:

$$\mathbf{p}^u_A = \sigma\left(g_A\left(\mathbf{z}^u_A, \theta^g_A\right)\right), \tag{11}$$

$$\mathbf{p}^u_B = \sigma\left(g_B\left(\mathbf{z}^u_B, \theta^g_B\right)\right). \tag{12}$$

We then use a threshold T to filter out some potentially unreliable unlabeled images. If $\mathbf{p}^u_{A,i} > T$, we assume the i-th unlabeled image has a high probability to be a reliable data and should be pulled closer with the same class. To achieve this, we define a class-aware matrix $\mathbf{M}$, each element of which can be formulated as follows:

$$m_{ij} = \begin{cases} 1, & \text{if } i = j \\ 1, & \text{if } y_{\mathbf{z}^u_{A,i}} = y_{\mathbf{z}^u_{B,j}} \text{ and } \mathbf{p}^u_{A,i}, \mathbf{p}^u_{B,j} > T \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $y_{\mathbf{z}^u_{A,i}}$ and $y_{\mathbf{z}^u_{B,j}}$ denote the self-generated pseudo-labels. Although use class-aware matrix $\mathbf{M}$ can filter out most of unreliable samples, the predictions of unlabeled images above threshold T still might be wrong. In view of this, we propose to reweigh these potentially unreliable unlabeled images to minimize the impact of them and maximize the utilization of their valuable information. Specifically, we direct the model to focus on high-confidence unlabeled images while diminishing the potential bias originating from unreliable ones. Thus, the class-aware matrix $\mathbf{M}$ can be reformulated as $\hat{\mathbf{M}}$ through simply multiply their probability outputs. Each element $\hat{m}_{ij}$ of $\hat{\mathbf{M}}$ can be defined as follows:

$$\hat{m}_{ij} = \begin{cases} \mathbf{p}^u_{A,i} \cdot \mathbf{p}^u_{B,j} \cdot m_{ij}, & \text{if } i \neq j \\ m_{ij}, & \text{otherwise} \end{cases} \tag{14}$$

Thus, the CCL regularization can be obtained by combining the class-aware matrix $\hat{\mathbf{M}}$ and supervised contrastive learning scheme, which can be formulated as follows:

$$L^A_{CCL} = -\sum_i \frac{1}{J+1} \sum_{p \in P(i)} \log \frac{\hat{m}_{ij} \cdot \exp\left(\hat{\mathbf{z}}^u_{A,i} \cdot \hat{\mathbf{z}}^u_{B,p} \Big/ \tau\right)}{\sum_{j=1}^N \mathbb{I}_{i \neq j} \exp\left(\hat{\mathbf{z}}^u_{A,i} \cdot \hat{\mathbf{z}}^u_{B,j} \Big/ \tau\right)}, \tag{15}$$

$$L^B_{CCL} = -\sum_i \frac{1}{J+1} \sum_{p \in P(i)} \log \frac{\hat{m}_{ij} \cdot \exp\left(\hat{\mathbf{z}}^u_{B,i} \cdot \hat{\mathbf{z}}^u_{A,p} \Big/ \tau\right)}{\sum_{j=1}^N \mathbb{I}_{i \neq j} \exp\left(\hat{\mathbf{z}}^u_{B,i} \cdot \hat{\mathbf{z}}^u_{A,j} \Big/ \tau\right)}. \tag{16}$$

3.4. Objective Function

Overall, the total objective function of subnetworks in PCML can be formulated as follows:

$$L_A = L^A_{sup} + L^A_{unsup}, \; L_B = L^B_{sup} + L^B_{unsup} \tag{17}$$

where L^A_{sup}, L^B_{sup}, and L^A_{unsup}, L^B_{unsup} denote the supervised loss and the unsupervised loss for subnetwork A or subnetwork B, respectively. The supervised losses for subnetwork A and subnetwork B can be expressed as follows:

$$L^A_{sup} = CE\left(\mathbf{P}^l_A, \mathbf{y}\right) + L^A_{ICR}, \tag{18}$$

$$L^B_{sup} = CE\left(\mathbf{P}^l_B, \mathbf{y}\right) + L^B_{ICR}. \tag{19}$$

where $CE(\cdot)$ denotes the cross-entropy loss. The unsupervised losses for subnetwork A and subnetwork B can be formulated as follows:

$$L^A_{unsup} = \lambda_u \cdot L^A_{CPS} + \lambda_c \cdot L^A_{CCL}, \tag{20}$$

$$L^B_{unsup} = \lambda_u \cdot L^B_{CPS} + \lambda_c \cdot L^B_{CCL}. \tag{21}$$

Here, L^A_{CPS} and L^B_{CPS} denote the cross pseudo supervision (CPS) losses, e.g., $L^A_{CPS} = CE\left(\mathbf{P}^u_A, \mathbf{y}^u_B\right)$. λ_u and λ_c are hyper-parameters to balance the cross pseudo supervision loss and CCL regularization.

Compared with regular optimization of a single network, PCML needs to jointly optimize two different subnetworks. Therefore, compared with the regular training procedures of network backbone, PCML needs approximately 2 × training time. Finally, we summarize the thorough optimization process of PCML framework in Algorithm 1.

Algorithm 1: Optimization of PCML Framework

Input: Labeled dataset D^l, unlabeled dataset D^u, trade-off coefficients λ_u and λ_c, max iteration IT, batch size.
Output: Parameters θ_A^T and θ_B^T of subnetworks A and B.

Initialize: Parameters θ_A^0 and θ_B^0
for $t = 0$ *to* $T - 1$ **do**
  for *each mini-batch M*, **do**
    Compute $\mathbf{z}_A^l$, $\mathbf{z}_A^u$, $\mathbf{z}_B^l$ and $\mathbf{z}_B^u$ using Equations (1) and (2);
    Compute $\mathbf{P}_A^l$ and $\mathbf{P}_B^l$ using Equations (3) and (4);
    Compute $\mathbf{y}_A^l$, $\mathbf{y}_B^l$ and **IN** using Equations (5) and (6);
    Compute $\hat{\mathbf{P}}_A^l$, $\hat{\mathbf{P}}_B^l$ and $\hat{\mathbf{y}}^l$ using Equations (7) and (8);
    Compute L_{ICR}^A and L_{ICR}^B using Equation (9);
    Compute $\mathbf{p}_A^u$ and $\mathbf{p}_B^u$ using Equations (11) and (12);
    Compute matrix $\hat{\mathbf{M}}$ using Equation (14);
    Compute $\hat{\mathbf{z}}_{A,i}^u$ and $\hat{\mathbf{z}}_{B,i}^u$, $\forall i$;
    Compute L_{CCL}^A and L_{CCL}^B using Equations (15) and (16);
    `// Optimize model parameter for subnetwork A`
    ;
    Compute supervised loss L_{sup}^A with L_{ICR}^A and $CE\left(\mathbf{P}_A^l, \mathbf{Y}^l\right)$ using Equation (18);
    Compute unsupervised loss L_{unsup}^A using Equation (20);
    Optimize θ_A^t using Equation (17);
    `// Optimize model parameter for subnetwork B`
    Compute supervised loss L_{sup}^B with L_{ICR}^B and $CE\left(\mathbf{P}_B^l, \mathbf{Y}^l\right)$ using Equation (19);
    Compute unsupervised loss L_{unsup}^B using Equation (21);
    Optimize θ_B^t using Equation (17).
  end
end

4. Experiments

In this section, we evaluate PCML using five real-world interior decoration style image datasets: (1) TV background wall, (2) chandelier, (3) living room, (4) dining room, and (5) bedroom (shown in Figure 3). First of all, we provide descriptions of the five datasets. Subsequently, we outline the comparison methods and offer the implementation details of the experiments. Finally, we present and analyze the experimental results.

4.1. Datasets and Pre-Processing

(1) *TV background wall dataset*: The available training dataset contains 1643 images of interior decoration style, labeled by four different types of styles (country style, Chinese style, European style, and simple style). Each type has 250, 270, 322, and 801 images, respectively, with each image having a resolution of 900 × 700 pixels. (2) *Chandelier dataset*: This dataset consists of 969 interior decoration style images. It contains 295, 169, 351, and 154 images for four types of styles, respectively. (3) *Living room dataset*: This dataset consists of 1489 interior decoration style images, involving 138 country style images, 248 Chinese style images, 523 European style images, and 580 simple style images, respectively. (4) *Dining room dataset*: This dataset consists of 520 images, involving 91 country style images, 98 Chinese style images, 178 European style images, and 153 simple style images, respectively. (5) *Bedroom dataset*: This dataset contains 643 interior decoration style images, involving 149 country style images, 119 Chinese style images, 191 European style images, and 184 simple style images, respectively.



Figure 3. Images from five datasets: (**a**) TV background wall, (**b**) chandelier, (**c**) living room, (**d**) dining room, and (**e**) bedroom. The first to the fourth rows denote the country style, Chinese style, European style, and simple style, respectively.

In the experiment, each dataset was randomly partitioned into three subsets: 70% for training, 10% for validation, and 20% for testing.

4.2. Experimental Setup

To validate the effectiveness of the proposed SSL framework for medical image segmentation, we conducted a comparison between our PCML and state-of-the-art SSL methods. The comparison methods include the following: (1) ResNet [44]; (2) DenseNet121 [45]; (3) MixMatch [26]; (4) ReMixMatch [46]; (5) CoMatch [41]; (6) FixMatch [27]; (7) MT model [15]; (8) SRC-MT [22]; (9) the proposed PCML.

In the experiment, we re-implemented all comparison methods using open-source code. We used DenseNet as the network backbone for the comparison methods MixMatch, ReMixMatch, CoMatch, FixMatch, MT, and SRC-MT, due to its superior performance. For the proposed PCML, we employed ResNet and DenseNet as subnetwork A and subnetwork B, respectively. For pseudo-labeling-based SSL comparison methods MixMatch, ReMixMatch, CoMatch, FixMatch, we applied a weak augmentation and a strong augmentation on the input data, respectively. For consistency-learning-based SSL methods MT and SRC-MT, the input image perturbation and random transformations, i.e., rotation, translation, horizontal flips, were applied to each image [22]. For pseudo-labeling-based SSL methods, we set the confidence threshold to 0.95 to keep consistency with other pseudo-labeling-based SSL comparison methods [27,29,30,32,38]. In addition, for consistency-learning-based SSL methods, the smoothing parameter exponential moving average was set to 0.99. All other settings of the comparison methods remained consistent with the original work. For the proposed PCML, the projection network contains two linear layers with 2048 and 64 hidden neurons. The trade-off coefficients λ_u and λ_c were set to 0.5 and 0.1 for all datasets. In addition, the threshold was set to 0.9 for living room dataset, and 0.8 for the rest datasets. For the used five datasets, subnetworks were trained by Adam optimizer for 10K iterations, with the fixed learning rate 0.0001. The batch size was set to 24, comprising 12 labeled image and 12 unlabeled images. The implementation of the proposed PCML framework was performed using PyTorch 2.4.1 on a two RTX 4090 GPUs.

We adopted AUC, accuracy (ACC), sensitivity (SEN), specificity (SPE), precision (PREC), and F1 score (F1) as the evaluation metrics to comprehensively evaluate the

classification performance of all comparison methods. The higher values of these metrics indicate better performance of the model.

4.3. Comparison with State-of-the-Art Methods

Tables 1–5 present the classification performance of all SSL comparison methods using 20% labeled images and 80% unlabeled images in training set on five image datasets. In our experiment, the classification results obtained by training the fully supervised models, i.e., ResNet and DenseNet, with 100% labeled data, are regarded as the upper-bound performance. Meanwhile, the classification results achieved by ResNet and DenseNet trained with only 20% labeled images serve as the baseline performance. The best classification results for each evaluation metric were highlighted in bold. Additionally, we listed the performance improvements achieved by all SSL methods relative to the better baseline method, DenseNet, with these enhancements presented in parentheses. Based on the experimental results provided in Tables 1–5, the following observations can be obtained.

Table 1. Performance comparison with different semi-supervised learning methods on TV background wall dataset.

Methods	Percentage		Metrics					
	Labeled	Unlabeled	AUC (%)↑	ACC (%)↑	SEN (%)↑	SPE (%)↑	PREC (%)↑	F1 (%)↑
DenseNet [45]	100%	0	97.44	94.04	86.12	95.09	86.18	86.08
ResNet [44]	100%	0	97.30	93.73	85.47	95.30	83.55	84.23
DenseNet [45]	20%	0	93.76	88.53	73.84	91.45	72.73	72.90
ResNet [44]	20%	0	93.31	89.30	76.39	91.22	75.50	75.58
MixMatch [26]	20%	80%	94.77 (1.01)	90.83 (2.29)	78.98 (5.14)	91.81 (0.36)	80.85 (8.12)	79.48 (6.58)
ReMixMatch [46]	20%	80%	95.48 (1.72)	90.52 (1.99)	76.11 (2.27)	92.10 (0.65)	80.42 (7.69)	77.55 (4.65)
CoMatch [41]	20%	80%	95.43 (1.67)	91.28 (2.75)	78.67 (4.48)	92.53 (1.08)	82.05 (9.32)	79.85 (6.95)
FixMatch [27]	20%	80%	95.29 (1.53)	89.76 (1.22)	77.26 (3.42)	91.81 (0.36)	77.19 (4.46)	76.90 (4.00)
MT [15]	20%	80%	95.55 (1.79)	90.06 (1.53)	74.03 (0.20)	91.51 (0.06)	81.07 (8.34)	75.98 (3.08)
SRC-MT [22]	20%	80%	95.84 (2.08)	91.36 (2.83)	80.11 (6.27)	93.08 (1.63)	81.33 (8.60)	79.76 (6.86)
PCML (Ours)	20%	80%	**96.82 (3.06)**	**93.58 (5.05)**	**84.12 (10.3)**	**94.48 (3.03)**	**87.27 (14.5)**	**85.36 (12.5)**

Table 2. Performance comparison with different semi-supervised learning methods on chandelier dataset.

Methods	Percentage		Metrics					
	Labeled	Unlabeled	AUC (%)↑	ACC (%)↑	SEN (%)↑	SPE (%)↑	PREC (%)↑	F1 (%)↑
DenseNet [45]	100%	0	99.61	97.28	95.58	97.91	93.85	94.61
ResNet [44]	100%	0	99.33	96.11	92.39	97.11	92.37	92.34
DenseNet [45]	20%	0	93.96	87.69	78.75	91.09	77.72	76.53
ResNet [44]	20%	0	93.71	86.92	75.01	90.37	75.97	73.51
MixMatch [26]	20%	80%	95.75 (1.79)	91.58 (3.89)	84.72 (5.97)	93.55 (2.47)	83.14 (5.42)	83.63 (7.10)
ReMixMatch [46]	20%	80%	96.51 (2.55)	92.10 (4.40)	86.13 (7.37)	93.86 (2.78)	85.22 (7.50)	85.03 (8.50)
CoMatch [41]	20%	80%	97.57 (3.61)	92.75 (5.05)	86.55 (7.80)	94.42 (3.33)	85.90 (8.18)	85.85 (9.32)
FixMatch [27]	20%	80%	97.66 (3.70)	93.39 (5.70)	86.86 (8.10)	95.07 (3.98)	85.54 (7.82)	86.11 (9.58)
MT [15]	20%	80%	96.38 (2.42)	92.75 (5.05)	87.45 (8.70)	94.55 (3.46)	86.38 (8.66)	86.70 (10.2)
SRC-MT [22]	20%	80%	97.35 (3.39)	92.49 (4.79)	88.66 (9.91)	94.42 (3.33)	84.50 (6.77)	84.07 (7.54)
PCML (Ours)	20%	80%	**98.25 (4.29)**	**95.21 (7.51)**	**90.88 (12.1)**	**96.65 (5.56)**	**90.47 (12.8)**	**90.55 (14.0)**

Table 3. Performance comparison with different semi-supervised learning methods on living room dataset.

Methods	Percentage		Metrics					
	Labeled	Unlabeled	AUC (%)↑	ACC (%)↑	SEN (%)↑	SPE (%)↑	PREC (%)↑	F1 (%)↑
DenseNet [45]	100%	0	94.31	92.42	83.30	94.26	83.91	83.52
ResNet [44]	100%	0	94.86	92.09	81.81	93.90	84.50	82.52
DenseNet [45]	20%	0	89.74	86.95	69.34	89.78	69.45	67.28
ResNet [44]	20%	0	91.06	87.88	70.41	90.89	68.94	67.81
MixMatch [26]	20%	80%	91.26 (1.53)	88.89 (1.94)	72.91 (3.57)	91.63 (1.85)	77.07 (7.62)	73.36 (6.07)
ReMixMatch [46]	20%	80%	92.12 (2.38)	89.39 (2.44)	72.91 (3.57)	92.19 (2.41)	74.09 (4.64)	75.81 (8.53)
CoMatch [41]	20%	80%	91.58 (1.84)	89.65 (2.69)	74.69 (5.36)	92.17 (2.38)	77.54 (8.08)	74.06 (6.77)
FixMatch [27]	20%	80%	92.53 (2.80)	88.72 (1.77)	70.55 (1.21)	91.63 (1.85)	76.60 (7.15)	69.57 (2.29)
MT [15]	20%	80%	91.91 (2.17)	87.96 (1.01)	71.77 (2.44)	91.16 (1.38)	73.74 (4.29)	69.70 (2.42)
SRC-MT [22]	20%	80%	92.41 (2.67)	88.22 (1.26)	71.98 (2.64)	91.07 (1.29)	74.23 (4.78)	70.43 (3.15)
PCML (Ours)	20%	80%	**93.92 (4.19)**	**91.25 (4.29)**	**77.27 (7.93)**	**93.33 (3.55)**	**79.74 (10.3)**	**76.74 (9.46)**

Table 4. Performance comparison with different semi-supervised learning methods on dining room dataset.

Methods	Percentage		Metrics					
	Labeled	Unlabeled	AUC (%)↑	ACC (%)↑	SEN (%)↑	SPE (%)↑	PREC (%)↑	F1 (%)↑
DenseNet [45]	100%	0	98.03	94.71	91.66	95.67	89.27	90.2
ResNet [44]	100%	0	97.87	93.75	89.48	95.12	86.94	88.06
DenseNet [45]	20%	0	91.89	87.26	76.37	89.71	74.64	75.24
ResNet [44]	20%	0	91.29	83.89	72.65	87.08	68.83	69.04
MixMatch [26]	20%	80%	95.20 (3.31)	89.18 (1.92)	79.68 (3.31)	90.46 (0.75)	80.20 (5.55)	78.60 (3.37)
ReMixMatch [46]	20%	80%	95.89 (4.00)	91.11 (3.85)	82.60 (6.23)	92.82 (3.11)	78.12 (3.48)	80.27 (5.03)
CoMatch [41]	20%	80%	96.08 (4.18)	89.66 (2.40)	82.20 (5.83)	91.93 (2.22)	75.89 (1.24)	76.62 (1.38)
FixMatch [27]	20%	80%	94.93 (3.04)	89.90 (2.64)	83.21 (6.85)	91.74 (2.03)	82.01 (7.36)	81.72 (6.48)
MT [15]	20%	80%	93.97 (2.08)	88.46 (1.20)	81.76 (5.39)	91.92 (2.21)	76.00 (1.36)	77.65 (2.41)
SRC-MT [22]	20%	80%	95.45 (3.56)	91.11 (3.85)	85.17 (8.80)	94.09 (4.39)	77.86 (3.22)	79.53 (4.29)
PCML (Ours)	20%	80%	**97.23 (5.34)**	**93.51 (6.25)**	**88.96 (12.6)**	**94.75 (5.05)**	**86.35 (11.7)**	**87.59 (12.4)**

Table 5. Performance comparison with different semi-supervised learning methods on bedroom dataset.

Methods	Percentage		Metrics					
	Labeled	Unlabeled	AUC (%)↑	ACC (%)↑	SEN (%)↑	SPE (%)↑	PREC (%)↑	F1 (%)↑
DenseNet [45]	100%	0	88.40	85.74	72.55	89.59	72.35	71.95
ResNet [44]	100%	0	88.57	85.16	72.58	89.03	71.12	70.68
DenseNet [45]	20%	0	81.35	78.71	57.78	84.75	59.35	56.82
ResNet [44]	20%	0	82.45	78.32	57.60	84.86	61.64	56.12
MixMatch [26]	20%	80%	84.04 (2.69)	81.84 (3.13)	65.42 (7.64)	86.67 (1.92)	63.59 (4.24)	64.01 (7.20)
ReMixMatch [46]	20%	80%	85.23 (3.88)	81.25 (2.54)	63.51 (5.74)	86.34 (1.58)	62.17 (2.82)	62.11 (5.30)
CoMatch [41]	20%	80%	85.20 (3.85)	80.86 (2.15)	65.62 (7.84)	86.32 (1.56)	62.25 (2.90)	62.25 (5.43)
FixMatch [27]	20%	80%	84.96 (3.61)	81.45 (2.73)	67.54 (9.77)	86.34 (1.58)	63.89 (4.54)	63.97 (7.16)
MT [15]	20%	80%	84.44 (3.09)	80.86 (2.15)	64.05 (6.27)	86.67 (1.92)	61.37 (2.02)	61.67 (4.86)
SRC-MT [22]	20%	80%	86.01 (4.66)	80.08 (1.37)	60.23 (2.46)	86.60 (1.84)	61.64 (2.29)	60.04 (3.23)
PCML (Ours)	20%	80%	**86.71 (5.36)**	**82.42 (3.71)**	**69.26 (11.5)**	**86.80 (2.05)**	**65.10 (5.75)**	**66.52 (9.71)**

In general, SSL methods consistently achieved improved classification performance compared to the baseline methods ResNet and DenseNet by leveraging unlabeled images. This verifies the efficacy of exploring the internal knowledge underlying unlabeled data. The proposed PCML framework exhibits superior classification results com-pared to other SSL methods across all evaluative metrics. These results confirm the effectiveness of integrating ICR regularization, which directs the two subnetworks to review labeled images with inconsistent predictions, and CCL regularization, which facilitates the learning of discriminative feature representations for unlabeled images in the training procedure.

The proposed method PCML exhibits promising classification results across five real-world interior decoration style image datasets. Compared to the baseline method DenseNet, it demonstrates average improvements of 4.45%, 5.36%, 10.9%, 3.85%, 11.0%, and 11.6% on AUC, ACC, SEN, SPE, PREC, and F1, respectively. Specifically, PCML can achieve consistent improvements over all the comparison methods across all metrics on the five image datasets. For the TV background wall dataset, PCML surpasses the classification performance of the most competitive method SRC-MT by 0.98%, 2.22%, 4.01%, 1.4%, 5.94%, and 5.6% on AUC, ACC, SEN, SPE, PREC, and F1, respectively. For the chandelier dataset, compared to the most competitive method FixMatch, PCML achieved the improvements of 0.59%, 1.82%, 4.02%, 1.58%, 4.93%, and 4.44%, respectively. For the living room dataset, PCML surpasses the classification performance of the most competitive method CoMatch by 2.34%, 1.6%, 2.58%, 1.16%, 2.2%, and 2.68% on AUC, ACC, SEN, SPE, PREC, and F1, respectively. For the dining room dataset, compared to the most competitive method SRC-MT, PCML achieved the improvements of 1.78%, 2.4%, 3.78%, 0.66%, 8.49%, and 8.06%, respectively. In addition, for the bedroom dataset, PCML surpasses the classification performance of the most competitive method MixMatch by 2.67%, 0.58%, 3.84%, 0.13%, 1.51%, and 2.51% on AUC, ACC, SEN, SPE, PREC, and F1, respectively. It is noteworthy to observe that the classification performance of our PCML surpasses the upper-bound performance on some evaluative metrics. For example, on TV background wall dataset, PCML outperforms the fully supervised DenseNet with 100% labeled data in terms of

PREC metric. In addition, the classification results of PCML closely approaches the upper-bound performance. The disparities between certain metrics are minimal, with less than 1% differences, such as the SPE and F1 classification metrics for the TV background wall dataset. This observation further validates the advantages gained by reviewing the labeled imaged with inconsistent predictions and learning discriminative feature representations of unlabeled images, which also explains our significant performance gains.

4.4. Analysis of the Proposed PCML Framework

4.4.1. Efficacy of Different Components

To obtain a better insight into the performance of the proposed PCML method, we conducted an ablation study to investigate the impact of different components in PCML. We listed the classification results of ablation study on the TV background wall dataset and the living room dataset, as shown in Tables 6 and 7. The ablation study includes the following models: (a) backbone network ResNet (Baseline 1); (b) backbone network DenseNet (Baseline 2); (c) subnetworks ResNet and DenseNet with CPS regularization (Scenario 1); (d) subnetworks ResNet and DenseNet with CPS and ICR regularizations (Scenario 2); (e) subnetworks ResNet and DenseNet with ICR and CCL regularizations (Scenario 3, PCML).

Table 6. Ablation study of different components on the segmentation performance of the TV background wall dataset.

ResNet	DenseNet	CPS	ICR	CCL	TV Background Wall Dataset					
					AUC (%)↑	ACC (%)↑	SEN (%)↑	SPE (%)↑	PREC (%)↑	F1 (%)↑
✓					93.31	89.30	76.39	91.22	75.50	75.58
	✓				93.76	88.53	73.84	91.45	72.73	72.90
✓	✓	✓			96.17	91.44	79.42	93.00	81.84	79.52
✓	✓	✓	✓		96.40	92.28	83.15	93.29	81.94	82.40
✓	✓	✓	✓	✓	**96.82**	**93.58**	**84.12**	**94.48**	**87.27**	**85.36**

Table 7. Ablation study of different components on the segmentation performance of the living room dataset.

ResNet	DenseNet	CPS	ICR	CCL	TV Background Wall Dataset					
					AUC (%)↑	ACC (%)↑	SEN (%)↑	SPE (%)↑	PREC (%)↑	F1 (%)↑
✓					91.06	87.88	70.41	90.89	68.94	67.81
	✓				89.74	86.95	69.34	89.78	69.45	67.28
✓	✓	✓			92.35	88.97	70.71	91.88	78.68	68.26
✓	✓	✓	✓		93.91	88.64	72.41	90.90	79.50	72.33
✓	✓	✓	✓	✓	**93.92**	**91.25**	**77.27**	**93.33**	**79.74**	**76.74**

As shown in Tables 6 and 7, Scenario 1 yields better classification performance than Baseline 1 and Baseline 2, which demonstrates the effectiveness of different subnetworks with CPS regularization. The architecture of different subnetworks can promote the diversified predictions, thereby preventing the two subnetworks from collapsing into each other. Scenario 2 can obtain better classification performance than Scenario 1, which verifies ICR regularization can conduct review training on the potential mis-predicted labeled images that enables the SSL model to leverage the limited reliable knowledge underlying labeled images. Moreover, compared to Scenario 2, Scenario 3 (PCML) demonstrates further enhancement in classification performance. These results underscore the importance of selectively utilizing more reliable unlabeled images for contrastive learning, which contributes to the overall robustness of the model.

In addition, we used Gradient Weighted Class Activation Mapping (Grad-CAM) [47] to visualize and localize the salient image regions that exert a substantial influence on the model's prediction score for a given class. To provide better interpretability of the proposed framework, we visualized Grad-CAMs of Scenario 1, Scenario 2, and PCML. As illustrated in Figures 4 and 5, the first row represents the Chinese style in the TV background wall

dataset and the living room dataset, respectively. The second row represents the European style in the TV background wall dataset and the living room dataset, respectively. We listed the original images in Figures 4a and 5a. In addition, Figures 4b and 5b display the Grad-CAM visualizations of Scenario 1. Figures 4c and 5c display the Grad-CAM visualizations of Scenario 2. Figures 4d and 5d display the Grad-CAM visualizations of PCML. The visualizations show that the proposed PCML method can learn the features of salient regions exhibiting style changes and focuses on the distinctive features associated with specific decoration style.

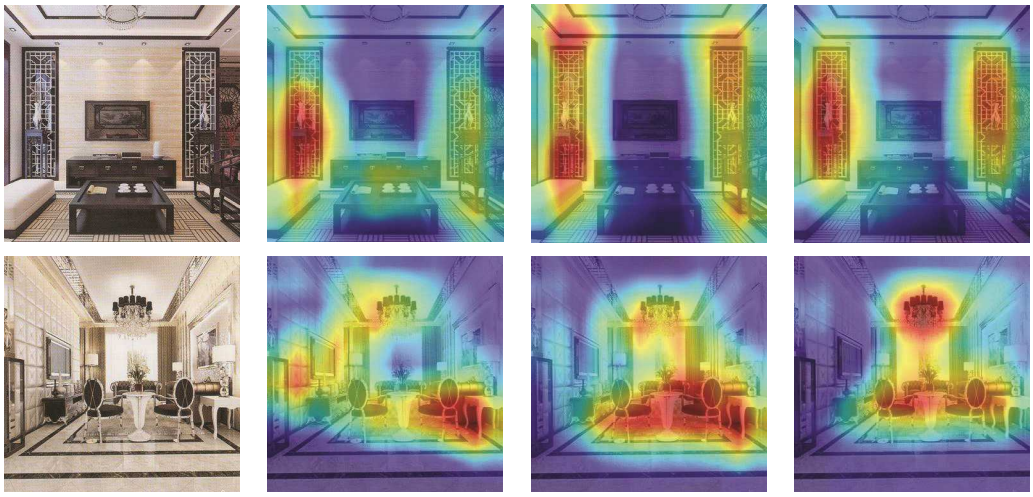


Figure 4. Grad-CAMs visualization of interior decoration style image attention regions from the TV background wall dataset: (**a**) original images; (**b**) Scenario 1; (**c**) Scenario 2; (**d**) Scenario 3. The first row denotes Chinese style and the second row denotes European style.

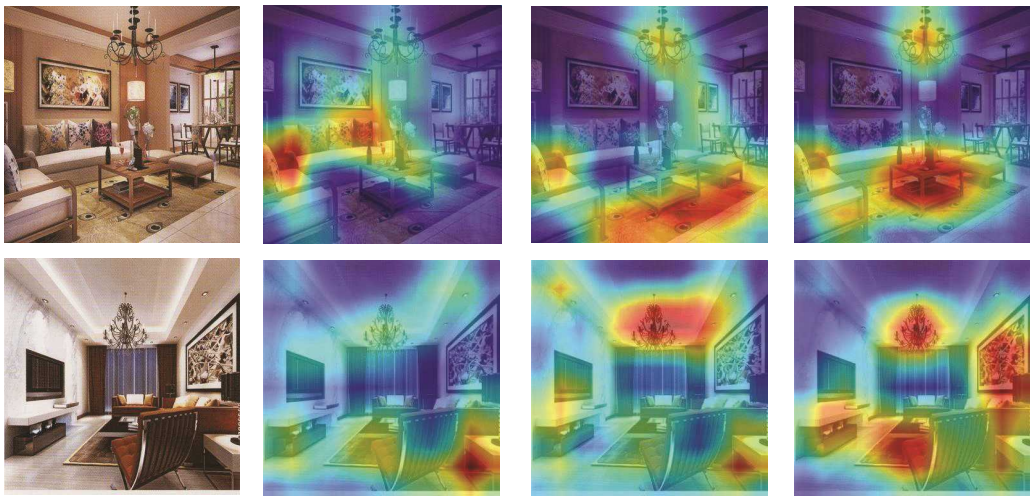


Figure 5. Grad-CAMs visualization of interior decoration style image attention regions from the Living room dataset: (**a**) Original images; (**b**) Scenario 1; (**c**) Scenario 2; (**d**) Scenario 3. The first row denotes Chinese style and the second row denotes European style.

4.4.2. Impact of Hyper-Parameters

We further investigated the influence of hyper-parameters on the classification performance of the proposed PCML. PCML contains three hyper-parameters—T, λ_u, and λ_c—where T is used to filter out the noise unlabeled images, λ_u is the weight for cross pseudo supervision loss, and λ_c is the weight for CCL regularization. In the experiment, we fixed two of the parameters and varied the other parameter to observe the classification

performance of PCML. The classification results were given in Table 8. The threshold T was used to filter out some potentially unreliable unlabeled images with low confidence. When $\lambda_u = 0.5$, $\lambda_c = 0.1$, with the increase in T, the classification performance is improved accordingly, demonstrating that threshold T can effectively filter out unreliable unlabeled images. As T keeps increasing, the classification performance degrades, which means that some reliable unlabeled images were excluded. In addition, for parameters λ_u and λ_c, we can observe the similar phenomena, as shown in Table 8. In total, PCML exhibits stable predictions despite variations in both parameters λ_u and λ_c, which demonstrates the effectiveness of the proposed pseudo-label-guided contrastive mutual learning framework.

Table 8. Sensitivity of trade-off coefficients on the classification performance (%) of the TV background wall dataset.

$\boldsymbol{\lambda_u = 0.5}$						**Metrics**
$\boldsymbol{\lambda_c = 0.1}$	**AUC**	**ACC**	**SEN**	**SPE**	**PREC**	**F1**
$T = 0.5$	96.24	91.28	78.46	92.58	82.79	79.37
$T = 0.6$	96.12	91.97	82.01	92.58	82.55	81.71
$T = 0.7$	96.50	91.97	80.21	92.89	84.75	81.86
$T = 0.8$	**96.82**	**93.58**	**84.12**	**94.48**	**87.27**	**85.36**
$T = 0.9$	96.51	91.67	81.17	93.00	82.19	80.74
$\boldsymbol{T = 0.8}$						**Metrics**
$\boldsymbol{\lambda_c = 0.1}$	**AUC**	**ACC**	**SEN**	**SPE**	**PREC**	**F1**
$\lambda_u = 0.05$	95.31	90.37	78.91	91.62	78.53	78.23
$\lambda_u = 0.1$	95.59	91.67	81.53	92.18	82.20	81.46
$\lambda_u = 0.3$	96.50	92.81	82.24	93.88	84.80	82.79
$\lambda_u = 0.5$	**96.82**	**93.58**	**84.12**	**94.48**	**87.27**	**85.36**
$\lambda_u = 0.7$	95.94	92.97	82.06	94.44	83.84	82.34
$\boldsymbol{T = 0.8}$						**Metrics**
$\boldsymbol{\lambda_u = 0.5}$	**AUC**	**ACC**	**SEN**	**SPE**	**PREC**	**F1**
$\lambda_c = 0.05$	96.30	92.28	80.61	93.18	85.32	82.11
$\lambda_c = 0.1$	**96.82**	**93.58**	**84.12**	**94.48**	**87.27**	**85.36**
$\lambda_c = 0.3$	96.51	92.58	82.24	93.87	82.91	82.26
$\lambda_c = 0.5$	95.23	91.82	81.19	93.70	81.03	80.61
$\lambda_c = 0.7$	94.95	90.67	79.40	92.97	78.49	78.05

4.4.3. Impact of Input Noise

To evaluate the influence of noise on the classification performance of the proposed PCML framework, we utilize different kinds of input image perturbations as input noise to evaluate the performance of our PCML. Specifically, random transformations including rotation, translation, and horizontal flips were applied to each input image. The rotation angle was randomly set in the range of -10 to 10 degrees. Horizontal and vertical translations for pixels were applied within a range of -2% to 2% of the image width. Additionally, the input image was randomly flipped horizontally and vertically with a probability of 50%. We use the above input noise into one subnetwork and two subnetworks to evaluate the performance of the proposed method. It includes the following scenarios: (a) without input noise (setting 1); (b) with input noise to one subnetwork (setting 2); (c) with input noise to two subnetworks (setting 3). As illustrated in Table 9, the experimental results across different settings show minimal variation, which indicates that the proposed PCML is robust to the input perturbation noise.

Table 9. Impact of input perturbation noise on the classification results (%) obtained by PCML.

Settings	TV Background Wall					
	AUC	**ACC**	**SEN**	**SPE**	**PREC**	**F1**
1	96.82	93.58	84.12	94.48	87.27	85.36
2	96.24	92.13	80.82	93.47	82.63	81.02
3	96.09	91.67	81.93	93.26	80.11	80.64
Settings	**Chandelier**					
	AUC	**ACC**	**SEN**	**SPE**	**PREC**	**F1**
1	98.25	95.21	90.88	96.65	90.47	90.55
2	97.96	94.43	89.38	96.29	89.63	89.11
3	98.90	95.47	91.73	96.71	90.16	90.78
Settings	**Living Room**					
	AUC	**ACC**	**SEN**	**SPE**	**PREC**	**F1**
1	93.92	91.25	77.27	93.33	79.74	76.74
2	93.34	90.91	78.47	93.03	79.02	77.58
3	93.16	91.25	79.19	93.12	79.07	78.26
Settings	**Dining Room**					
	AUC	**ACC**	**SEN**	**SPE**	**PREC**	**F1**
1	97.23	93.51	88.96	94.75	86.35	87.59
2	97.84	94.71	88.08	95.46	89.71	88.68
3	98.15	93.99	85.16	94.30	87.59	85.86
Settings	**Bedroom**					
	AUC	**ACC**	**SEN**	**SPE**	**PREC**	**F1**
1	86.71	82.42	69.26	86.80	65.10	66.52
2	87.68	82.03	67.81	87.7	70.43	63.14
3	88.32	82.23	68.81	86.98	70.58	64.37

5. Conclusions

In this paper, we propose a pseudo-label-guided contrastive mutual learning framework, named PCML, to facilitate semi-supervised interior decoration style classification by harnessing reliable knowledge within limited labeled data and selectively utilizing the reliable unlabeled data for learning discriminative feature representations. The framework employs two subnetworks with different architectures, and integrates two novel modules: inconsistency-aware relearning regularization, and class-aware contrastive learning regularization. The different subnetworks can direct the model generate diverse predictions. Thus, inconsistency-aware relearning regularization to can perform a review training for these images with different predictions. In addition, class-aware contrastive learning regularization can learn the discriminative feature representations of unlabeled images using the corresponding reliable pseudo-labels. More importantly, the synergistic learning among the mutual learning framework, inconsistency-aware relearning regularization, and class-aware contrastive learning regularization during the training process enables each subnetwork to selectively incorporate the reliable knowledge imparted by the other subnetwork, thereby mitigating the issue of confirmation bias. The comprehensive performance evaluations on multiple interior decoration style image datasets demonstrate the superiority of the proposed PCML over existing SSL methods. In future work, we plan to further comprehensively evaluate our PCML on more complex interior decoration tasks.

Author Contributions: Conceptualization, W.H.; methodology, K.B. and H.Z.; software, K.B.; validation, L.G., S.L. and J.S.; data curation, Q.X. and X.S.; writing—original draft preparation, L.G. and K.B.; writing—review and editing, W.H.; supervision, L.G. and W.H.; funding acquisition, W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under Grant (23KJB520012).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PCML	pseudo-label-guided contrastive mutual learning
ICR	inconsistency-aware relearning
CCL	class-aware contrastive learning
SSL	semi-supervised learning
MT	mean teacher
Grad-CAM	Gradient Weighted Class Activation Mapping

References

1. Liu, S.; Bo, Y.; Huang, L. Application of image style transfer technology in interior decoration design based on ecological environment. *J. Sens.* **2021**, *2021*, 9699110. [CrossRef]
2. Xu, J.; Li, M.; Huang, D.; Wei, Y.; Zhong, S. A comparative study on the influence of different decoration styles on subjective evaluation of hotel indoor environment. *Buildings* **2022**, *12*, 1777. [CrossRef]
3. Weiss, T.; Yildiz, I.; Agarwal, N.; Ataer-Cansizoglu, E.; Choi, J.W. Image-Driven Furniture Style for Interactive 3D Scene Modeling. In Proceedings of the Computer Graphics Forum, Geneva, Switzerland, 20–23 October 2020; Wiley Online Library: Hoboken, NJ, USA, 2020; Volume 39, pp. 57–68.
4. Kim, J.; Lee, J.K. Stochastic detection of interior design styles using a deep-learning model for reference images. *Appl. Sci.* **2020**, *10*, 7299. [CrossRef]
5. Tian, J.; Zakaria, S.A. Application of Image Classification Algorithm Based on Deep Learning in Residential Interior Design Style Recognition. *Rev. Ibér. Sist. Tecnol. Inf.* **2023**, *E63*, 340–352.
6. Tong, H.; Wan, Q.; Kaszowska, A.; Panetta, K.; Taylor, H.A.; Agaian, S. ARFurniture: Augmented reality interior decoration style colorization. *Electron. Imaging* **2019**, *31*, 1–9. [CrossRef]
7. Wu, Z.; Jia, X.; Jiang, R.; Ye, Y.; Qi, H.; Xu, C. CSID-GAN: A Customized Style Interior Floor Plan Design Framework Based on Generative Adversarial Network. *IEEE Trans. Consum. Electron.* **2024**. [CrossRef]
8. Yu, L.; Wang, S.; Li, X.; Fu, C.W.; Heng, P.A. Uncertainty-aware self-ensembling model for semi-supervised 3D left atrium segmentation. In Proceedings of the Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, 13–17 October 2019; Proceedings, Part II 22; Springer: Berlin/Heidelberg, Germany, 2019; pp. 605–613.
9. Su, J.; Luo, Z.; Lian, S.; Lin, D.; Li, S. Mutual learning with reliable pseudo label for semi-supervised medical image segmentation. *Med. Image Anal.* **2024**, *94*, 103111. [CrossRef]
10. Sajjadi, M.; Javanmardi, M.; Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Adv. Neural. Inf. Process. Syst.* **2016**, *29*, 1–9.
11. Iscen, A.; Tolias, G.; Avrithis, Y.; Chum, O. Label propagation for deep semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5070–5079.
12. Chen, X.; Yuan, Y.; Zeng, G.; Wang, J. Semi-supervised semantic segmentation with cross pseudo supervision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2613–2622.
13. Ouali, Y.; Hudelot, C.; Tami, M. Semi-supervised semantic segmentation with cross-consistency training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12674–12684.
14. Liu, Y.; Tian, Y.; Chen, Y.; Liu, F.; Belagiannis, V.; Carneiro, G. Perturbed and strict mean teachers for semi-supervised semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4258–4267.
15. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–10.
16. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised contrastive learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18661–18673.

17. Zhong, Y.; Yuan, B.; Wu, H.; Yuan, Z.; Peng, J.; Wang, Y.X. Pixel contrastive-consistent semi-supervised semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 7273–7282.
18. Rasmus, A.; Berglund, M.; Honkala, M.; Valpola, H.; Raiko, T. Semi-supervised learning with ladder networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9.
19. Miyato, T.; Maeda, S.I.; Koyama, M.; Ishii, S. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1979–1993. [CrossRef] [PubMed]
20. Luo, Y.; Zhu, J.; Li, M.; Ren, Y.; Zhang, B. Smooth neighbors on teacher graphs for semi-supervised learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8896–8905.
21. Hang, W.; Feng, W.; Liang, S.; Yu, L.; Wang, Q.; Choi, K.S.; Qin, J. Local and global structure-aware entropy regularized mean teacher model for 3d left atrium segmentation. In Proceedings of the Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, 4–8 October 2020; Proceedings, Part I 23; Springer: Berlin/Heidelberg, Germany, 2020; pp. 562–571.
22. Liu, Q.; Yu, L.; Luo, L.; Dou, Q.; Heng, P.A. Semi-supervised medical image classification with relation-driven self-ensembling model. *IEEE Trans. Med Imaging* **2020**, *39*, 3429–3440. [CrossRef] [PubMed]
23. Grandvalet, Y.; Bengio, Y. Semi-supervised learning by entropy minimization. *Adv. Neural Inf. Process. Syst.* **2004**, *17*, 529–536.
24. Lee, D.H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Proceedings of the Workshop on Challenges in Representation Learning, ICML, Atlanta, GA, USA, 16–21 June 2013; Volume 3, p. 896.
25. Li, Y.; Chen, J.; Xie, X.; Ma, K.; Zheng, Y. Self-loop uncertainty: A novel pseudo-label for semi-supervised medical image segmentation. In Proceedings of the Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, 4–8 October 2020; Proceedings, Part I 23; Springer: Berlin/Heidelberg, Germany, 2020; pp. 614–623.
26. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–11.
27. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 596–608.
28. Kalluri, T.; Varma, G.; Chandraker, M.; Jawahar, C. Universal semi-supervised semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5259–5270.
29. Zhang, B.; Wang, Y.; Hou, W.; Wu, H.; Wang, J.; Okumura, M.; Shinozaki, T. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 18408–18419.
30. Rizve, M.N.; Duarte, K.; Rawat, Y.S.; Shah, M. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv* **2021**, arXiv:2101.06329.
31. Wu, Y.; Xu, M.; Ge, Z.; Cai, J.; Zhang, L. Semi-supervised left atrium segmentation with mutual consistency training. In Proceedings of the Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, 27 September–1 October 2021; Proceedings, Part II 24; Springer: Berlin/Heidelberg, Germany, 2021; pp. 297–306.
32. Li, Y.; Wang, X.; Yang, L.; Feng, L.; Zhang, W.; Gao, Y. Diverse cotraining makes strong semi-supervised segmentor. *arXiv* **2023**, arXiv:2308.09281.
33. Wang, Y.; Xiao, B.; Bi, X.; Li, W.; Gao, X. Mcf: Mutual correction framework for semi-supervised medical image segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 15651–15660.
34. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 1597–1607.
35. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 9729–9738.
36. Henaff, O. Data-efficient image recognition with contrastive predictive coding. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 4182–4192.
37. Wu, Z.; Xiong, Y.; Yu, S.X.; Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3733–3742.
38. Wang, X.; Gao, J.; Long, M.; Wang, J. Self-tuning for data-efficient deep learning. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 10738–10748.
39. Hang, W.; Huang, Y.; Liang, S.; Lei, B.; Choi, K.S.; Qin, J. Reliability-aware contrastive self-ensembling for semi-supervised medical image classification. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Singapore, 18–22 September 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 754–763.
40. Zhang, Y.; Zhang, X.; Li, J.; Qiu, R.C.; Xu, H.; Tian, Q. Semi-supervised contrastive learning with similarity co-calibration. *IEEE Trans. Multimed.* **2022**, *25*, 1749–1759. [CrossRef]
41. Li, J.; Xiong, C.; Hoi, S.C. Comatch: Semi-supervised learning with contrastive graph regularization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 9475–9484.

42. Lee, D.; Kim, S.; Kim, I.; Cheon, Y.; Cho, M.; Han, W.S. Contrastive regularization for semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3911–3920.
43. Yang, F.; Wu, K.; Zhang, S.; Jiang, G.; Liu, Y.; Zheng, F.; Zhang, W.; Wang, C.; Zeng, L. Class-aware contrastive semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 14421–14430.
44. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
45. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
46. Berthelot, D.; Carlini, N.; Cubuk, E.D.; Kurakin, A.; Sohn, K.; Zhang, H.; Raffel, C. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv* **2019**, arXiv:1911.09785.
47. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

mathematics

MDPI

Article

Advancing AI-Driven Linguistic Analysis: Developing and Annotating Comprehensive Arabic Dialect Corpora for Gulf Countries and Saudi Arabia

Nouf Al-Shenaifi, Aqil M. Azmi * and Manar Hosny

Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; noalshenaifi@ksu.edu.sa (N.A.-S.); mifawzi@ksu.edu.sa (M.H.)
* Correspondence: aqil@ksu.edu.sa

Abstract: This study harnesses the linguistic diversity of Arabic dialects to create two expansive corpora from X (formerly Twitter). The Gulf Arabic Corpus (GAC-6) includes around 1.7 million tweets from six Gulf countries—Saudi Arabia, UAE, Qatar, Oman, Kuwait, and Bahrain—capturing a wide range of linguistic variations. The Saudi Dialect Corpus (SDC-5) comprises 790,000 tweets, offering in-depth insights into five major regional dialects of Saudi Arabia: Hijazi, Najdi, Southern, Northern, and Eastern, reflecting the complex linguistic landscape of the region. Both corpora are thoroughly annotated with dialect-specific seed words and geolocation data, achieving high levels of accuracy, as indicated by Cohen's Kappa scores of 0.78 for GAC-6 and 0.90 for SDC-5. The annotation process leverages AI-driven techniques, including machine learning algorithms for automated dialect recognition and feature extraction, to enhance the granularity and precision of the data. These resources significantly contribute to the field of Arabic dialectology and facilitate the development of AI algorithms for linguistic data analysis, enhancing AI system design and efficiency. The data provided by this research are crucial for advancing AI methodologies, supporting diverse applications in the realm of next-generation AI technologies.

Keywords: Arabic dialects; Arabic corpora; Twitter; dialect identification; lexicon

MSC: 68T50; 68T37; 62H30; 91B74

1. Introduction

Arabic is not only a major world language, spoken natively by approximately three hundred million people primarily in the Middle East and North Africa (MENA), but also the liturgical language of two billion Muslims globally. It features one of the most widely used writing systems in the world. This script transcends its native speakers, extending throughout the Islamic world, as it is employed to write the Qur'an, the holy book of Muslims [1].

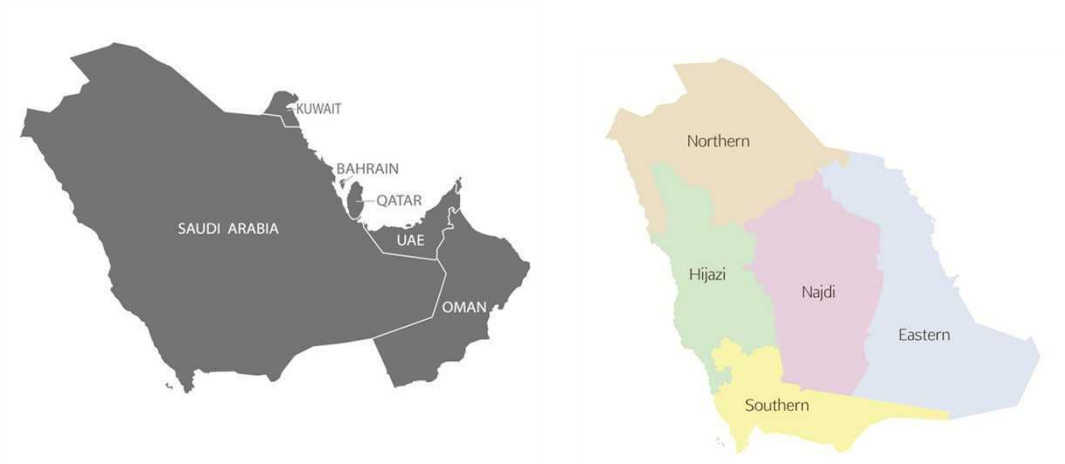
In the contemporary landscape of the Arab region, there has been a marked escalation in the deployment of Arabic dialects for informal written communication and interactions on social media platforms [2–6]. This trend has led to an exponential growth in the quantity of Dialectal Arabic content across these digital venues, particularly on social media, as evidenced by recent studies [7–10]. Consequently, this surge has catalyzed considerable scholarly interest among researchers in Arabic Natural Language Processing (NLP). There is a robust initiative underway to cultivate annotated linguistic resources specifically designed for these dialects. The primary objectives of this initiative are to enrich our understanding of the linguistic intricacies of Dialectal Arabic (DA) [11] and to accelerate the advancement of specialized tools and applications for its processing [12,13].

Despite the broad spectrum of tools and resources dedicated to Arabic NLP, the primary emphasis remains on Modern Standard Arabic (MSA), the lingua franca language

universally utilized throughout the Arab world, which is deeply rooted in Classical Arabic [14–23]. However, the adaptation of these NLP resources to DA introduces significant challenges. These challenges stem from the substantial linguistic divergences between the various dialects and MSA [24,25]. This situation highlights a critical gap in the applicability of existing tools to the linguistic realities of the Arab region, underlining the need for targeted research and development efforts in the field of NLP to bridge these disparities.

Arabic dialects demonstrate significant divergences from MSA in several linguistic domains, including morphology, phonology, and syntax [4,26,27]. Furthermore, the lexicon across various Arabic dialects varies considerably, and most dialects do not employ standardized orthographies [6,28]. While the development of resources for NLP tasks in DA remains nascent in comparison to MSA [29,30], there are ongoing initiatives to construct dialect-specific tools and resources. These efforts include the creation of annotated corpora and morphological analyzers tailored to particular dialects [31]. Notably, the development of NLP tools for the Egyptian and Levantine dialects has progressed more substantially [32–34]. Despite the extensive presence of dialectal content online, Gulf Arabic still experiences a significant deficiency in NLP tools and resources, highlighting a critical area for further linguistic research [4,6,35,36].

Arabic dialects are predominantly classified into several principal groups based on geographical regions, including Gulf, Egyptian, Levantine, North African (Maghrebi), Iraqi, Yemeni, and Sudanese dialects [37–40]. The Gulf dialect comprises the linguistic variants spoken in countries adjacent to the Arabian Gulf, such as Saudi Arabia, Kuwait, Qatar, United Arab Emirates (UAE), Bahrain, and Oman, see Figure 1a. The Egyptian dialect encompasses the linguistic varieties found primarily in Egypt and select areas of Sudan. The Levantine dialect is primarily spoken in the Levant region, including Palestine, Syria, Lebanon, and Jordan. The Maghrebi dialect includes a range of dialects spoken across North Africa, excluding Egypt, and covers countries such as Morocco, Algeria, Tunisia, and Libya. The Iraqi dialect refers to the linguistic variety prevalent in Iraq [41], and the Yemeni dialect is used in Yemen. In Saudi Arabia, a country characterized by its substantial geographical diversity, there is a significant variation in linguistic dialects across different regions; see Figure 1b [4,42]. Historically, there has been a tendency to not recognize the Saudi dialect as a distinct linguistic entity, typically categorizing it within the broader Gulf dialects [43].



(**a**) Arabian Gulf countries (**b**) Saudi Arabia

Figure 1. The regional dialectal map displays: (**a**) the six countries of the Arabian Gulf, and (**b**) the distinct dialectal subregions of Saudi Arabia.

The profound social and political transformations occurring in the Gulf region, particularly in Saudi Arabia, underscore the need for a dedicated Saudi Dialect Corpus. Saudi Arabia, characterized by its conservative and autocratic nature, is home to a predominantly young population, with about 50% under the age of 25. The transformative government-

sponsored reforms have sparked significant debates within the society, prominently featured on social media platforms such as Twitter due to restrictions on public debate and protests [10].

The need for a dedicated corpus is exemplified by the intense discussions on Twitter that followed the Saudi government's decision to lift the driving ban for women in September 2017. This landmark policy shift ignited widespread debates and garnered support, as documented in [44]. Researchers gathered and analyzed tweets and hashtags in the local Saudi dialect from the initial days after the ban was lifted, providing valuable insights into public sentiment on this crucial issue. This work underscores the necessity of a specialized corpus that accurately captures the unique linguistic nuances of the Saudi dialect, which is vital for precise sentiment analysis and in-depth cultural research.

A proper Saudi Dialect Corpus would thus not only facilitate a deeper understanding of the public discourse and sentiment in the region but also enhance the development of tailored NLP tools. These tools are essential for accurately interpreting and responding to the nuanced language used in social media, which is often imbued with cultural and regional specificities not covered by standard Arabic corpora. This makes the development and availability of a dedicated Saudi Dialect Corpus crucial for researchers working on Arabic NLP, especially in applications involving sentiment analysis, social monitoring, and cultural studies.

In Saudi Arabia, the dialectical landscape is distinguished by the prevalence of specific regional dialects: Hijazi in the western region, Najdi in the central region, Southern dialect in the southern territories, Northern dialect in the northern areas, and Eastern dialect in the eastern part of the Arabian peninsula. This research paper outlines our efforts in constructing an extensively annotated corpus of Gulf dialectical Arabic. We utilized data sourced from X, formerly known as Twitter (throughout this paper, we will continue to use the terms "Twitter" and "tweets", as there are no widely accepted substitutes for the latter). The data extraction process was improved by incorporating advanced techniques for textual analysis, notably feature extraction. This approach leveraged geographical metadata embedded within user profiles to ensure the accurate representation of diverse dialectical variations across the specified regions. This methodology enhances the precision and robustness of our data processing and annotation efforts, contributing significantly to the depth and quality of linguistic analysis. These techniques are crucial for advancing AI-driven frameworks, optimizing data utilization, and enhancing the reliability of AI applications across various sectors.

This study introduces two pivotal resources: the Gulf Arabic Corpus (GAC-6) and the Saudi Dialect Corpus (SDC-5). The GAC-6 is an expansive corpus comprising approximately 1.7 million Arabic tweets that reflect the dialectical nuances of the Gulf region, encompassing Saudi Arabia, Bahrain, Kuwait, Oman, Qatar, and the Emirates. Concurrently, the SDC-5 emerges as a comprehensive annotated corpus dedicated to Saudi dialects, encapsulating linguistic variations across five key Saudi regions: Hijazi, Najdi, Southern, Northern, and Eastern. Our methodology includes the meticulous manual annotation of a data subset with specific dialect labels, establishing a gold standard for dialect identification. The resources elucidated in this paper hold significant potential not only for advancing NLP applications such as machine translation but also for facilitating in-depth linguistic analysis of Gulf and Saudi dialects. This contributes to a broader understanding of regional linguistic diversity, aiding both academic research and practical applications in computational linguistics.

The objective of this research is to augment the domain of language resources by developing comprehensive linguistic datasets for both Gulf and Saudi dialects. This initiative leverages the substantial repository of Arabic textual content available on social media platforms, with a specific focus on Twitter. The contributions of this paper are manifold and include the following:

- The development of an extensive corpora of Gulf and Saudi dialects, which are sourced from Twitter and automatically tagged with dialect labels. These corpora are de-

signed to significantly support and enable a wide range of Arabic NLP research endeavors in the future.
- The employment of native speakers for the manual annotation of the segments of the datasets, thereby validating the accuracy of the automatically assigned dialect labels. This step ensures that the linguistic characteristics of each dialect are accurately captured.
- The evaluation of the corpora's quality through the measurement of inter-annotator agreement, quantified using the Kappa statistic. This assessment ensures the reliability and consistency of the annotation process, confirming that the data are robust for academic and practical applications in computational linguistics.

The structure of this paper is organized as follows: Section 2 reviews the relevant literature and related studies. Section 3 details the methodology used in compiling, preprocessing, and annotating the corpora. Section 4 provides an in-depth discussion of the compiled corpora, including general statistics, and assesses the quality of the annotations. Finally, Section 5 offers concluding remarks and outlines potential directions for future research.

2. Related Work

This section provides an overview of the literature pertaining to Arabic dialectal corpora, emphasizing the significant contributions and key findings within this research domain.

The exploration of Arabic dialects has garnered considerable interest in recent years. A myriad of Arabic dialectal corpora has been developed, serving as invaluable resources for the study and linguistic analysis of these dialects. In recent times, there has been an augmented effort toward the aggregation of datasets and the formulation of comprehensive corpora from a variety of sources, thereby facilitating in-depth investigations into Arabic dialectology.

A seminal piece of research in the realm of corpus development was conducted by [45], who introduced a pioneering dataset dedicated to Dialectal Arabic, known as the Arabic Online Commentary (AOC) Dataset. The AOC Dataset represents the inaugural dialectal corpus made available to the academic community, comprising approximately 52 million words derived from the comment sections of online Arabic news platforms.

Several studies, such as those documented by [46], have utilized datasets that were manually annotated through crowdsourcing efforts. In particular, Alsarsour et al. [46] introduced the Dialectal Arabic Tweets (DART) dataset, a collection comprising approximately 25,000 Arabic tweets that were manually annotated via crowdsourcing. This dataset is characterized by its balanced representation across five major Arabic dialect groups: Egyptian, Maghrebi, Levantine, Gulf, and Iraqi. Furthermore, Zaghouani and Charfi [47] developed a multidialectal corpus, also annotated manually through crowdsourcing, which includes around 2.4 million tweets originating from 11 Arab regions, including North Levant, South Levant, Egypt, Gulf, Morocco, Tunisia, Algeria, Yemen, Iraq, Libya, and Sudan. Sadat et al. [48] contributed a sentence-level manually annotated dataset containing about 62,000 sentences sourced from online blogs across various Arab nations.

Khalifa et al. [33] compiled a substantial Gulf Arabic Corpus, consisting of 1200 forum novels, with annotations at the document level derived from the novels' titles and authors' names. Alshutayri and Atwell [49] outlined the methodology for constructing an Arabic dialect text corpus from social media platforms like Twitter and Facebook, as well as comments from newspapers, with each entry receiving a dialect tag through crowdsourcing. Some scholars have also embarked on creating parallel corpora, which consist of sentences translated into Arabic dialects from other datasets. An exemplar of such a corpus is presented in [50], where the authors unveiled the Multidialectal Parallel Corpus of Arabic (MPCA), a compilation of approximately 2000 sentences that were manually translated into various dialects from Egyptian Arabic.

Moreover, Bouamor et al. [2] introduced the Multi Arabic Dialect Applications and Resources (MADAR) corpus, a parallel corpus encompassing 25 dialects from Arabic-

speaking cities across 15 Arab countries. This corpus was developed through the manual translation of selected sentences from the Basic Travel Expression Corpus (BTEC) [51]. Notably, the corpus lacks representation of certain Gulf Arabic dialects, specifically those from Bahrain, Kuwait, and the United Arab Emirates.

Other studies, such as the next one, have employed a semi-automatic approach for corpus annotation. Mubarak and Darwish [52] developed a multidialectal Twitter corpus of Arabic, consisting of 6.5 million tweets. This corpus is annotated based on distinct dialects and the geographical locations of users. Similarly, Abu Kwaik et al. [24] introduced the Shami Dialects Corpus (SDC), which encompasses four dialects (Palestinian, Jordanian, Lebanese, and Syrian) and contains approximately 118,000 sentences. This corpus is compiled from Arabic tweets, with automatic annotations derived from the Twitter API's geographical location feature, supplemented by manual annotations for web content.

Furthermore, there are datasets that have been collected and labeled entirely through automated processes. Abdul-Mageed et al. [3] unveiled a vast corpus of tweets representing city-level dialects from 29 Arab cities across 10 Arab countries, with diverse dialectical features. The annotation for this corpus was conducted automatically using a Python geocoding library. Additionally, Abdelali et al. [25] created a balanced, non-genre-specific, country-level Arabic dialectal tweet corpus. This corpus was generated using a series of filters; user accounts were selected based on country-specific keywords, and tweets were further filtered to exclude users predominantly using MSA. The final corpus comprises 540,000 tweets from 2525 Twitter users, along with a test set consisting of 182 tweets per country, which were manually classified by native Arabic speakers.

Conversely, certain dialectal corpora have concentrated on specific dialects, with a focus on morphological annotation. An exemplar is the work by [53], who introduced the Saudi corpus for NLP Applications and Resources (SUAR), targeting the Saudi dialect. This corpus encompasses 104,000 words sourced from various online social media platforms, and it underwent morphological annotation via the MADAMIRA tool [54]. This initial automated annotation was subsequently subjected to manual review to ensure accuracy and validate the analysis.

Alowisheq et al. [55] unveiled the Multi-domain Arabic Resources for Sentiment Analysis (MARSA), a sentiment-annotated corpus specific to the Gulf dialect. This corpus comprises 61,000 tweets, each manually annotated with sentiment labels by two independent annotators, ensuring the reliability of sentiment assessment.

Further, Elgibreen et al. [56] introduced the King Saud University Saudi Corpus (KSUSC), a comprehensive new corpus containing over 161 million sentences harvested from a variety of sources. While this corpus is extensive and spans multiple domains, it lacks annotations, presenting a vast but unstructured resource for linguistic analysis. This study also conducted a review of existing Arabic corpora, highlighting the need for more in-depth research into corpora representing the Saudi dialect.

Moreover, Alruily [57] developed a dialectal Saudi Twitter corpus and provided an analysis of its linguistic peculiarities, such as compounding, abbreviation, spelling discrepancies, and the emergence of neologisms, shedding light on the unique challenges associated with processing this dialect. Lastly, Al-Ghadir and Azmi [58] capitalized on the vibrant social media environment of Saudi Arabia to investigate the posting patterns of local users, delineating these behaviors by gender and educational attainment. Concentrating on author profiling in this milieu, this study provides an understanding of demographic trends, thereby enhancing the comprehension of dialectical subtleties as manifested on social media platforms.

3. Our Methodology

Social media platforms and microblogging sites, notably Twitter, have emerged as significant repositories of natural language textual data, offering a rich vein of content for research purposes [59,60]. Twitter, in particular, is integrated into the daily routines of millions, positioning it as one of the foremost social media networks currently in op-

eration [61]. The platform facilitates the aggregation of a substantial volume of text, contributed by a diverse array of Arab speakers who often express their thoughts and opinions in their respective Arabic dialects [62].

Tweets are typically short and informal, frequently composed in the users' own dialects, and reveal a plethora of spoken language features. This makes them an invaluable resource for the study of Arabic dialects, offering insights into the linguistic nuances and vernacular expressions prevalent within the Arab-speaking community [63,64].

Our goal is to develop two distinct corpora: the first, GAC-6, targets the dialects commonly spoken in the Arabian Gulf, encompassing Saudi Arabia, Bahrain, Kuwait, Oman, Qatar, and the UAE. The second, SDC-5, focuses on the five principal dialects within Saudi Arabia—Hijazi, Najdi, Southern, Northern, and Eastern. Both corpora are compiled from data collected from Twitter.

In the development of the GAC-6 and the SDC-5, we maintained stringent ethical standards and emphasized privacy protection throughout the data collection phase. All data used in this study were derived from publicly available Twitter posts, adhering to Twitter's data usage policies. We collected no private information, such as direct messages or personal identifiers, beyond what is publicly visible on user profiles. To enhance privacy protection, all usernames and profile details were anonymized and omitted from the dataset. Additionally, the data were aggregated to eliminate the possibility of tracing back to individual users. We also took precautions to ensure that no sensitive data, which could potentially disclose the identities of individuals or communities, were gathered.

Next, we will explore the details of the corpora development process. This includes discussions on compilation and preprocessing techniques, annotation methodologies, and the protocols used for corpus validation, ensuring a comprehensive understanding of the corpora's foundational integrity.

3.1. Compilation and Preprocessing of the Corpora

Twitter provides an excellent Application Programming Interface (API) and development platform, featuring a comprehensive suite of tools and meticulously crafted documentation. This framework enables researchers to tap into the vast reservoir of social media content and extract additional metadata, such as geographical information [65,66]. Consequently, Twitter was chosen as the primary data source for the construction of the envisaged corpus. The collection of data was facilitated by the Twitter REST APIs, which are accessible through Twitter user credentials via Open Authentication (OAuth) [67]. Leveraging the Twitter API streaming library "Tweepy", a Python library dedicated to tweet retrieval, we amassed thousands of tweets characterized by the use of dialectal expressions typical of Gulf Arabic and Saudi Dialect speakers [68,69]. Our methodology in this study was predicated on the extraction of dialect-specific tweets through the application of filters based on seed words pertinent to each dialect. This approach circumvents the necessity of sifting through an extensive volume of Arabic tweets and subsequently undertaking a labor-intensive annotation process. Algorithm 1 outlines our approach for the automated construction of an annotated dialectal corpus.

The initial step in our corpus construction involved the identification of seed words for each dialect. Seed words are defined as terms that are frequently and predominantly used within a particular dialect and not found in others [37,46]. We focused on unique expressions characteristic of each dialect, used exclusively by its native speakers, to compile the corpus. These dialect-specific terms were employed as query parameters, coupled with the "lang = ar" filter to confine the search to Arabic tweets, facilitating the collection of a real-time tweet stream from speakers of the targeted dialects [49]. Each tweet was then annotated with the user's name and location.

To enhance the reliability of the dialect attribution, we incorporated geographical information from the Twitter profiles associated with each tweet, ensuring that the tweets originated from within the designated dialectal regions. The corpus was further expanded by aggregating additional tweets from each user's profile. We compiled a list of user pro-

files pertinent to each of the six Gulf dialect regions and the five Saudi dialects by identifying tweets containing specific seed words and dialectal expressions exclusive to a single dialect. Only users with a minimum of 1000 tweets were considered, from which we downloaded up to 500 tweets per account. During the collection process, duplicate tweets and retweets were excluded to maintain the corpus's uniqueness.

Algorithm 1: A high-level algorithm for the automatic construction of the proposed corpus. During construction, we rely on four components of a tweet: the text, author, timestamp, and location.

Output: Collected tweets in corpus C in CSV format.

```
begin
    Initialize DL (dialects list), SL (seed words list), and LL (locations list)
    Identify list of dialects saving it into DL
    foreach d ∈ DL do
        Identify list of seed words and locations saving into SL[d] and LL[d],
          respectively
    end
    foreach d ∈ DL do
        foreach dialectal term s ∈ SL[d] do
            Retrieve tweets T containing s
            Ensure location of each t ∈ T belongs to LL[d]; otherwise, remove it
              from T
            Label tweets with dialect d
            Save tweets into corpus C
        end
    end
    Expand corpus C by collecting 500 tweets from each author
    Filter out retweets and duplicates in C
    Preprocess tweets in C
end
```

The dialect classification methodology was twofold: initially, it relied on the presence of dialect-specific terms within the tweets, and subsequently, it required a match between the user's profile location and the designated dialect region. The general architecture for our corpus construction is presented in Figure 2.

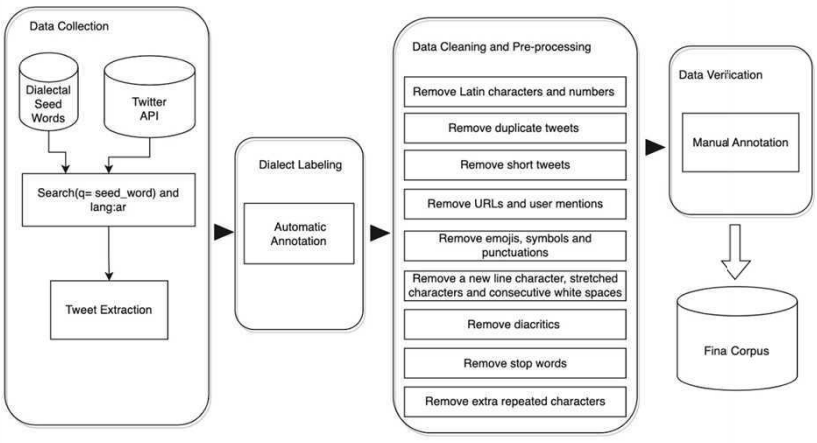


Figure 2. The general architecture of the proposed approach.

Our methodology was fundamentally anchored in the utilization of dialect-specific terminology. For instance, the term "watermelon" is articulated as جح in the Saudi dialect and as رقي in the Kuwaiti dialect. Employing this strategy enabled us to refine our tweet selection, ensuring a focus on the specific dialects of the Gulf region and Saudi Arabia. The compilation of seed words and dialectal expressions for each Saudi dialect was compiled from https://lahajat.blogspot.com/p/blog-page_7.html (accessed on 9 June 2023). Conversely, the compilation of seed words for the Gulf dialects, encompassing Kuwait, Bahrain, Qatar, the United Arab Emirates, Saudi Arabia, and Oman, was derived from Mo3jam (https://en.mo3jam.com/, a dictionary of colloquial Arabic/Arabic slang, accessed on 24 October 2023). A selection of these seed words from each Gulf dialect, accompanied by representative tweet examples, is listed in Table 1. Furthermore, Table 2 presents illustrative tweets in the Saudi dialect, demonstrating the application of dialect-specific terms in authentic social media discourse.

Given the noise and extraneous information typical of data sourced from social media, the processes of data cleaning and preprocessing are crucial [53,70]. The primary objective of preprocessing the extracted data was to cleanse it of noise and irrelevant content, thereby enhancing the dataset's quality for more precise dialect identification tasks [46].

The initial phase of preprocessing involved manually discarding tweets we identified as advertisements. Subsequent to this manual filtering, the tweets were subjected to a systematic preprocessing regimen, as shown in Algorithm 2 [71]. This algorithmic approach to preprocessing facilitated the refinement of the dataset, culminating in a polished corpus. Consequently, the Gulf Arabic Corpus was reduced to an approximate total of 1.7 million tweets, while the Saudi Dialect Corpus was cut to about 790,000 tweets, thereby ensuring a cleaner, more focused dataset for subsequent analysis.

Table 1. Sample seed words for each dialect in the Arabian Gulf are provided alongside a sample tweet from that specific dialect. Seed words are distinctive terms exclusive to a particular dialect. For convenience, an English translation (tr) is also included.

	Seed Words		**Sample Tweet**	
Region	**Arabic**	**English (tr)**	**Arabic**	**English (tr)**
Saudi Arabia	ارتعت، تكفون	I was horrified, please	احتاج عصفر ارتعت روعه كان قلبي بيوقف تعالو اقرو علي تكفون	I need a yellow safflower. I was horrified. My heart was stopping. Come and tell me that you will be satisfied.
Bahrain	سهده، صج، رزنامة، اليمعة	Quiet, true, calendar, Friday	الظاهر رزنامه متأخره عندنا اليوم الخميس وعندهم يوم اليمعه صج سهده الشوارع	Apparently, we have a late calendar. Today is Thursday and there are, on Friday, truly quiet streets.
Kuwait	ماكو، تنطرون	Nothing, waiting for	صدقوني ماكو احلى من انكم تنطرون الشروق عالبحر	Believe me, there is nothing better than waiting for the sunrise over the sea.
Oman	احسش، جزاتش، سح	I feel, you deserve, dates (fruit)	حبيبتي احسش قفلتي على قلبي بقفل وجزاتش اهديش سطل سح	My dear, I feel like you have locked my heart with a lock and you deserve to be provided with a bucket of dates.
Qatar	دلاغ، جوتي	Sock, shoes	ستايل ولدي اليوم ذكرني بالتعاون بين غوتشي وأديداس يعني ما يصير تلبس دلاغ طويل أزرق وفيه رسمة سيارات ملونة والجوتي	My son's style today reminded me of the collaboration between Gucci and Adidas. I mean, you can't wear a long blue sock that has a drawing of colorful cars, and those shoes.
Emirates	رقاد، ما اروم	Sleepy, can't	الله ياخذ الرياضه سويت رياضه وللحين منسدحه فيني رقاد ما اروم اتحرك ادرس	May God take away exercise. I did exercise, and for now, I am lying down, sleepy, and can't move or study.

Table 2. Sample seed words accompanied by a tweet example from regional Saudi dialects. We also included an English translation.

	Seed Words		Sample Tweet	
Region	**Arabic**	**English (tr)**	**Arabic**	**English (tr)**
Hijazi	التاني، هديك، اشبك	The other, that, what is going on	ابدا مافي تفاهم كل واحد يعطي التاني هديك النظرة اللي اشبك	There is absolutely no understanding. Each one gives the other that look. That is what is going on.
Najdi	تسان علمتين طملة، واخزياه	Why don't, tell me, wet, shame	رحت أمر أختي بالكلية وأنا عند الباب قالت ترى معي بنت عمي طيب تسان علمتين قبل والله ي السيارة طملة من المطر واخزياه	I went to pick up my sister at the college. While I was at the door, she said "See, I'm with my cousin". So why don't you tell me before? I swear that the car is wet from the rain and shame.
Southern	يطعني، عنش	Protect me, About you	يابعد الدنيا يسوير الله يطعني عنش ياروحي	My darling swayer, may God protect me from you, my dear.
Northen	شنوح	Why	والله زعلانه شنوح ما فزنا احسب بناخذ كاس العالم	I swear I'm upset because why didn't we win? I thought we'd take the World Cup.
Eastern	هطف	Dumb	منظر يغيض ويغبن كل هطف وهطفه	A sight that enrages and belittles every little person.

Algorithm 2: Preprocessing the tweets

Input: Raw tweets t
Output: Cleaned tweets t'

begin
- Discard short tweets (those with single words, e.g., نعم "yes")
- Filter out retweets
- Exclude duplicate tweets
- Exclude tweets that contain non-Arabic words
- Remove all stop-words from t
- Filter out each word w in tweets such as numerals, diacritical marks, emojis, symbols including newline characters, punctuation, URL links, hashtags, and user mentions (i.e.,@Username)
- Filter out stretched characters (kashida) and consecutive white spaces
- Normalize speech effects in t (e.g., "Helloo", "Helloooo", etc., are mapped to "Hello")

end

3.2. Annotating the Corpus

In this section, we describe the methodology adopted for the annotation of the corpus, detailing the procedural steps and tools employed for annotating each tweet within the corpus, followed by the strategies implemented to assess the quality of the annotation task.

The genesis of a corpus extends beyond mere data accumulation; it encompasses a rigorous process of data verification and validation to ensure the corpus's reliability and applicability [72–74]. The primary goal of this endeavor is to forge a dialectal corpus of tweets, distinguished by high-quality annotations, to serve as a resource for scholars engaged in the study of Arabic dialects within the domain of Arabic NLP. This includes, but is not limited to, investigations pertaining to Arabic dialect identification.

The success of dialect identification is intrinsically linked to the precision of annotation outcomes [75]. In order to evaluate the quality of our corpus, constructed and anno-

tated via the proposed algorithm, we conducted a manual annotation exercise. From each region represented in the Arabian Gulf corpus, we randomly selected 2000 tweets, resulting in a total of 12,000 tweets. Similarly, for the Saudi Corpus, a total of 10,000 tweets were chosen for manual annotation. For this task, we utilized Label Studio, (https://labelstud.io/, accessed on 28 September 2023), an open-source data labeling platform renowned for its versatility in annotating, labeling, and preparing diverse datasets.

Within the Label Studio environment, we established a project dedicated to the annotation of tweets. Annotators were presented with tweets and instructed to assign a dialect label to each one. For the GAC-6, the tweets were categorized into one of six dialect labels corresponding to the Gulf regions: Saudi Arabia, Bahrain, Kuwait, Oman, Qatar, and UAE. In the case of the SDC-5, annotators classified each tweet under one of the five designated Saudi labels: Hejazi, Najdi, Southern, Northern, and Eastern. This meticulous process of manual annotation serves as a cornerstone for ensuring the integrity and utility of the corpus for research in Arabic dialect identification and other NLP applications.

Almuzaini and Azmi [76] employed crowdsourcing to annotate a large volume of data in MSA. However, they encountered quality issues and lapses due to incompetent or dishonest annotators, particularly when the tasks were poorly defined or required specialized knowledge. Considering the anonymity of crowdsourcers, we chose to interact directly with the annotators to minimize these risks.

For the annotation process, we employed two primary annotators, supported by a third annotator responsible for resolving any discrepancies or ambiguities that may arise during the initial annotation phase. To ensure high-quality and consistent output, all annotators underwent comprehensive training. This training encompassed detailed annotation guidelines, which included definitions, examples, and specific linguistic features characteristic of each dialect. Additionally, the annotators participated in sessions to familiarize themselves with the Label Studio annotation platform, and they practiced annotating sample data, receiving feedback to fine-tune their understanding and application of the guidelines.

To guarantee the accuracy of the annotations, the third annotator reviewed instances where the initial annotations differed. The criteria for annotation focused on identifying distinct lexical items, phonological variations, and syntactic constructions unique to each dialect. Annotators were trained to recognize explicit markers—specific words or phrases characteristic of a dialect—as well as contextual clues like cultural references or idiomatic expressions, ensuring a thorough and nuanced analysis. Figure 3 illustrates sample instances of the annotation task.

Throughout this process, annotators were instructed to identify the dialect of the text presented to them and mark their selection using the provided checkboxes corresponding to each dialect. It was imperative that the annotators possess native-level proficiency in Arabic to ensure the accuracy of the dialect labeling.

All annotators involved were native Arabic speakers, each with a collegiate level of education. Specifically, for the annotation of the Saudi Dialect Corpus (SDC-5), annotators were recruited from within Saudi Arabia to ensure an intrinsic understanding of the regional dialects. Similarly, the annotation of the Gulf Arabic Corpus (GAC-6) was entrusted to native speakers from the Gulf countries, ensuring an authentic representation of the dialectal nuances.



Figure 3. The graphical Arabic user interface for manually annotating the Gulf Arabic Corpus (GAC-6). The annotator receives a sample tweet (displayed in the top line) from the GAC-6 and is instructed to classify the text into one of the designated Gulf Arabic dialects, with the six options displayed at the botton.

4. Results and Discussion

4.1. Overview of Our Compiled Corpora

In this study, we have constructed a large-scale corpus consisting of Gulf dialect Arabic text, derived from data sourced from Twitter. For the purposes of this research, the following two corpora have been developed:

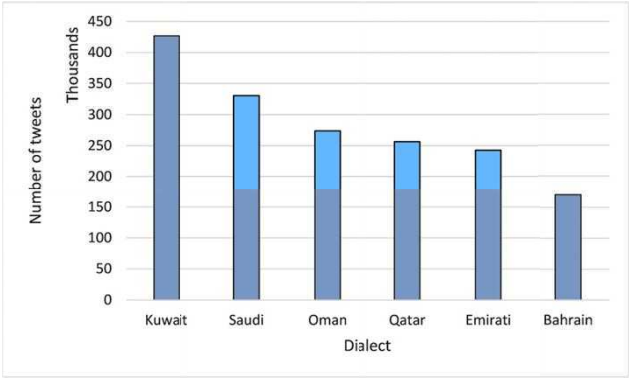
- The Gulf Arabic Corpus (GAC-6), which encompasses the dialects prevalent in the Gulf region, specifically Saudi Arabia, Bahrain, Kuwait, Oman, Qatar, and the Emirates (UAE). This corpus contains approximately 1.7 million Arabic tweets, offering a broad representation of the linguistic diversity within the Gulf countries.
- The Saudi Dialect Corpus (SDC-5), comprising around 790,000 tweets in Saudi Arabic, representing the five main dialects found within Saudi Arabia: Hijazi, Najdi, Southern, Northern, and Eastern. This corpus provides a focused insight into the linguistic variations across different regions of Saudi Arabia.

In the construction of the GAC-6, our initial collection comprised 2.6 million tweets. Following a meticulous cleaning process, which included the removal of redundant tweets, the corpus was reduced to 1.7 million tweets. The final composition of the corpus featured a diverse distribution of tweets across various Gulf dialects: 330,408 tweets were categorized under the Saudi Arabian dialect, 169,977 tweets under the Bahraini dialect, 426,771 tweets under the Kuwaiti dialect, 273,920 tweets under the Omani dialect, 256,377 tweets under the Qatari dialect, and 242,590 tweets under the Emirati dialect. The distribution of tweets by dialect within the Gulf Arabic Corpus is shown in Figure 4a. Notably, the Saudi and Kuwaiti dialects are represented by a larger volume of tweets compared to the Omani, Qatari, and Bahraini dialects. This discrepancy in tweet volumes may be attributed to the higher popularity of Twitter in Saudi Arabia and Kuwait.

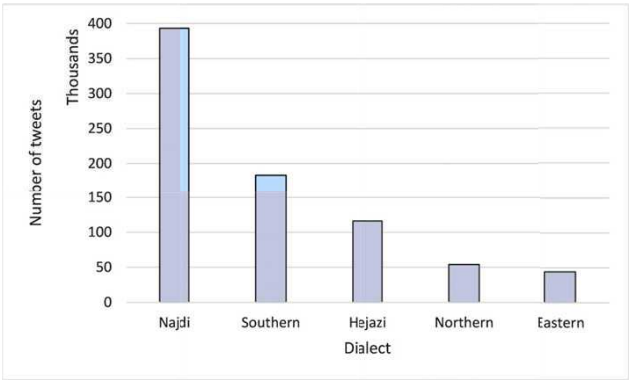
For the development of the SDC-5, our objective was to encompass all five dialects prevalent within Saudi Arabia. Prior corpora focusing on Saudi dialects [42] have often omitted the Southern and Northern dialects, attributing this exclusion to the relative scarcity and limited usage of these dialects in comparison to the Najdi, Hijazi, and Eastern dialects. Contrary to these precedents, our research endeavored to construct a corpus that inclusively represents the five Saudi dialects: Hijazi, Najdi, Southern, Northern, and Eastern. As shown in Figure 4b, the Saudi Dialect Corpus comprises a total of 790,000 tweets, distributed as follows: 116,117 tweets in the Hijazi dialect, 393,342 tweets in the Najdi dialect, 183,487 tweets in the Southern dialect, 53,883 tweets in the Northern dialect, and

44,029 tweets in the Eastern dialect. This distribution underscores our commitment to providing a comprehensive representation of the linguistic diversity within Saudi Arabia.

The comprehensive statistical overview of both the GAC-6 and the SDC-5 is shown in Table 3. This summary delineates the number of tweets and the extracted word count for each dialect within both corpora. Additionally, it presents an analysis of the tweet lengths across the dialects, specifying the minimum, maximum, and average number of words per tweet.



(**a**)



(**b**)

Figure 4. The number of tweets, expressed in thousands, amassed for each dialect across the two corpora. (**a**) Gulf Arabic Corpus (GAC-6). (**b**) Saudi Dialect Corpus (SDC-5).

Table 3. The general statistics of our two corpora, the Gulf Arabic Corpus (GAC-6) and the Saudi Dialect Corpus (SDC-5). We report the size of the tweet in words.

				Tweet Size		
Corpus	**Dialects**	**# Tweets**	**# Words**	**Min**	**Max**	**Avg**
GAC-6	Kuwaiti	426,771	4,738,460	3	48	11.13
	Saudi	330,408	3,638,971	3	44	11.35
	Omani	273,920	3,060,739	4	45	11.73
	Qatari	256,377	3,676,452	4	46	14.34
	Emirati	242,590	2,562,961	5	40	10.56
	Bahraini	169,977	2,051,254	4	35	12.67
SDC-5	Najdi	393,342	3,960,246	3	44	10.68
	Southern	183,487	2,994,528	4	38	16.32
	Hejazi	116,117	1,123,901	3	47	9.67
	Northern	53,883	550,647	4	39	10.29
	Eastern	44,029	441,850	4	35	10.35

Within the GAC-6, the average tweet length stands at 11.96 words. The corpus features a tweet in the Kuwaiti dialect as the longest, containing 48 words, while the shortest tweets, found within both the Kuwaiti and Saudi dialects, comprise merely three words. Conversely, in the SDC-5, the Hejazi dialect boasts the longest tweet, encompassing 47 words. The shortest tweets, each consisting of just three words, are observed in both the Najdi and Hejazi dialects, highlighting the variance in expression and conciseness across the different dialects represented in the corpora.

While the development of GAC-6 and SDC-5 contributes valuable resources to Arabic NLP, it is essential to acknowledge several limitations of this study. Data collection was limited to publicly available Twitter posts, which might not capture the complete spectrum of dialectal variations due to user demographics and the informal nature of Twitter content, such as the brevity of tweets. Additionally, the representation of specific dialects, particularly the Southern and Northern Saudi dialects, is somewhat underrepresented in the corpora, potentially limiting the generalizability of our findings to all dialects. Lastly, despite the use of both manual and automated annotation techniques aimed at ensuring high-quality data, variability in inter-annotator agreement for some dialects persists, suggesting a need for further refinement of the annotation guidelines and processes.

4.2. Evaluating the Quality of Corpus Annotations

Ensuring the quality of annotations is crucial for the reliability of the corpus and its subsequent utility in developing precise dialect identification models. To assess the annotation quality within our study, we employed inter-annotator agreement (IAA) measures on the dialect annotations of the tweets. IAA measures provide insights into the consistency of annotator choices regarding dialect annotations, underpinning the validity of the annotated data.

The premise is that if annotators exhibit discrepancies in their annotations, it could be indicative of potential challenges for a dialect identification model to accurately classify those instances [77]. In our research, we quantified inter-annotator agreement using Cohen's Kappa coefficient (κ), a widely recognized statistical measure designed to evaluate the level of agreement between two annotators beyond chance in classification tasks. Cohen's Kappa is calculated using the following equation, which accounts for the observed agreement and the expected agreement by chance, thereby offering a normalized measure of annotator concordance:

$$\kappa = \frac{P_o - P_e}{1 - P_e}, \tag{1}$$

where P_o is the observed agreement among annotators, and P_e can be defined as the expected agreement obtained by the random assignment of labels by annotators during the annotation process. The P_e is given by

$$P_e = \frac{1}{N^2} \sum_{T} n_{T_1} \cdot n_{T_2}, \tag{2}$$

where n_{T_1} and n_{T_2} is the number of tokens labeled with tag T by annotator 1 and annotator 2, respectively, and N is the total number of annotated tokens.

The analysis of annotation quality through Cohen's Kappa revealed a commendable degree of agreement among the annotators for both corpora under study. For the GAC-6, the average Cohen's Kappa was approximately 78%, indicating a robust level of concurrence. The SDC-5 exhibited an even higher level of annotator consensus, with an average Kappa value of 90%.

The detailed breakdown of inter-annotator agreement for each region within both corpora, as presented in Table 4, showcases varying degrees of agreement across the dialects. For the GAC-6, the Bahraini, Omani, and Qatari dialect annotations fell within the "satisfactory" range, with Kappa values between 0.6 and 0.8. In contrast, the Saudi, Kuwaiti, and

Emirati dialects demonstrated "really good" agreement levels, with Kappa values ranging from 0.8 to 1, underscoring a high reliability in these annotations.

In the context of the SDC-5, the Hijazi dialect annotations achieved a notably high Kappa value of 92%, reflecting near-perfect annotator alignment. The Najdi and Southern dialects also showcased excellent agreement, with Kappa values around 91%, while the Northern and Eastern dialects exhibited very good agreement, with Kappa values at 87% and 89%, respectively. These results underscore the high quality of the annotation process, bolstering the reliability of the corpora for dialect identification research and applications.

Table 4. Cohen's Kappa (κ) for the individual dialects within each corpora, GAC-6 and SDC-5, along with the overall average.

Corpus	Dialects	κ
GAC-6	Kuwaiti	0.89
	Saudi	0.91
	Omani	0.65
	Qatari	0.67
	Emirati	0.88
	Bahraini	0.69
	Average	0.78
SDC-5	Najdi	0.91
	Southern	0.91
	Hejazi	0.92
	Northern	0.87
	Eastern	0.89
	Average	0.90

5. Conclusions

In this study, we have outlined the methodologies utilized in compiling and developing two substantial linguistic resources for the Arabic language: the Gulf Arabic Corpus (GAC-6) and the Saudi Dialect Corpus (SDC-5). These datasets, richly annotated and multi-dialectal, encapsulate a broad spectrum of linguistic nuances from the Gulf and Saudi regions. GAC-6 comprises approximately 1.7 million labeled tweets from six Gulf dialects, while SDC-5 features around 790,000 tweets that reflect the predominant dialects within Saudi Arabia: Hijazi, Najdi, Southern, Northern, and Eastern.

The annotation strategy combined manual and automated methods, leveraging user location data and dialect-specific seed words to achieve precise dialect identification. A portion of the corpus was subjected to thorough manual annotation, with inter-annotator agreement metrics confirming the quality of the annotations and the overall reliability of the datasets. These resources are set to greatly advance Arabic Natural Language Processing, supporting sophisticated inquiries into dialect identification, sentiment analysis, author profiling, machine translation, and morphological analysis.

Looking ahead, we plan to expand these corpora by incorporating diverse textual data from additional online platforms, thus broadening the scope of dialectal representation. We aim to further refine our annotation methods and enhance the robustness of our techniques. Moreover, by making these comprehensive resources accessible to the global research community, we contribute to the fields of Arabic and computational linguistics. This initiative aligns with the movement toward employing sophisticated data-driven AI technologies to dissect and understand the intricate linguistic framework of the Arabic language, thereby facilitating the development of next-generation AI applications.

Author Contributions: Conceptualization, N.A.-S. and A.M.A.; methodology, N.A.-S.; investigation, N.A.-S.; data curation, N.A.-S.; writing—original draft, N.A.-S.; writing—review and editing, A.M.A.; supervision, A.M.A. and M.H.; funding acquisition, A.M.A. All authors have read and agreed to the published version of this manuscript.

Funding: The authors acknowledge funding from the Research, Development, and Innovation Authority (RDIA), Saudi Arabia, Saudi Basic Science Initiative—Basic Science Grants (BSG), number 10529.

Data Availability Statement: The data will be provided upon request. Please contact the corresponding author for access.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations (ordered alphabetically) are used in this manuscript:

AI	Artificial Intelligence
DA	Dialectal Arabic
GAC-6	Gulf Arabic Corpus
IAA	inter-annotator agreement
MENA	Middle East and North Africa
MSA	Modern Standard Arabic
NLP	Natural Language Processing
SDC-5	Saudi Dialect Corpus

References

1. Azmi, A.; Alsaiari, A. Arabic typography: A survey. *Int. J. Electr. Comput. Sci.* **2010**, *9*, 16–22.
2. Bouamor, H.; Habash, N.; Salameh, M.; Zaghouani, W.; Rambow, O.; Abdulrahim, D.; Obeid, O.; Khalifa, S.; Eryani, F.; Erdmann, A.; et al. The MADAR Arabic Dialect Corpus and Lexicon. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018.
3. Abdul-Mageed, M.; Alhuzali, H.; Elaraby, M. You Tweet What You Speak: A City-Level Dataset of Arabic Dialects. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018.
4. Azmi, A.M.; Aljafari, E.A. Universal web accessibility and the challenge to integrate informal Arabic users: A case study. *Univers. Access Inf. Soc.* **2018**, *17*, 131–145. [CrossRef]
5. Jarrar, M.; Habash, N.; Alrimawi, F.; Akra, D.; Zalmout, N. Curras: An annotated corpus for the Palestinian Arabic dialect. *Lang. Resour. Eval.* **2017**, *51*, 745–775. [CrossRef]
6. Azmi, A.M.; Aljafari, E.A. Modern information retrieval in Arabic–catering to standard and colloquial Arabic users. *J. Inf. Sci.* **2015**, *41*, 506–517. [CrossRef]
7. Haff, K.E.; Jarrar, M.; Hammouda, T.; Zaraket, F. Curras + Baladi: Towards a levantine corpus. *arXiv* **2022**, arXiv:2205.09692.
8. Jarrar, M.; Zaraket, F.A.; Hammouda, T.; Alavi, D.M.; Wahlisch, M. LISAN: Yemeni, Iraqi, Libyan, and Sudanese Arabic Dialect Copora with Morphological Annotations. In Proceedings of the 2023 20th ACS/IEEE International Conference on Computer Systems and Applications, Giza, Egypt, 4–7 December 2022.
9. Shoufan, A.; Alameri, S. Natural language processing for dialectical Arabic: A survey. In Proceedings of the Second Workshop on Arabic Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 36–48.
10. Azmi, A.M.; Al-Ghadir, A.I. Using Twitter as a digital insight into public stance on societal behavioral dynamics. *J. King Saud Univ. Comput. Inf. Sci.* **2024**, *36*, 102078. [CrossRef]
11. AlShenaifi, N.; Azmi, A. Arabic dialect identification using machine learning and transformer-based models. In Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP 2022), Abu Dhabi, United Arab Emirates, 8 December 2022.
12. Alshargi, F.; Dibas, S.; Alkhereyf, S.; Faraj, R.; Abdulkareem, B.; Yagi, S.; Kacha, O.; Habash, N.; Rambow, O. Morphologically annotated corpora for seven Arabic dialects: Taizi, Sanaani, Najdi, Jordanian, Syrian, Iraqi and Moroccan. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 1–2 August 2019; pp. 137–147.
13. Lichouri, M.; Abbas, M.; Freihat, A.A.; Megtouf, D.E.H. Word-level vs. sentence-level language identification: Application to Algerian and Arabic dialects. *Procedia Comput. Sci.* **2018**, *142*, 246–253. [CrossRef]
14. Azmi, A.; Al-Thanyyan, S. Ikhtasir—A user selected compression ratio Arabic text summarization system. In Proceedings of the 2009 International Conference on Natural Language Processing and Knowledge Engineering, Dalian, China, 24–27 September 2009; pp. 1–7.
15. Al-Jouie, M.F.; Azmi, A.M. Automated Evaluation of School Children Essays in Arabic. In Proceedings of the 3rd International Conference on Arabic Computational Linguistics (ACLing 2017), Dubai, United Arab Emirates, 5–6 November 2017; Volume 117, pp. 19–22.
16. Mohammed, E.A.; Aziz, M.J.A. English to Arabic machine translation based on reordring algorithm. *J. Comput. Sci.* **2011**, *7*, 120–128. [CrossRef]

17. Alnefaie, R.; Azmi, A.M. Automatic minimal diacritization of Arabic texts. In Proceedings of the 3rd International Conference on Arabic Computational Linguistics (ACLing 2017), Dubai, United Arab Emirates, 5–6 November 2017; Volume 117, pp. 169–174.
18. Tarmom, T.; Teahan, W.; Atwell, E.; Alsalka, M. Compression vs Traditional Machine Learning Classifiers to Detect Code-switching in Varieties and Dialects: Arabic as a Case Study. *J. Nat. Lang. Eng.* **2020**, *26*, 663–676. [CrossRef]
19. Alhussain, A.; Azmi, A.M. Beyond Event-Centric Narratives: Advancing Arabic Story Generation with Large Language Models and Beam Search. *Mathematics* **2024**, *12*, 1548. [CrossRef]
20. Al-Thanyyan, S.S.; Azmi, A.M. Simplification of Arabic text: A hybrid approach integrating machine translation and transformer-based lexical model. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 101662. [CrossRef]
21. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60.
22. Alwaneen, T.H.; Azmi, A.M. Stacked dynamic memory-coattention network for answering why-questions in Arabic. *Neural Comput. Appl.* **2024**, *36*, 8867–8883. [CrossRef]
23. Mannaa, Z.M.; Azmi, A.M.; Aboalsamh, H.A. Computer-assisted i'raab of Arabic sentences for teaching grammar to students. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *10*, 8909–8926. [CrossRef]
24. Abu Kwaik, K.; Saad, M.; Chatzikyriakidis, S.; Dobnik, S. Shami: A Corpus of Levantine Arabic Dialects. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, European Language Resources Association (ELRA), Miyazaki, Japan, 7–12 May 2018.
25. Abdelali, A.; Mubarak, H.; Samih, Y.; Hassan, S.; Darwish, K. QADI: Arabic dialect identification in the wild. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 1–10.
26. Alharbi, A.; Lee, M. Kawarith: An Arabic Twitter corpus for crisis events. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 42–52.
27. Darwish, K.; Sajjad, H.; Mubarak, H. Verifiably effective Arabic dialect identification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1465–1468.
28. Al-Kabi, M.; Al-Ayyoub, M.; Alsmadi, I.; Wahsheh, H. A prototype for a standard Arabic sentiment analysis corpus. *Int. Arab J. Inf. Technol.* **2016**, *13*, 163–170.
29. Ahmed, A.; Ali, N.; Alzubaidi, M.; Zaghouani, W.; Abd-alrazaq, A.A.; Househ, M. Freely available Arabic corpora: A scoping review. *Comput. Methods Programs Biomed. Update* **2022**, *2*, 100049. [CrossRef]
30. Mubarak, H. Dial2MSA: A tweets corpus for converting dialectal Arabic to modern standard Arabic. In Proceedings of the 3rd Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT3), Miyazaki, Japan, 8 May 2018; pp. 49–53.
31. Zaghouani, W. Critical Survey of the Freely Available Arabic Corpora. In Proceedings of the Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop Programme, Reykjavik, Iceland, 27 May 2014; pp. 1–8.
32. Al-Sabbagh, R.; Girju, R. YADAC: Yet another Dialectal Arabic Corpus. In Proceedings of the Eighth International Conference on Language Resources and Evaluation, Istanbul, Turkey, 21–27 May 2012; pp. 2882–2889.
33. Khalifa, S.; Habash, N.; Abdulrahim, D.; Hassan, S. A Large Scale Corpus of Gulf Arabic. In Proceedings of the Tenth International Conference on Language Resources and Evaluation, Portorož, Slovenia, 23–28 May 2016; pp. 4282–4289.
34. Khalifa, S.; Habash, N.; Eryani, F.; Obeid, O.; Abdulrahim, D.; Al Kaabi, M. A morphologically annotated corpus of Emirati Arabic. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018.
35. Abdulrahim, D.; Inoue, G.; Shamsan, L.; Khalifa, S.; Habash, N. The Bahrain Corpus: A Multi-genre Corpus of Bahraini Arabic. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 20–25 June 2022; pp. 2345–2352.
36. Al-Mulla, S.; Zaghouani, W. Building a corpus of Qatari Arabic expressions. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, Marseille, France, 11–16 May 2020; pp. 24–31.
37. Alshutayri, A.O.O.; Atwell, E. Exploring Twitter as a source of an Arabic dialect corpus. *Int. J. Comput. Linguist. (IJCL)* **2017**, *8*, 37–44.
38. Elaraby, M.; Abdul-Mageed, M. Deep models for Arabic dialect identification on benchmarked data. In Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018), Santa Fe, NM, USA, 20 August 2018; pp. 263–274.
39. Harrat, S.; Meftouh, K.; Smaili, K. Creating parallel Arabic dialect corpus: Pitfalls to avoid. In Proceedings of the 18th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING), Budapest, Hungary, 17–23 April 2017.
40. Alshutayri, A.; Atwell, E. Classifying Arabic dialect text in the social media Arabic dialect corpus (SMADC). In Proceedings of the 3rd Workshop on Arabic Corpus Linguistics, Cardiff, UK, 22 July 2019; pp. 51–59.
41. Alshutayri, A.; Atwell, E. Arabic dialects annotation using an online game. In Proceedings of the IEEE 2nd International Conference on Natural Language and Speech Processing (ICNLSP), Algiers, Algeria, 25–26 April 2018; pp. 1–5.
42. Bayazed, A.; Torabah, O.; AlSulami, R.; Alahmadi, D.; Babour, A.; Saeedi, K. SDCT: Multi-dialects corpus classification for Saudi Tweets. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 216–223. [CrossRef]
43. Darwish, K.; Magdy, W. Arabic information retrieval. *Found. Trends Inf. Retr.* **2014**, *7*, 239–342. [CrossRef]

44. Al-Razgan, M.; Alrowily, A.; Al-Matham, R.N.; Alghamdi, K.M.; Shaabi, M.; Alssum, L. Using diffusion of innovation theory and sentiment analysis to analyze attitudes toward driving adoption by Saudi women. *Technol. Soc.* **2021**, *65*, 101558. [CrossRef]
45. Zaidan, O.; Callison-Burch, C. The Arabic online commentary dataset: An annotated dataset of informal Arabic with high dialectal content. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19 June 2011; pp. 37–41.
46. Alsarsour, I.; Mohamed, E.; Suwaileh, R.; Elsayed, T. Dart: A large dataset of dialectal Arabic tweets. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018.
47. Zaghouani, W.; Charfi, A. Arap-tweet: A large multi-dialect twitter corpus for gender, age and language variety identification. *arXiv* **2018**, arXiv:1808.07674.
48. Sadat, F.; Kazemi, F.; Farzindar, A. Automatic identification of Arabic dialects in social media. In Proceedings of the First International Workshop on Social Media Retrieval and Analysis (SoMeRA'14), Gold Coast, QLD, Australia, 11 July 2014; Association for Computing Machinery: New York, NY, USA; pp. 35–40.
49. Alshutayri, A.; Atwell, E. Creating an Arabic dialect text corpus by exploring Twitter, Facebook, and online newspapers. In Proceedings of the OSACT3: The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools, LREC, Miyazaki, Japan, 8 May 2018; pp. 54–61.
50. Bouamor, H.; Habash, N.; Oflazer, K. A Multidialectal Parallel Corpus of Arabic. In Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, 26–31 May 2014; pp. 1240–1245.
51. Takezawa, T.; Kikui, G.; Mizushima, M.; Sumita, E. Multilingual spoken language corpus development for communication research. *Int. J. Comput. Linguist. Chin. Lang. Process.* **2007**, *12*, 303–324.
52. Mubarak, H.; Darwish, K. Using Twitter to collect a multi-dialectal corpus of Arabic. In Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP), Doha, Qatar, 25 October 2014; pp. 1–7.
53. Al-Twairesh, N.; Al-Matham, R.; Madi, N.; Almugren, N.; Al-Aljmi, A.H.; Alshalan, S.; Alshalan, R.; Alrumayyan, N.; Al-Manea, S.; Bawazeer, S.; et al. SUAR: Towards building a corpus for the Saudi dialect. *Procedia Comput. Sci.* **2018**, *142*, 72–82. [CrossRef]
54. Pasha, A.; Al-Badrashiny, M.; Diab, M.; El Kholy, A.; Eskander, R.; Habash, N.; Pooleery, M.; Rambow, O.; Roth, R. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In Proceedings of the 9th International Conference on Language Resources and Evaluation, Reykjavik, Iceland, 26–31 May 2014; pp. 1094–1101.
55. Alowisheq, A.; Al-Twairesh, N.; Altuwaijri, M.; Almoammar, A.; Alsuwailem, A.; Albuhairi, T.; Alahaideb, W.; Alhumoud, S. MARSA: Multi-domain Arabic resources for sentiment analysis. *IEEE Access* **2021**, *9*, 142718–142728. [CrossRef]
56. Elgibreen, H.; Faisal, M.; Al Sulaiman, M.; Abdou, S.; Mekhtiche, M.A.; Moussa, A.M.; Alohali, Y.A.; Abdul, W.; Muhammad, G.; Rashwan, M.; et al. An incremental approach to corpus design and construction: Application to a large contemporary Saudi corpus. *IEEE Access* **2021**, *9*, 88405–88428. [CrossRef]
57. Alruily, M. Issues of dialectal Saudi twitter corpus. *Int. Arab J. Inf. Technol.* **2020**, *17*, 367–374. [CrossRef]
58. Al-Ghadir, A.; Azmi, A. A Study of Arabic Social Media Users—Posting Behavior and Author's Gender Prediction. *Cogn. Comput.* **2019**, *11*, 71–86. [CrossRef]
59. Cotterell, R.; Callison-Burch, C. A Multi-Dialect, Multi-Genre Corpus of Informal Written Arabic. In Proceedings of the 9th International Conference on Language Resources and Evaluation, Reykjavik, Iceland, 26–31 May 2014; pp. 241–245.
60. Ibrahim, H.S.; Abdou, S.M.; Gheith, M. MIKA: A tagged corpus for modern standard Arabic and colloquial sentiment analysis. In Proceedings of the IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), Kolkata, India, 9–11 July 2015; pp. 353–358.
61. Hegazi, M.O.; Al-Dossari, Y.; Al-Yahy, A.; Al-Sumari, A.; Hilal, A. Preprocessing Arabic text on social media. *Heliyon* **2021**, *7*, e06191. [CrossRef]
62. Charfi, A.; Zaghouani, W.; Mehdi, S.H.; Mohamed, E. A fine-grained annotated multi-dialectal Arabic corpus. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), Varna, Bulgaria, 3–12 September 2019; pp. 198–204.
63. Althobaiti, M.J. Creation of annotated country-level dialectal Arabic resources: An unsupervised approach. *Nat. Lang. Eng.* **2022**, *28*, 607–648. [CrossRef]
64. Alshutayri, A.; Atwell, E. A social media corpus of Arabic dialect text. In *Building Computer-Mediated Communication Corpora for Sociolinguistic Analysis*; Stemle, E., Wigham, C.R., Eds.; Presses Universitaires Blaise Pascal: Clermont-Ferrand, France, 13 June 2019; pp. 1–23.
65. Refaee, E.; Rieser, V. An Arabic twitter corpus for subjectivity and sentiment analysis. In Proceedings of the 9th International Language Resources and Evaluation Conference, Reykjavik, Iceland, 26–31 May 2014; pp. 2268–2273.
66. Gugliotta, E.; Dinarelli, M. TArC: Tunisian Arabish Corpus First complete release. *arXiv* **2022**, arXiv:2207.04796.
67. Alabbas, W.; al Khateeb, H.M.; Mansour, A.; Epiphaniou, G.; Frommholz, I. Classification of colloquial Arabic tweets in real-time to detect high-risk floods. In Proceedings of the IEEE International Conference On Social Media, Wearable And Web Analytics (Social Media), London, UK, 19–20 June 2017; pp. 1–8.
68. Baly, R.; Khaddaj, A.; Hajj, H.; El-Hajj, W.; Shaban, K.B. Arsentd-lev: A multi-topic corpus for target-based sentiment analysis in Arabic levantine tweets. *arXiv* **2019**, arXiv:1906.01830.
69. Zaghouani, W.; Charfi, A. Guidelines and annotation framework for Arabic author profiling. *arXiv* **2018**, arXiv:1808.07678.

70. Al-Laith, A.; Shahbaz, M.; Alaskar, H.F.; Rehmat, A. AraSenCorpus: A semi-supervised approach for sentiment annotation of a large Arabic text corpus. *Appl. Sci.* **2021**, *11*, 2434. [CrossRef]
71. Zahir, J. IADD: An Integrated Arabic Dialect Identification Dataset. *Data Brief* **2022**, *40*, 107777. [CrossRef]
72. Kwaik, K.A.; Chatzikyriakidis, S.; Dobnik, S.; Saad, M.; Johansson, R. An Arabic Tweets Sentiment Analysis Dataset (ATSAD) using Distant Supervision and Self Training. In Proceedings of the 4th Workshop on Open-source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, Marseille, France, 12 May 2020; pp. 1–8.
73. Mahany, A.; Khaled, H.; Nouh, E.; Aljohani, N.; Ghoniemy, S. Annotated Corpus with Negation and Speculation in Arabic Review Domain: NSAR. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 38–46. [CrossRef]
74. Guellil, I.; Azouaou, F.; Chiclana, F. ArAutoSenti: Automatic Annotation and New Tendencies for Sentiment Classification of Arabic Messages. *Soc. Netw. Anal. Min.* **2020**, *10*, 75. [CrossRef]
75. Almuqren, L.; Alzammam, A.; Alotaibi, S.; Cristea, A.; Alhumoud, S. A review on corpus annotation for Arabic sentiment analysis. In *Social Computing and Social Media: Applications and Analytics (SCSM 2017), Part II, LNCS 10283*; Springer: Cham, Switzerland, 2017; pp. 215–225.
76. Almuzaini, H.A.; Azmi, A.M. An unsupervised annotation of Arabic texts using multi-label topic modeling and genetic algorithm. *Expert Syst. Appl.* **2022**, *203*, 117384. [CrossRef]
77. Al-Twairesh, N.; Al-Khalifa, H.; Al-Salman, A.; Al-Ohali, Y. Arasenti-tweet: A corpus for Arabic sentiment analysis of Saudi tweets. *Procedia Comput. Sci.* **2017**, *117*, 63–72. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Event Stream Denoising Method Based on Spatio-Temporal Density and Time Sequence Analysis

Haiyan Jiang, Wei Tang, Qinghui Song, Qingjun Song * and Wenchao Hao

College of Intelligent Equipment, Shandong University of Science and Technology, Tai'an 271000, China
* Correspondence: skdsqj@sdust.edu.cn; Tel.: +86-13793848379

Abstract: An event camera is a neuromimetic sensor inspired by the human retinal imaging principle, which has the advantages of high dynamic range, high temporal resolution, and low power consumption. Due to the interference of hardware and software and other factors, the event stream output from the event camera usually contains a large amount of noise, and traditional denoising algorithms cannot be applied to the event stream. To better deal with different kinds of noise and enhance the robustness of the denoising algorithm, based on the spatio-temporal distribution characteristics of effective events and noise, an event stream noise reduction and visualization algorithm is proposed. The event stream enters fine filtering after filtering the BA noise based on spatio-temporal density. The fine filtering performs time sequence analysis on the event pixels and the neighboring pixels to filter out hot noise. The proposed visualization algorithm adaptively overlaps the events of the previous frame according to the event density difference to obtain clear and coherent event frames. We conducted denoising and visualization experiments on real scenes and public datasets, respectively, and the experiments show that our algorithm is effective in filtering noise and obtaining clear and coherent event frames under different event stream densities and noise backgrounds.

Keywords: event camera; event stream visualization; denoising

1. Introduction

An event camera is an emerging bio-visual sensor that records data based on light intensity changes at each pixel point [1,2]. When the light intensity change exceeds a preset threshold, the event camera asynchronously outputs the pixel address and timestamp to form the event stream data. In scenarios such as strenuous motion, changing lighting conditions, or platform power constraints, the motion blur and exposure problems faced by traditional frame cameras can significantly reduce the detection of valid feature points, thus decreasing the reliability of the visual information, while event cameras can effectively compensate for these shortcomings and provide more stable and reliable visual data. The unique nature of event cameras allows them to show significant advantages in several fields, such as image reconstruction [3–7], 3D reconstruction [8–12], optical flow [13–16], and target tracking [17–20]. In addition, event cameras show potential applications in visual navigation and localization [21–23].

There are many challenges faced by event cameras in practical applications. For example, changes in the camera motion state and in the scene mean the event stream is not uniformly distributed over different periods, making it difficult to obtain clear and informative coherent event frames with traditional event stream visualization methods. Due to the high sensitivity of event cameras to changes in ambient brightness and junction leakage currents, there is an output even if there is no change in light intensity, called background activity (BA) [24], which affects the image quality, wastes communication bandwidth, and consumes unnecessary computational power. In high-speed operating environments and long exposures, event cameras experience hot pixels similar to conventional image sensors [25], where some pixels produce abnormal signals or offsets, which

can cause these pixels to fail to reset correctly and continue to output events, called hot noise. Asynchronous events in an event camera capturing a rapidly changing scene are subject to a large amount of noise, which degrades the quality of the event data, blurs the event information in the event stream, and may even mask the true event signals. Traditional graph and video denoising methods are difficult to directly apply to event stream denoising, so there is an urgent need for efficient and feasible event stream denoising methods. Figure 1 shows the spatio-temporal visualization of the hot pixel noise and the background activity noise in a stream of events, with the X and Y axes representing the horizontal and vertical coordinates of the events, and the Taxis coordinates corresponding to the timestamps of the events. From the figure, it can be seen that the hot noise is parallel to the time axis and occurs continuously at a single pixel; the BA noise distribution is more random and sparser from the visualization.

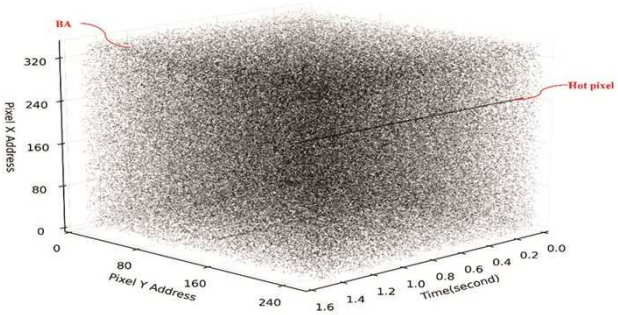


Figure 1. Random distribution for BA noise and a straight line over time for hot noise.

Although event cameras work differently from traditional frame cameras, event stream data are still affected by various noises, such as sensor noise, ambient light changes, motion blur, etc. Therefore, event stream denoising is still a challenging problem.

In this paper, we propose an event stream denoising method based on spatio-temporal correlation differences. The design philosophy of the proposed method is as follows. The regular motion of objects generates a large number of events in the event camera, which are regarded as real events. However, due to the impact of the event camera hardware and other factors, a significant amount of noise is also present in the event stream. This noise can be categorized into BA noise and hot noise based on their spatio-temporal distribution characteristics. We propose a cascade filtering method for denoising these two types of noise based on their spatio-temporal distribution properties. Firstly, the event stream undergoes coarse filtering based on spatio-temporal density to remove BA noise, which exhibits sparsity in the spatiotemporal domain, and then proceeds to fine filtering. Secondly, leveraging the high temporal resolution of the event camera, we treat the event stream generated by each pixel within each time segment as a time series. To address the characteristic of hot noise, which exhibits a fixed high-frequency pattern in the spatiotemporal domain, we propose a time sequence analysis method to filter out hot noise. Our method primarily aims to enhance the robustness of denoising across various noise environments. Therefore, in the fine filtering stage, we also consider the impact of residual BA noise. Furthermore, we propose an adaptive overlap-based event stream visualization method, which adjusts the number of events from the previous frame that are overlapped onto the current frame based on the overlapping strategy. The main contributions of this work can be summarized as follows.

- An event stream denoising method based on spatio-temporal density and temporal sequence analysis is proposed. The denoising process is divided into coarse filtering and fine filtering, and after establishing the spatio-temporal neighborhood of events, the noise is removed sequentially according to the spatio-temporal distribution characteristics of different noises.
- An adaptive overlap-based event stream visualization method is proposed for the event stream density change over time, which dynamically adjusts the overlap time

according to the change of the event density of neighboring frames, and obtains clear and information-coherent event frames for further event camera-based research.

2. Related Work

In this section, we present and discuss the classical event stream noise reduction methods and event stream visualization methods.

2.1. Existing Event Stream Noise Reduction Algorithms

In recent years, a large number of scholars have conducted in-depth research on event camera denoising, and the event stream denoising method based on spatio-temporal correlation is one of the main research directions, which removes noise by targeting the regular difference between noise and real events in time and space. Wu et al. [26] introduced a denoising method based on a probabilistic undirected graph model, proposed a priori knowledge based on the working principle of DVS and the object motion law, constructed a probabilistic undirected graph model, transformed the event stream denoising problem into a model energy minimization optimization problem, and optimized the model by using the improved ICM to obtain the denoised event stream. The model is built based on the distribution characteristics of BA noise, which does not take into account the influence of hot noise. Feng et al. [27] proposed a denoising method based on event density. Firstly, spatio-temporal neighborhood and event density matrices are built using coarse filtering, and the BA noise is removed by calculating the L1 paradigm number of the matrix. The valid events are stored in the coarse filtering results, then the event density matrices are established and the scintillation noise judgment is used to filter out the hot noise using the flicker noise determination value R.

Khodamorad et al. [28] proposed a novel spatio-temporal filter with memory complexity that stores fewer timestamps in a specific way based on the sparse random nature of BA noise. This allocates two memory cells per row and column to store the latest event in the whole row or column, and if two events are captured in the same but different rows in a short window of time, the most recent event overwrites the old event in memory. The complexity of the filter created based on the new storage approach is reduced from $O(N^2)$ to $O(N)$, which improves memory utilization. However, it is difficult to remove dense noise using a single spatio-temporal filter.

To better utilize the spatio-temporal features, some scholars introduced neural networks to improve the event stream denoising performance. Event-denoising neural networks can be categorized into three types according to their input: 3D vector input, image input, and asynchronous event input. Ryumin et al. [29] proposed a research methodology for audio-visual speech recognition (AVSR) in driver assistance systems. Their spatio-temporal fusion strategy leverages contextual information from both the spatial and temporal domains, preserving the contextual integrity of both modalities while ensuring their synchronization. Arindam et al. [30] proposed a new spike neural network-based method NN2 for filtering noise events from data captured by asynchronous time-based image sensors on the IBM TrueNorth neural synapse system, a neural morphology processor. Fang et al. [31] proposed an event-based deep learning approach for the event denoising method AEDNet, which directly utilizes the raw DVS data from the event stream without destroying the inherent spatio-temporal correlation. The temporal and spatial information is processed separately using temporal window and spatial feature embedding modules, which decompose the events into temporal correlation and spatial affinity and process these two parts separately to remove the noise. In addition, the event frames obtained from the transformation of asynchronous event streams are denoised using a conventional image-denoising algorithm. Xie et al. proposed an image-based denoising method by visualizing the event stream to obtain event frames, which are subsequently denoised using traditional K-SVD [32], but this method loses the temporal attributes of the data, which seriously affects the high-speed characteristics of the event camera.

Overall, most existing event stream denoising methods only focus on filtering out random noise, ignoring the impact of hot noise or not considering the robustness of denoising algorithms in dense noise environments. Although simple algorithm designs may improve denoising efficiency, the accuracy of denoising may decrease, seriously affecting the further use of event streams.

2.2. Existing Approaches to Event Stream Visualization

Asynchronous event stream information form is difficult to process, and a single event contains too little information, so the processing needs to aggregate events for processing. In addition, the event stream information is not intuitive for human eyes to observe, so it needs to be visualized to advance to the next step. Related work based on event camera visualization can be briefly divided into two categories: 1. For the event stream itself, through a fixed time window and other ways to obtain the event frame, which is convenient for further observation and application processing; and 2. image reconstruction of the event stream information, to obtain the estimated intensity of the image to achieve the effect of visualization. Xie et al. [33] proposed an overlap accumulation-based method to convert events with periods from 0 ms to 30 ms to frame 0 and events with periods from 10 ms to 40 ms to frame 1, as shown in Figure 2a. Although it makes the event information very rich, it also brings the problem of information redundancy or incoherent information in event frames, which is not conducive to real-time detection and monitoring. Chen et al. [34] proposed an event stream visualization scheme to achieve high and low frame rate adjustment and verified its effectiveness on the FPGA platform. However, the method reduces the amount of information contained in the event frames at high frame rates, and the information obtained during smooth motion is too little for further processing. In addition, the traditional fixed time interval approach divides the event stream into multiple sub-segments according to a fixed time window starting from the timestamp of the first event, and all events within each sub-segment are accumulated and projected to the event frame at the end of that sub-segment as shown in Figure 2b. The fixed number of events approach divides the event stream into multiple sub-segments by counting every N event starting from the first event, and all events within each sub-segment are accumulated and projected to the event frame at the end of the sub-segment, as shown in Figure 2c.

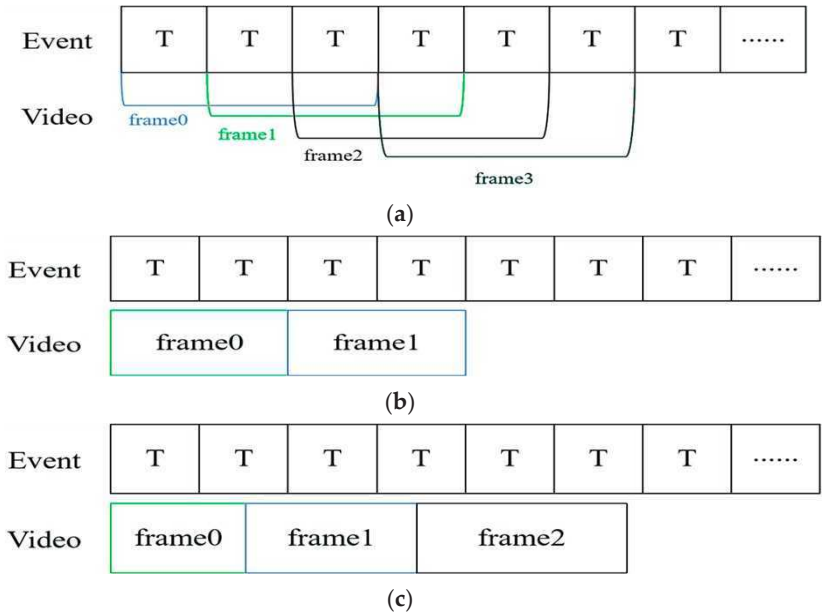


Figure 2. Event stream visualization methods for (**a**) overlapping accumulation, (**b**) fixed time intervals, and (**c**) fixed number of events.

3. Event Stream Denoising and Visualization Method

We proposed an event stream noise reduction and visualization algorithm. Based on the noise distribution characteristics, the event stream noise is categorized into BA noise and hot noise, and the event stream is sequentially coarse filtered and fine filtered. In this paper, we pay more attention to the optimization and improvement of the robustness of the denoising algorithm, so the parameter selection and the influence of the residual BA noise are taken into account in different filtering stages. The spatio-temporal neighborhood regularization and the free choice of threshold make our algorithm applicable to various noise environments. The visualization algorithm adaptively overlaps the events of the previous frame according to the change in event density.

3.1. Representation of Event Data

Event cameras asynchronously generate data that exhibits a sparse distribution in space. Typically, the event camera outputs events containing the pixel address, time, and type of event. The event stream is defined as $\{e_i\}$, where each event $e_i = (t_i, x_i, y_i, p_i)$ contains the coordinates of the event's position on the pixel plane (x_i, y_i), the timestamp of the event's occurrence t_i, and the polarity of the event p_i.

The polarity of an event is determined by the change in light intensity. If at time t_i, the luminance of a pixel point increases compared to the previous moment and this increase exceeds a predetermined threshold, the event is recorded as a positive polarity event $p_i = +1$; conversely, if the luminance decreases and exceeds the threshold, it is recorded as a negative polarity event $p_i = -1$. The polarity of an event can be defined by the following mathematical expression:

$$p = \begin{cases} +1, \log I(x, y, t + \Delta t) - \log I(x, y, t) > \Delta I \\ -1, \log I(x, y, t + \Delta t) - \log I(x, y, t) < \Delta I \end{cases} \quad (1)$$

where $\log \mathrm{I}(x, y, t)$ represents the logarithmic light intensity of the pixel with coordinate (x, y) at moment t, and ΔI represents the light intensity change threshold.

3.2. Event Stream Denoising Based on Spatio-Temporal Density and Temporal Analysis

Valid events are usually triggered by the motion of an object or by a change in lighting. Most of these activated pixels are adjacent to each other. Due to the handshake circuit design of the DVS, events in the same row of pixels have the same timestamp, while the difference in event times between pixels in neighboring rows is very small, usually measured in nanoseconds. The number of events generated by background activity (BA) in a given spatio-temporal region is usually much lower than the number of events generated by actual object motion. Since hot pixels are continuously activated and output events due to their inability to be properly reset, the location of occurrence is random, and thus the number of events occurring on hot pixels is usually much higher than that on neighboring pixels within a given spatio-temporal region. Based on the analysis of BA noise and hot pixel noise generation principle and spatio-temporal distribution characteristics, we are more concerned about how to target the removal of different noises. Figure 3 shows the framework of the denoising algorithm.

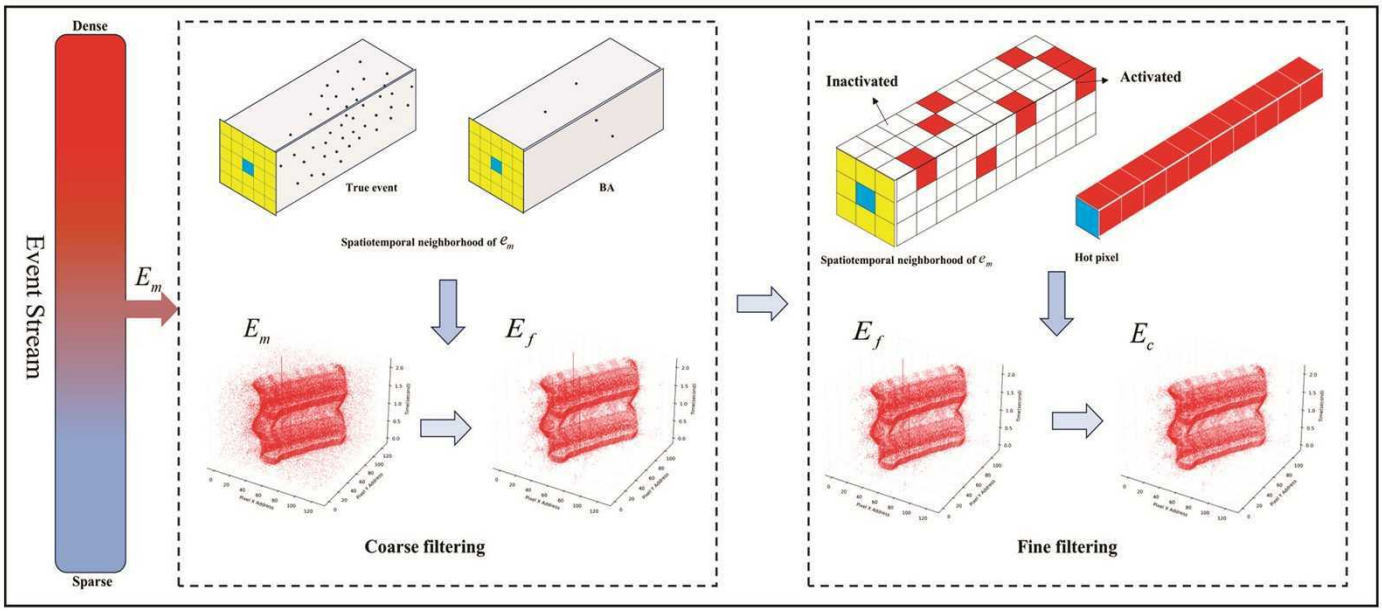


Figure 3. Proposed event stream denoising framework based on spatio-temporal density and time sequence analysis, where E_m represents the unprocessed event stream, E_f represents the event stream that enters fine filtering after coarse filtering, and E_c represents the processed event stream. Blue for newly arrived events, yellow for their spatial neighborhoods, and red for pixels activated.

3.2.1. Spatio-Temporal Density-Based Denoising of Event Stream

To improve the performance and robustness of the denoising algorithm, the information of temporal and spatial neighborhoods is comprehensively utilized in the coarse filtering stage to maximize the extraction of useful information and the suppression of noise, to make it more suitable for applications in various complex scenarios. The temporal and spatial neighborhoods are set for each event. The red region in Figure 4 is the newly arrived event e_m, which is located in the center of the spatial neighborhood of size $L_1 \times L_1$ (L_1 is an odd number), the temporal neighborhood is $(t_1 - \Delta t_1, t_1)$, and N_{local} denotes the number of events in the spatial and temporal neighborhoods $\Omega^{L_1}_{\Delta t_1}$. The yellow region is the spatial neighborhood of e_m.

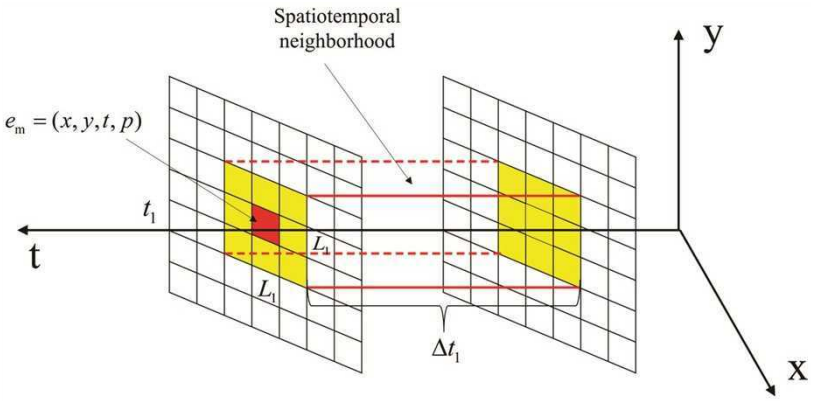


Figure 4. Spatio-temporal neighborhood of e_m.

All events in the spatio-temporal neighborhood are counted, and the density of the spatio-temporal neighborhood is calculated. If the event spatio-temporal density is greater

than a threshold, it is considered a valid event; if it is less than the threshold, it will be considered noise. The spatio-temporal density is defined as follows:

$$D(e_m) = \frac{N_{local}(e_m)}{{L_1}^2 \cdot \Delta t_1} \tag{2}$$

where $N_{local}(e_m)$ is the total number of events in the spatio-temporal neighborhood of e_m.

The noise identification formula of e_m is as follows:

$$e_m = \begin{cases} Noise,\ if\ D(e_m) < DT \\ Vaild,\ if\ D(e_m) \geq DT \end{cases} \tag{3}$$

where DT is a preset threshold for distinguishing BA noise from valid events.

DT can be dynamically adjusted to adapt to different noise environments according to the characteristics of the actual scene. The regularization of the number of events is achieved by dividing the number of events in the neighborhood by the volume of the neighborhood. This means that the density value reflects the level of event activity per unit volume regardless of the size of the neighborhood. Evaluating event densities from a relative perspective prevents the results from being unfairly affected by changes in neighborhood size. Comparing the spatio-temporal density of each event with the density threshold effectively filters out the BA noise and retains the valid events in the coarse filtering results for fine filtering.

3.2.2. Event Stream Denoising Based on Timing Analysis

Event cameras upload a fixed format event when the brightness value of a pixel at a location changes, so pixels can be categorized into active and inactive states during operation. A hot pixel is continuously activated at a fixed location and there may be real events or BA noise in the neighboring pixels. The spatio-temporal density approach does not provide effective removal of hot noise. Therefore, it is necessary to fine filter the event stream after coarse filtering. The hot noise filtering method used by Feng assumes that in an ideal environment, there is no BA noise in the hot noise spatio-temporal neighborhood, or only a very small amount of noise exists. In the experimental process of Section 4, it is found that all BA noise often cannot be effectively filtered out in a very dense noise environment, and at this time the accuracy of the denoising algorithm for hot noise will show a significant decrease. Therefore, we use the denoising algorithm based on time sequence analysis to remove hot noise.

For the coarsely filtered events, a new spatio-temporal neighborhood $\Omega_{\Delta t_2}^{L_2}$ is constructed. The working states of the event pixels as well as the neighboring pixels are sampled within Δt_2, with 1 representing a pixel activated and 0 representing a pixel inactive, and a Boolean time series is obtained at the end of the sampling. The working state $S \in \{1, 0\}$ of the pixel is recorded at the sampling time point t_k, then the event pixel is obtained as a time sequence $\left\{S_i(t_0), S_i(t_1), S_i(t_2), \cdots, S_i\left(t_{f \cdot \Delta t_2 - 1}\right)\right\}$. For each spatial neighborhood pixel of pixel i, the corresponding time series can be obtained $\left\{S_j(t_0), S_j(t_1), S_j(t_2), \cdots, S_j\left(t_{f \cdot \Delta t_2 - 1}\right)\right\}$, $j \in N(i)$, f is the sampling frequency.

Hot pixels exhibit persistent activation states, and neighboring pixels usually have a much lower proportion of activated states than hot pixels. The DC component of a time series after discrete Fourier transform indicates the average level of the time series, and for Boolean time series, this value reflects the proportion of 1 states in the time series. Hot noise is removed by comparing the DC component of an event stream pixel DC_i with the average DC component of its neighboring pixels $\overline{DC_j}$. Although a certain amount of hot noise can be filtered out by directly comparing the values of DC_i and of $\overline{DC_j}$, in order to

enable the noise reduction algorithm to be more flexible, a threshold value is introduced θ. The noise identification formula of e_m is as follows.

$$e_m = \begin{cases} Noise,\ if\ DC_i \geq \overline{DC_j} + \theta \\ Vaild,\ if\ DC_i < \overline{DC_j} + \theta \end{cases} \tag{4}$$

In the fine filtering algorithm, the choice of threshold θ is very important and determines the sensitivity and specificity of the determination. A threshold that is too low may result in too many valid events being misclassified as noise, while a threshold that is too high may result in failure to detect true hot noise. Therefore, it may be necessary to experiment and analyze specific data to adjust the threshold value to achieve the best denoising performance.

3.3. An Adaptive Overlap-Based Event Stream Visualization Method

Motion state changes and scene variations during event camera operation can lead to an uneven amount of event information per unit of time. Compared with visualization methods that simply fix the number of events or a fixed time window, the overlapping cumulative event stream visualization method optimizes the visual presentation of a dynamic scene by sharing events between consecutive frames and achieves an increase in the information richness of each frame while maintaining a high frame rate. This, to some extent, enables the event frames to ensure the continuity of event information despite the uneven amount of event information. However, the fixed overlap parameter is difficult to apply to various complex scenes.

In this paper, an adaptive overlap time strategy is proposed to dynamically adjust the overlap time according to the change in event density. The adaptive strategy can make it possible to increase the overlap to capture more details when the scene changes faster and decrease the overlap to reduce redundancy when the scene changes slower. A baseline event density threshold is defined, the number of events in the i-th time slice is counted, and the time at which the i+1-th time slice overlaps the previous frame is determined by Equation (5). Since the dynamic scene changes in different degrees, it is difficult to use a fixed step size to adjust the overlap time, so the step size value is based on the event density difference. Define the time slice length and overlap time O. The expression for adaptive overlap of events is as follows.

$$O_{i+1} = O + sign(d - D_i) \cdot (k \cdot |d - D_i|) \tag{5}$$

where D_i is the number of events in the i-th time-slice length, k is the step adjustment factor, and O_{i+1} is the time at which the i+1-th time-slice overlaps the previous frame.

The noise processing and visualization process is shown in Figure 5. A set of event streams with uneven density distribution are filtered by coarse fine filtering, and then clear and coherent event frames are obtained based on the adaptive overlapping method. The event stream's color represents the events' density (black is the densest, and white is the sparsest). Different colors represent the events mapped to each event frame. The adaptive overlap method ensures flexible event quantity mapping and event overlap for each event frame.

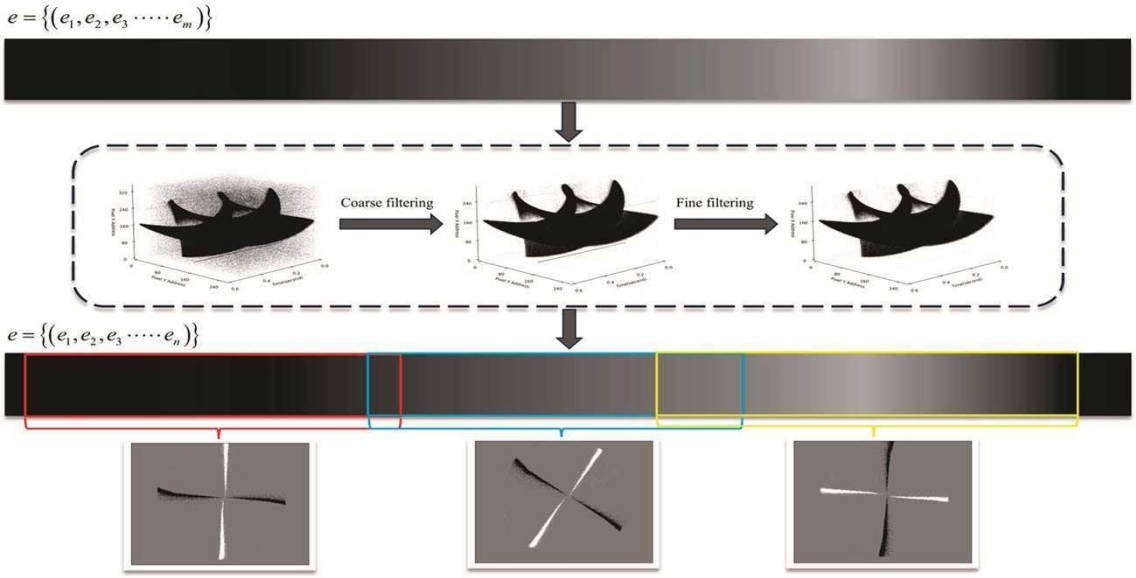


Figure 5. Denoising and visualization process, The event stream passes through the cascaded filter, filtering out n-m events. Three color boxes represent three event frames. The overlapping areas between the boxes represent the events that overlap with the previous frame.

4. Experiment and Evaluation

The experiments in this chapter are divided into three sections. Section 4.1 captures a real scene using the Davis346 event camera and denoises the dataset using different denoising algorithms. Section 4.2 adds random noise to multiple public AER datasets and compares the denoising results of different denoising algorithms. Section 4.3 evaluates the visualization algorithms. Sections 4.1 and 4.2 use the filters of NN2, Khodamoradi, and Feng for comparison with our algorithm. The computer processor is an Intel(R) Core(TM) i7-9700 CPU at 3.3 GHz, 8 GB of RAM, and the operating system is Windows 10 64-bit.

4.1. Event Stream Denoising Experiments in Real Scenes

In this experimental phase, we use a Davis346 with a spatial resolution of 346 × 260. Because event cameras are very good at capturing fast-moving objects, in Scene I we use it to capture a fast-moving soccer ball, which goes from slow to fast until it is completely thrown. In Scene II, we use Davis346 to record a black and white disk rotating at high speed, and the rotation speed of the disk changes from fast to slow with time. Due to the experimental scene with more factors interfering, the event stream output by Davis346 contains a lot of noise.

Since there is no base truth value of comparable events from real-world data, NIR (Noise in Real) and RIN (Real in Noise) are used to evaluate the denoising performance of different algorithms in real-world scenarios [25]. NIR measures the number of noisy events that are erroneously retained after denoising, and smaller NIR values indicate that the denoising algorithm is more effective in removing noise.

$$NIR = \sum \delta(x - x_i, y - y_i, t - t_i)\ e_i(x_i, y_i, t_i) \in \Lambda_{PE} \text{ and } \mathrm{P}_{ei} \leq P_{ARE} \tag{6}$$

where Λ_{PE} is the set of events delivered during the cycle, and P_{ARE} is the average true event probability of the BA computed over events with no target motion region.

RIN measures the number of real events that were mistakenly deleted; smaller RIN values mean that more real and valid events were retained while removing noise.

$$RIN = \sum \delta(x - x_i, y - y_i, t - t_i)\ e_i(x_i, y_i, t_i) \in \Lambda_{PE} \text{ and } \mathrm{P}_{ei} > P_{ARE} \tag{7}$$

where Λ_{FE} is the set of filtered events in the cycle.

In the proposed algorithm, we have multiple parameters and calibrate some of them based on the data used. The values of the parameters should vary depending on the application field and be minimized to achieve more accurate denoising. The method used in this paper uses the parameters $L_1/\Delta t_1 = 7/3$ ms, $L_2/\Delta t_2 = 5/6$ ms, DT = 10, and $\theta = 15$. The parameter in the Feng filter is set to $L/\Delta t = 5/5$ ms, and the parameter in the Khodamoradi filter is set to 2 ms. It should be noted that in the algorithm proposed in this paper, there is a minimum spatio-temporal neighborhood setting $L/\Delta t = 3/1$ ms. This parameter implies the strictest noise discrimination, although this does not apply to the vast majority of scenarios. We adjusted the parameters with the minimum spatio-temporal neighborhood as a benchmark, so the choice of algorithmic parameters for each scenario is based on the results of multiple experiments. Moreover, in the choice of parameters for the three comparative algorithms, we used the default parameters provided in the paper. Figure 6 shows the denoising results of the different algorithms for two scenarios, Table 1 shows the change in the number of events before and after denoising for both scenarios, as well as the noise evaluation based on *RIN* and *NIR*.

In the denoising results shown in Figure 6 and Table 1, our algorithm filters out 19.09% of the events in Scene I, which is higher than 17.34% for NN2, 14.87% for the Feng filter, and 8.66% for the Khodamoradi filter. In scene II, our algorithm filters out 19.04% of the events, higher than 16.09% for NN2, 13.9% for the Feng filter, and 9.57% for the Khodamoradi filter. The RIN and NIR values of this paper's algorithm in Scene I do not show a significant advantage due to the discontinuous and irregular distribution of the background noise region, but the 2D plots and spatio-temporal images show that our algorithm filters out more noise compared to other algorithms. This suggests that the reliability of RIN and NIR denoising assessment based on manual selection of background noise regions still needs to be enhanced.

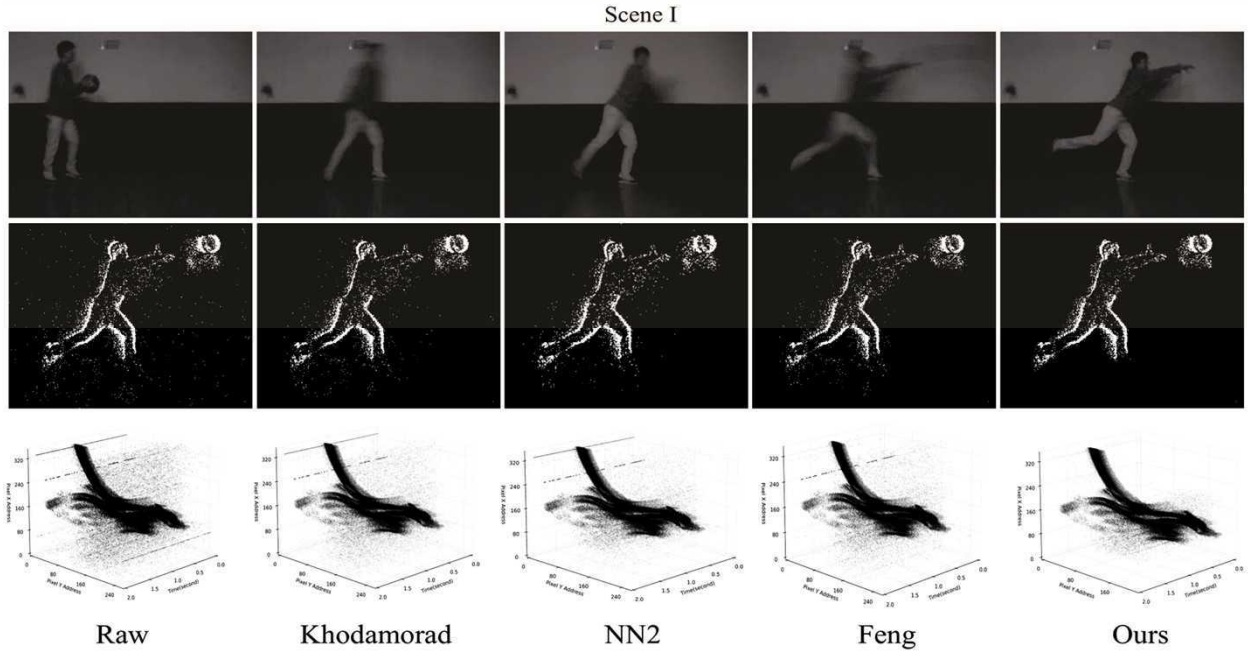


Figure 6. *Cont*.

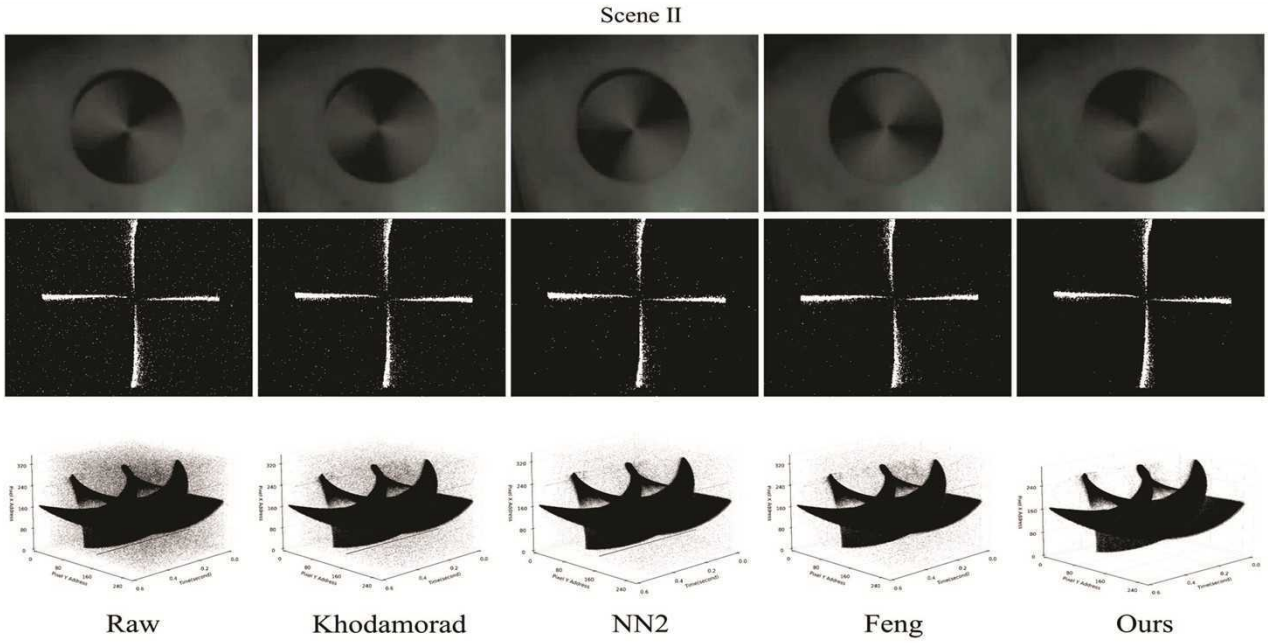


Figure 6. Event denoising results. The first row is the output of the frame-based camera, and the second and third rows are the 2D maps and spatio-temporal images of the original event stream and the event stream after denoising by different methods.

Table 1. Noise evaluation of Scene I and Scene II.

Scene	Index	Raw Data	Khodmradi	NN2	Feng	Ours
I	Number of events	196,232	179,243	162,203	167,043	158,882
	RIN	/	7029	5330	5208	5294
	NIR	/	29,445	8401	8790	8328
II	Number of events	731,014	661,027	609,112	629,428	591,840
	RIN	/	12,936	6230	8979	5022
	NIR	/	134,502	49,389	72,103	32,357

4.2. Denoising the AER Datasets

Denoising the event stream output by the event camera as a preprocessing method can facilitate the subsequent obtaining of clearer event frames. To verify the denoising performance of our proposed algorithm in different noise environments, we designed the following experiments. Events in the original AER dataset are treated as valid events, and a certain percentage (10%, 20%, 50%, 100%, 150%, and 200%) of noise (BA noise and hot noise) is added. In the denoising experiments of the AER dataset, we use the event denoising precision (EDP) and the event Signal to Noise Ratio (ESNR) to measure the accuracy of the denoising algorithm in removing noise [26]. EDP is a measure of whether the denoising algorithm effectively filters out the noise in the event stream, and the larger the value, the better the denoising effect. ESNR is a measure of the quality of the event stream, and a high ESNR value denotes a lower noise level and a higher quality of the event stream.

$$EDP = \frac{N_{effective}}{N_{total}} \tag{8}$$

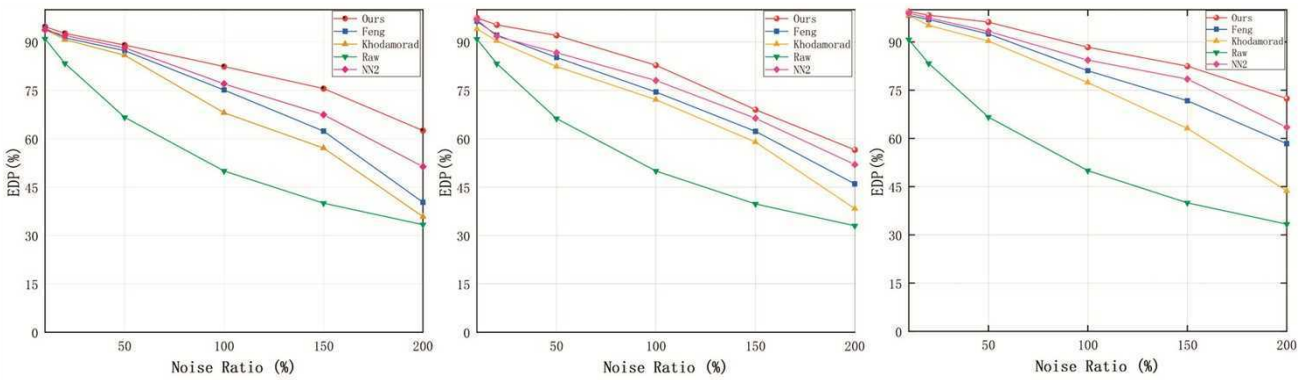
where $N_{\text{effective}}$ is the total number of valid events in the event stream and N_{total} is the total number of events in the event stream.

$$ESNR = 10\lg\left(\frac{N_{noise}}{N_{\text{effective}}}\right) \tag{9}$$

where N_{noise} is the total number of noise events in the event stream.

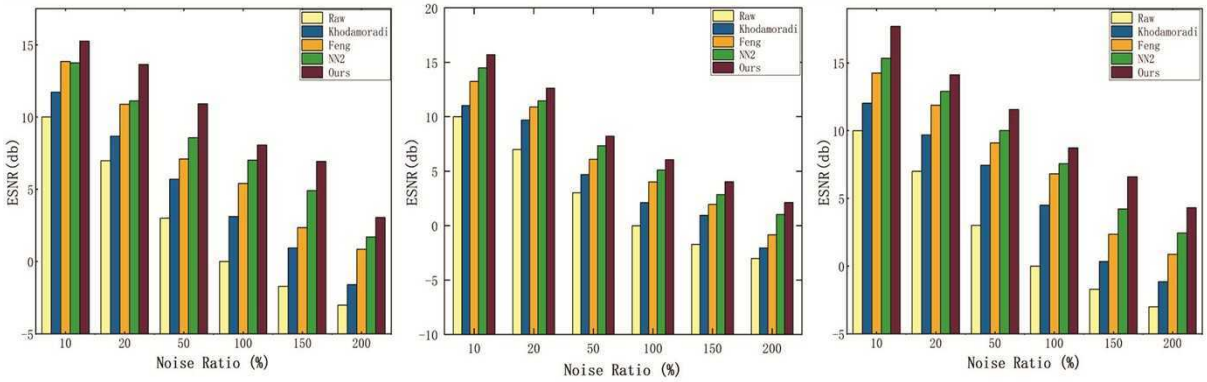
Experiments were performed on the public AER datasets N-MNIST [35], CIFAR10-DVS [36], and MNIST-DVS [37]. The MNIST-DVS dataset was zoomed from the original MNIST dataset image to three scales (Scale 4, Scale 8, and Scale 16) and displayed in slow motion on an LCD monitor. Event-based sensors were then used to record the number of movements. The MNIST-DVS dataset with a scale of 4 was used in the experiments. CIFAR10-DVS is an event dataset for object classification. The 10,000 frame-based images from CIFAR-10 are converted into 10,000 event streams by DVS with a resolution of 128 × 128 pixels, and the conversion is achieved by using Repetitive Closed Loop Smoothing (RCLS) motion on the frame-based images. The N-MNIST dataset is a spiked version of the MNIST dataset, containing the same samples as the original MNIST dataset and captured at the same visual scale (28 × 28 pixels) as the original MNIST dataset. The N-MNIST dataset is captured by mounting the ATIS sensor on a motorized gimbal and causing the sensor to move while viewing the MNIST example on the LCD monitor.

In this section of experiments, the method of this paper uses the parameters $L_1/\Delta t_1 = 7/3$ ms, $L_2/\Delta t_2 = 5/6$ ms, DT = 10, and $\theta = 15$. The parameter in the Feng filter is set to $L/\Delta t = 7/5$ ms and the Khodamoradi filter parameter is set to 2 ms. Figures 7 and 8 show the ESNR vs. EDP for the three datasets N-MNIST, CIFAR10-DVS, and MNIST-DVS.



(**a**) (**b**) (**c**)

Figure 7. EDP comparison of filter algorithm and our algorithm on different datasets. (**a**) CIFAR10-DVS; (**b**) MNIST-DVS; (**c**) N-MNIST.



(**a**) (**b**) (**c**)

Figure 8. ESNR comparison of filter algorithm and our algorithm on different datasets. (**a**) CIFAR10-DVS; (**b**) MNIST-DVS; (**c**) N-MNIST.

As shown in Figures 7 and 8, when the noise is sparser, the denoising methods may be more likely to accurately restore the true event distribution, when the difference in the denoising effect of different algorithms is small. However, when the noise is denser, the

denoising methods may be more challenged, and the EDP and ESNR values show a larger magnitude decrease.

As can be seen from Tables 2 and 3, there is a positive correlation between EDP and ESNR, indicating that ESNR and EDP can reflect the quality of the event stream well. The performance of Feng's filter shows a significant degradation after the addition of dense noise, which is because, in a dense noise environment, BA noise may show some spatial-temporal correlation between them, and BA noise may also show some spatial-temporal correlation between them and the real events, and at this time, it is difficult to discriminate the noise, a lot of BA noise is retained after coarse filtering, which affects the denoising accuracy of the fine filtering. The Khodamorad filter can efficiently filter out most of the noise according to the spatio-temporal correlation in a sparse noise environment, and when the noise is denser, the denoising accuracy decreases seriously and the hot noise is not effectively removed. The denoising effect of NN2 shows better robustness in a more dense noise environment. However, NN2 cannot effectively remove hot noise, because hot noise has a certain spatial-temporal correlation and the db value is higher than BA noise, so it is difficult to distinguish it from real events during testing.

Table 2. Comparison of the performance of different denoising algorithms on EDP.

Dataset	Algorithm	Noise Ratios					
		10%	**20%**	**50%**	**100%**	**150%**	**200%**
CIFAR10-DVS	Raw	90.95	83.33	66.67	50	39.99	33.33
	Khodamorad	93.92	90.76	85.88	68.05	57.13	35.82
	NN2	93.76	91.99	88.21	77.02	67.45	51.3
	Feng	94.21	91.29	87.33	75.09	62.35	40.32
	Ours	94.73	92.69	89.03	82.41	75.57	62.51
MNIST-DVS	Raw	90.92	83.27	66.27	50	39.76	33.03
	Khodamorad	94.02	90.35	82.37	72.12	58.96	38.32
	NN2	96.96	91.75	86.63	78.02	66.34	51.94
	Feng	96.43	92.17	85.2	74.48	62.33	45.99
	Ours	97.47	95.34	92.04	82.79	69.01	56.53
N-MNIST	Raw	90.74	83.31	66.61	50	39.96	33.33
	Khodamorad	98.22	95.16	90.26	77.34	63.11	43.69
	NN2	98.95	97.46	93.26	84.32	78.45	63.44
	Feng	98.23	96.93	92.46	81.03	71.69	58.37
	Ours	99.44	98.31	96.15	88.36	82.45	72.39

Our algorithm shows advantages in EDP and ESNR metrics, and it is noteworthy that the reduction in EDP and ESLR is the same in different proportions of noise environments, and there are no problems with significant reduction due to dense noise environments. Due to the regularization of the spatio-temporal neighborhood in the coarse filtering, the spatio-temporal density is not unfairly affected by the variation of the neighborhood size, and the vast majority of BA noise is filtered out in the coarse filtering stage. This avoids the drawback of counting the number of events in the dense noise environments based on the spatio-temporal correlation theory, and the method based on the temporal analysis in the fine filtering ensures that the thermal noise can be accurately filtered out even in the presence of the BA noise. However, more complex algorithms imply that they will occupy more memory and require more computation. In our experiments, we found that our proposed algorithm does not show an advantage in time, which is due to the fact that in fine filtering, the computation is increased due to the temporal analysis of the pixels where each event is located.

Table 3. Comparison of the performance of different denoising algorithms on ESNR.

Dataset	Algorithm	Noise Ratios					
		10%	20%	50%	100%	150%	200%
CIFAR10-DVS	Raw	10	6.98	3.01	0	−1.72	−3.02
	Khodamorad	11.72	8.68	5.7	3.12	0.94	−1.6
	NN2	13.74	11.12	10.92	8.06	6.93	3.04
	Feng	13.85	10.88	7.09	5.41	2.34	0.86
	Ours	15.26	13.64	10.92	8.06	6.93	3.04
MNIST-DVS	Raw	10	6.98	3.01	0	−1.72	−3.02
	Khodamorad	11.02	9.68	4.7	2.12	0.94	−2.06
	NN2	14.49	11.46	7.34	5.11	2.84	1.02
	Feng	13.25	10.88	6.09	4.01	1.94	−0.86
	Ours	15.7	12.64	8.2	6.06	4.03	2.14
N-MNIST	Raw	10	6.98	3.01	0	−1.72	−3.02
	Khodamorad	12.02	9.68	7.45	4.49	0.34	−1.16
	NN2	15.33	12.91	10.01	7.55	4.21	2.44
	Feng	14.25	11.88	9.09	6.81	2.34	0.86
	Ours	17.7	14.12	11.56	8.72	6.59	4.3

4.3. Event Stream Visualization

To verify and analyze the effect of event stream visualization, this section experiments on a real dataset as well as a public dataset. The real dataset uses the fast-rotating black-and-white turntable recorded in Section 4.1. As shown in Figure 9, the turntable rotates at a slower speed in Scene 1, and the event stream is sparse. The turntable rotates faster in Scene 2. The time stream is denser. The public event dataset was acquired by the Sensor Laboratory at ETH Zurich using a DAVIS 240C model event camera [38], with the urban part as the scene for the experiment as shown in Figure 10. In Scene 1, the motion state of the camera is smoother and the event stream is sparse. In Scene 2, the camera motion state is more violent and the event stream is denser. Table 4 shows the number of events in different scenes as well as time. The experiments compare four event frame formation methods: adaptive overlapping, fixed time window (FW), fixed number of events (FN), and overlapping accumulation (OA). The frame rate, the average number of events, and the variance of the number of events obtained by different visualization methods can reflect to a certain extent the uniformity of the information contained in the event frames, regardless of the intensity or slowness of the operation. Therefore, we use the above three indicators to evaluate the event frames obtained by different methods, as shown in Tables 5 and 6.

In this section of experiments, the method of this paper uses the parameters $O = 5$ ms, $k = 0.001$. The results of the event flow visualization are shown in Figures 9 and 10, where black represents negative events and white represents positive events.

Since scene changes as well as motion state changes lead to uneven distribution of events in time, accumulating events according to a fixed time window results in event frames containing different amounts of information, blurring of event edges when motion is smooth, and severe shadowing of object edges due to event accumulation when motion is violent. The overlapping accumulation method can both increase the frame frequency and ensure the same amount of event information as that of a fixed time window. However, when the density of the event stream changes, a single overlapping strategy can lead to redundant or incoherent information. Fixing the number of events ensures a consistent amount of information, but when the motion is very smooth, it leads to long output frames containing a large number of outdated events, thus losing the low-latency advantage of event cameras. As shown in Tables 5 and 6, the fixed time window method has a fixed output frame rate and the largest event variance. A fixed output frame rate implies a stable number of output frames, but a large event variance indicates a large difference in the amount of information between frames. The overlapping accumulation method has the largest output frame rate and the second-largest event variance. The maximum output

frame rate means that more motion details can be captured, but it also contains a large amount of duplicate content. On the other hand, the fixed number of events method has the smallest event variance, but also the lowest frame rate. The adaptive overlapping approach did not show a clear advantage in one metric or the other but ensured a higher frame rate output while avoiding too much information variance between frames.

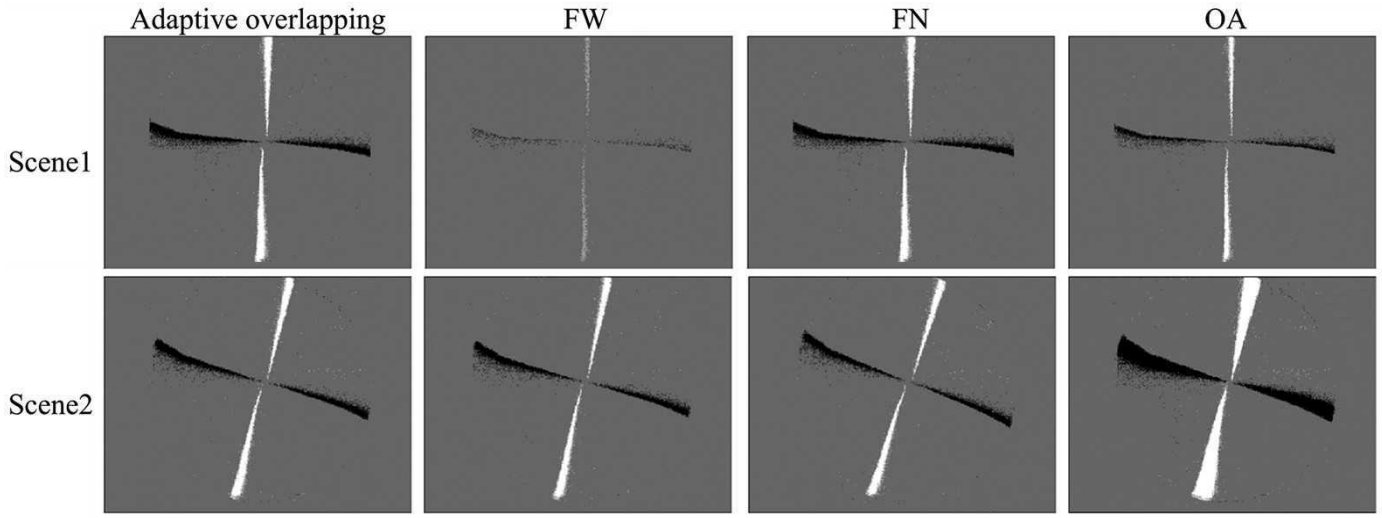


Figure 9. Visualization results for black and white turntable. The time window is 10 ms, and the number of events are 8000, $O = 5$ ms, and $k = 0.001$.

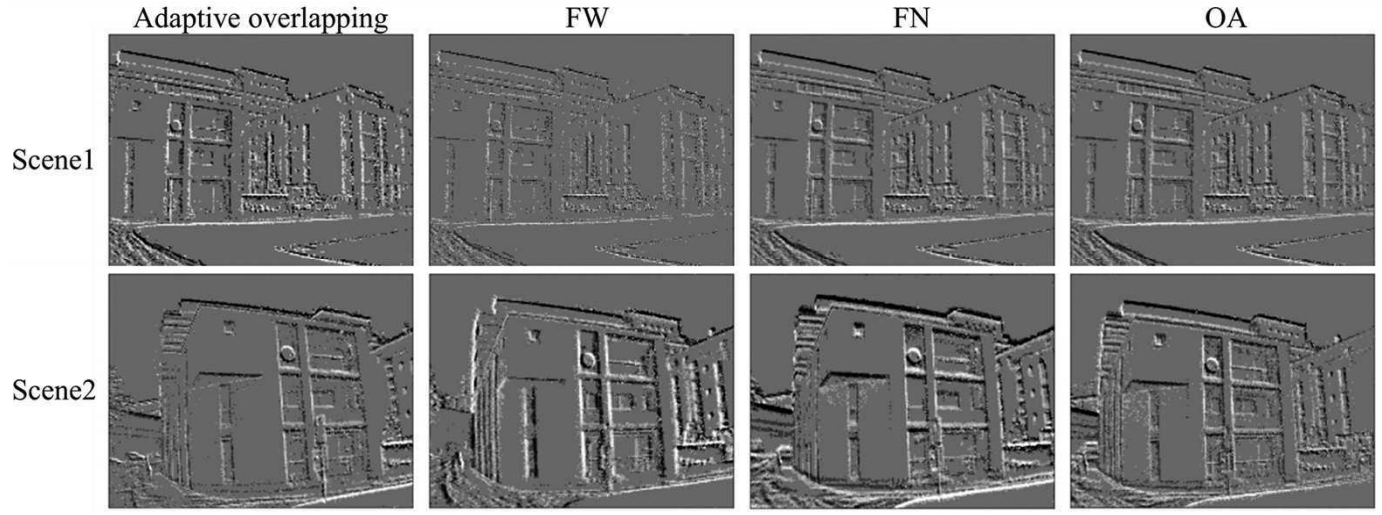


Figure 10. Visualization results for urban dataset. The time window is 10 ms, and the number of events are 12,000, $O = 5$ ms, and $k = 0.001$.

Table 4. Parameters for different scenes.

Dataset	Scene	Event Number	Time (s)
Turntable	1	184,402	1.2
	2	284,125	1.15
Urban	1	384,246	0.9
	2	620,143	1.1

Table 5. Performance comparison of four visualization methods for black and white turntable dataset.

	Algorithm	Evaluation Metrics		
		Frame Rate (fps)	Average Number of Events	Variance
Scene 1	FW	199.97	2234.2	1,523,975.02
	FN	44.61	7993.11	1973.29
	OA	322.94	3749.12	1,493,276.06
	Ours	157.49	7319.86	1,052,987.34
Scene 2	FW	196.32	7294.32	1,347,349.45
	FN	54.13	7998.21	447.96
	OA	380.17	1492.21	1,283,601.73
	Ours	165.49	6971.13	923,839.03

Table 6. Performance comparison of four visualization methods for urban dataset.

	Algorithm	Evaluation Metrics		
		Frame Rate (fps)	Average Number of Events	Variance
Scene 1	FW	195.38	5825.63	3,413,492.31
	FN	40.22	11,982.33	4216.03
	OA	375.94	11,049.12	3,013,484.39
	Ours	169.49	9932.36	2,269,272.26
Scene 2	FW	192.91	13,438.93	5,312,380.2
	FN	49.43	11,970.21	10,235.27
	OA	417.68	10,492.75	4,526,249.37
	Ours	201.33	9523.19	3,028,346.12

5. Conclusions

In this paper, an event stream denoising method based on spatio-temporal density and temporal sequence analysis is proposed. Based on the working principle of the event camera and the noise distribution characteristics, the noise is categorized into BA noise and hot noise, and the event stream is sequentially coarse filtered and fine filtered. We pay more attention to the optimization and improvement of the robustness of the denoising algorithm, so the parameter selection and the influence of the residual BA noise are taken into account in different filtering stages. The spatio-temporal neighborhood regularization and the free choice of threshold make our algorithm applicable to various noise environments. We use EDP and ESNR to evaluate the denoising effect of the AER dataset, and NIR and RIN to evaluate the denoising effect of the real scene. In addition, an event stream visualization method based on adaptive overlapping is proposed to validate the visualization algorithm by selecting some scenes in two datasets based on adaptive overlapping events of camera motion state changes or scene changes.

In future research and development work, we will continue to optimize our algorithm, focusing on further reducing the complexity of our algorithm so that it can be used in tight resource environments, and we will explore ways to integrate denoising techniques into real-time systems.

Author Contributions: H.J. is responsible for the methodology, programming, and supervision of the paper. X.W.: Writing-Review and Editing, programming. W.T.: Writing-Review and Editing, drawing, data organization. Q.S. (Qinghui Song): Supervision, data analysis, Writing—Review and Editing. Q.S. (Qingjun Song): Supervision, proofreading, Writing—Review and Editing. W.H.: proofreading, Writing—Review and Editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (52174145); Innovation capability improvement project of scientific and technological small and medium-sized enterprises of Shandong Province China (2022TSGC1271, 2023TSGC0620); Science and Technology Innovation Development Project of Tai'an City Shandong Province China (2021ZDZX006, 2023GX023).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the research results of this paper can be obtained from the corresponding author upon request.

Acknowledgments: The authors would like to thank the entire team that took part in validating the system.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Delbrück, T.; Linares-Barranco, B.; Culurciello, E.; Posch, C. Activity-driven, event-based vision sensors. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; pp. 2426–2429.
2. Gallego, G.; Delbrück, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A.J.; Conradt, J.; Daniilidis, K.; et al. Event-Based Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 154–180. [CrossRef] [PubMed]
3. Scheerlinck, C.; Rebecq, H.; Gehrig, D.; Barnes, N.; Mahony, R.E.; Scaramuzza, D. Fast Image Reconstruction with an Event Camera. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 156–163.
4. Pan, L.; Scheerlinck, C.; Yu, X.; Hartley, R.; Liu, M.; Dai, Y. Bringing a Blurry Frame Alive at High Frame-Rate with an Event Camera. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6813–6822.
5. Scheerlinck, C.; Barnes, N.; Mahony, R. Continuous-time intensity estimation using event cameras. In Proceedings of the Asian Conference on Computer Vision, Perth, WA, Australia, 2–6 December 2018; pp. 308–324.
6. Reinbacher, C.; Graber, G.; Pock, T. Real-time intensityimage reconstruction for event cameras using manifold regularization. In Proceedings of the British Machine Vision Conference, York, UK, 19–22 September 2016; pp. 1–12.
7. Barua, S.; Miyatani, Y.; Veeraraghavan, A. Direct face detection and video reconstruction from event cameras. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Lake Placid, NY, USA, 7–10 March 2016; pp. 1–9.
8. Rebecq, H.; Gallego, G.; Mueggler, E.; Scaramuzza, D. EMVS: Event-Based Multi-View Stereo—3D Reconstruction with an Event Camera in Real-Time. *Int. J. Comput. Vis.* **2018**, *126*, 1394–1414. [CrossRef]
9. Zhou, Y.; Gallego, G.; Rebecq, H.; Kneip, L.; Li, H.; Scaramuzza, D. Semi-dense 3D reconstruction with a stereo event camera. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 235–251.
10. Carneiro, J.; Ieng, S.H.; Posch, C.; Benosman, R. Event-based 3D reconstruction from neuromorphic retinas. *Neural Netw.* **2013**, *45*, 27–38. [CrossRef] [PubMed]
11. Zou, D.; Guo, P.; Wang, Q.; Wang, X.; Shao, G.; Shi, F.; Li, J.; Park, P.K. Context-aware event-driven stereo matching. In Proceedings of the IEEE International Conference on Image Processing 2016, Phoenix, AZ, USA, 25–28 September 2016; pp. 1076–1080.
12. Pérez-Carrasco, J.A.; Zhao, B.; Serrano, C.; Acha, B.; Serrano-Gotarredona, T.; Chen, S.; Linares-Barranco, B. Mapping from frame-driven to framefree event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward ConvNets. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2706–2719. [CrossRef] [PubMed]
13. Bardow, P.; Davison, A.J.; Leutenegger, S. Simultaneous Optical Flow and Intensity Estimation from an Event Camera. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 884–892.
14. Zhu, A.Z.; Yuan, L.; Chaney, K.; Daniilidis, K. Live Demonstration: Unsupervised Event-Based Learning of Optical Flow, Depth and Egomotion. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 15–20 June 2019; p. 1694.
15. Benosman, K.; Ieng, S.-H.; Clercq, C.; Bartolozzi, C.; Srinivasan, M. Asynchronous frameless event-based optical flow. *NeuralNetw* **2012**, *27*, 32–37. [CrossRef] [PubMed]
16. Barranco, F.; Fermuller, C.; Aloimonos, Y. Contour motion estimation for asynchronous event-driven cameras. *Proc. IEEE* **2014**, *102*, 1537–1556. [CrossRef]
17. Rebecq, H.; Horstschaefer, T.; Scaramuzza, D. Real-time visual-inertial odometry for event cameras using keyframebased nonlinear optimization. In Proceedings of the British Machine Vision Conference 2017, London, UK, 4–7 September 2017; pp. 1–12.
18. Zhu, A.Z.; Atanasov, N.; Daniilidis, K. Event-based visual inertial odometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5816–5824.

19. Ramesh, B.; Zhang, S.; Lee, Z.W.; Gao, Z.; Orchard, G.; Xiang, C. Long-term object tracking with a moving event camera. In Proceedings of the British Machine Vision Conference, Newcastle, UK, 3–6 September 2018; p. 241.
20. Mitrokhin, A.; Fermüller, C.; Parameshwara, C.; Aloimonos, Y. Event-Based Moving Object Detection and Tracking. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018; pp. 1–9.
21. Rebecq, H.; Horstschaefer, T.; Gallego, G.; Scaramuzza, D. EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real-time. *IEEE Robot. Autom. Lett.* **2017**, *2*, 593–600. [CrossRef]
22. Censi, A.; Scaramuzza, D. Low-latency event-based visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014; pp. 703–710.
23. Vidal, A.R.; Rebecq, H.; Horstschaefer, T.; Scaramuzza, D. Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios. *IEEE Robot. Autom. Lett.* **2018**, *3*, 994–1001. [CrossRef]
24. Delbrück, T. Frame-free dynamic digital vision. In Proceedings of the International Symposium on Secure-Life Electronics Advanced Electronics for Quality Life and Society, Tokyo, Japan, 6–7 March 2008; pp. 21–26.
25. Changda, Y.; Xia, W.; Xin, Z.; Xuxu, L. Adaptive Event Address Map Denoising for Event Cameras. *IEEE Sens. J.* **2021**, *22*, 3417–3429.
26. Wu, J.; Ma, C.; Li, L.; Dong, W.; Shi, G. Probabilistic undirected graph based denoising method for dynamic vision sensor. *IEEE Trans. Multimed.* **2021**, *23*, 1148–1159. [CrossRef]
27. Feng, Y.; Lv, H.; Liu, H.; Zhang, Y.; Xiao, Y.; Han, C. Event Density Based Denoising Method for Dynamic Vision Sensor. *Appl. Sci.* **2024**, *10*, 2024. [CrossRef]
28. Khodamoradi, A.; Kastner, R. O(N)O(N)-Space Spatiotemporal Filter for Reducing Noise in Neuromorphic Vision Sensors. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 15–23. [CrossRef]
29. Ryumin, D.; Axyonov, A.; Ryumina, E.; Ivanko, D.; Kashevnik, A.; Karpov, A. Audio-visual speech recognition based on regulated transformer and spatio-temporal fusion strategy for driver assistive systems. *Expert Syst. Appl.* **2024**, *252*, 124159. [CrossRef]
30. Vandana, P.; Arindam, B.; Garrick, O. A noise filtering algorithm for event-based asynchronous change detection image sensors on truenorth and its implementation on truenorth. *Front. Neurosci.* **2018**, *12*, 118.
31. Fang, H.; Wu, J.; Li, L.; Hou, J.; Dong, W.; Shi, G. AEDNet: Asynchronous event denoising with Spatial-Temporal correlation among irregular data. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 1427–1435.
32. Xie, X.; Du, J.; Shi, G.; Yang, J.; Liu, W.; Li, W. DVS image noise removal using K-SVD method. In Proceedings of the International Conference on Graphic and Image Processing, Qingdao, China, 14–16 October 2017; pp. 1099–1107.
33. Xie, X.; Du, J.; Shi, G.; Hu, H.; Li, W. An Improved Approach for Visualizing Dynamic Vision Sensor and its Video Denoising. In Proceedings of the International Conference on Video and Image Processing, Singapore, 27–29 December 2017; pp. 176–180.
34. Huang, J.; Guo, M.; Chen, S. A dynamic vision sensor with direct logarithmic output and full-frame picture-on-demand. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 1–4.
35. Orchard, G.; Jayawant, A.; Cohen, G.K.; Thakor, N. Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.* **2015**, *9*, 437. [CrossRef] [PubMed]
36. Li, H.; Liu, H.; Ji, X.; Li, G.; Shi, L. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Front. Neurosci.* **2015**, *11*, 309. [CrossRef] [PubMed]
37. Serrano-Gotarredona, T.; Linares-Barranco, B. A 128 $\times$ 128 1.5% Contrast Sensitivity 0.9% FPN 3 µs Latency 4 mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Preamplifiers. *IEEE J. Solid-State Circuits* **2013**, *48*, 827–838. [CrossRef]
38. Mueggler, E.; Rebecq, H.; Gallego, G.; Delbruck, T.; Scaramuzza, D. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Robot. Res.* **2017**, *36*, 142–149. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

mathematics

MDPI

Article

Exploring the Diversity of Kink Solitons in (3+1)-Dimensional Wazwaz–Benjamin–Bona–Mahony Equation

Musawa Yahya Almusawa * and Hassan Almusawa

Department of Mathematics, Faculty of Science, Jazan University, P.O. Box 2097, Jazan 45142, Saudi Arabia; haalmusawa@jazanu.edu.sa

* Correspondence: malmusawi@jazanu.edu.sa

Abstract: The Wazwaz–Benjamin–Bona–Mahony (WBBM) equation is a well-known regularized long-wave model that examines the propagation kinematics of water waves. The current work employs an effective approach, called the Riccati Modified Extended Simple Equation Method (RMESEM), to effectively and precisely derive the propagating soliton solutions to the (3+1)-dimensional WBBM equation. By using this upgraded approach, we are able to find a greater diversity of families of propagating soliton solutions for the WBBM model in the form of exponential, rational, hyperbolic, periodic, and rational hyperbolic functions. To further graphically represent the propagating behavior of acquired solitons, we additionally provide 3D, 2D, and contour graphics which clearly demonstrate the presence of kink solitons, including solitary kink, anti-kink, twinning kink, bright kink, bifurcated kink, lump-like kink, and other multiple kinks in the realm of WBBM. Furthermore, by producing new and precise propagating soliton solutions, our RMESEM demonstrates its significance in revealing important details about the model behavior and provides indications regarding possible applications in the field of water waves.

Keywords: nonlinear partial differential equations; (3+1)-dimensional Wazwaz–Benjamin–Bona–Mahony equation; Riccati modified extended simple equation method; kink solitons; long waves

MSC: 32G34; 32W50; 34B15; 93C20

1. Introduction

Nonlinear Partial Differential Equations (NPDEs) possess applications in numerous fields of mathematics, engineering and physics. These frameworks efficiently explain a broad range of phenomena, including thermal conductivity, the dispersion of organic compounds, fluid mechanics, wave bending, magnetism, quantum field theory, hydrodynamics, and oceanic wave propagation. These NPDEs include renowned models such as Burgers' equation, the Fisher equation, the Boussinesq equation, the advection equation, and other coupled nonlinear equations [1–5].

Among the aforementioned nonlinear models, several well-known equations have been studied in detail in relation to NPDEs, including the regularized long-wave model, the modified Korteweg–de Vries (KdV) model and the Boussinesq model [6,7]. Benjamin, Bona, and Mahony initially presented the KdV model in 1972 to characterize wave motion below the surface in a homogeneous channel. This led to the development of the normalized KdV model known as the Benjamin–Bona–Mahony (BBM) model, which focuses on shallow wave propagation. The BBM model is used to analyse drifting waves and Rossby waves in rotating fluid systems [8–10]. The BBM model is provided as [11]

$$Q_t + Q_x + Q^n Q_x - Q_{xxt} = 0, \tag{1}$$

while the KdV model is articulated as [12]

$$Q_t + Q_x + QQ_x + Q_{xxx} = 0, \tag{2}$$

where the function $Q = Q(x, t)$ expresses the displacement of the surface of a fluid from its rest state or wave height. Moreover, the linear terms account for the propagation of wave, the Q_{xxt} and Q_{xxx} signify the dispersive terms, and the nonlinear term $Q^n Q_x$ represents wave interaction. A variety of waves, including surface pulses in water, prolonged signals in acoustic energy in periodic pebbles, nonlinear diffraction workflows, magnetic hydrodynamic waves in erythrocytes, and reverberating waves in compressible fluids, are modeled using the BBM equation and the KdV equation [13]. However, there are several notable differences between these two equations. The BBM equation and the KdV equation are equivalent only when low-frequency waves are considered, and the findings they yield about waves diverge. While the BBM equation is categorized as a regularity equation, the KdV equation serves as an existence equation [14]. The theory of the BBM model, including its uniqueness, stability, and consistency, is investigated in [15,16]. The BBM model has additionally been the focus of a great deal of research in recent years. There is general agreement that the BBM model is much simpler to solve than the KdV model; nonetheless, both equations possess stable solutions for single waves [17,18]. Sometimes the long-wave equation is used instead of the BBM equation. When $n = 2$, the BBM equation in Equation (1) changes to the modified BBM equation. Ref. [19] investigated the solutions to a number of issues, including the modified BBM model starting values.

A novel equation for describing the three-dimensional kinematics of water waves was proposed by Wazwaz in 2017 [20]. A variation of the Benjamin–Bona–Mahony equation is the WBBM equation. Consequently, Wazwaz assessed the subsequent manufacturer equations:

$$\begin{aligned} &Q_t + Q_x + Q^2 Q_y - Q_{xzt} = 0, \\ &Q_t + Q_z + Q^2 Q_x - Q_{xyt} = 0, \\ &Q_t + Q_y + Q^2 Q_z - Q_{xxt} = 0, \end{aligned} \tag{3}$$

where $Q = Q(t, x, y, z)$ is the differentiable functions that represent the amplitude of the wave function in the equation. The three-dimensional WBBM model, which is expressed as follows, is the main topic of this study:

$$Q_t + Q_x + (Q^3)_y - Q_{xzt} = 0. \tag{4}$$

In the literature, a plethora of numerical and analytical methods have been developed to address NPDEs. Despite the availability of numerical tools, analysts often opt for analytical methods because they may offer a comprehensive grasp of the underlying physical processes and accurate insights into the system's behavior. Analytical solutions for NPDEs are, nevertheless, hard to find. Consequently, research into creating analytical treatments for NPDEs is still crucial. Various analytical techniques have been developed to address this problem. The Khater method [21], Adomian decomposition method [22], (G′/G)-expansion method [23], variable separable method [24], Exp-function method [25], integral transforms method [26], homotopy perturbation method [27], differential transform method [28], Extended Direct Algebraic Method (EDAM) [29], and numerous others [30–32] are a few of these approaches. This method involves converting the NPDEs into Nonlinear Ordinary Differential Equations (NODEs) using variable transformation. Next, the NODEs are assumed to have a closed-form solution and are converted into systems of algebraic equations. The resulting system of algebraic equations may be solved to provide the soliton solutions for the appropriate NPDEs. The soliton solutions generated by RMESEM can aid in our comprehension of the system's behavior and underlying physical processes.

This work utilizes the strategic RMESEM for swiftly and correctly obtaining propagating soliton solutions to the (3+1)-dimensional WBBM problem. By using this updated

approach, we are able to find a large number of families of propagating soliton solutions for the WBBM model in the form of exponential, rational, hyperbolic, periodic, and rational hyperbolic functions. We also offer 3D, 2D, and contour graphics to further graphically depict the propagating behavior of the acquired solitons. These graphics clearly show the existence of kink solitons, such as anti-kink, twinning kink, bright kink, bifurcated kink, lump-like kink, and other multiple kinks in the context of WBBM. Furthermore, by producing new and accurate propagating soliton solutions, our employed RMESEM demonstrates its value in revealing important details about the equation dynamics and provides indications regarding possible applications in the field of water waves.

The remaining paperwork is organized as follows: The working mechanism of the RMESEM is explained in Section 2, the WBBM model is addressed in Section 3 to create new soliton solutions, the propagating behavior of kink solitons is visually shown in Section 4, the results are summarized in Section 5, and the final section presents Appendix A.

2. The Methodical Strategy of RMESEM

The literature has established a wide range of analytical methods to study soliton occurrences in nonlinear models [33–35]. In this part, we describe the operation mechanism of the improved RMESEM [36]. Examining the upcoming NLPDE,

$$F(Q, Q_t, Q_{\tau_1}, Q_{\tau_2}, QQ_{\tau_1}, \dots) = 0, \tag{5}$$

where $Q = Q(t, \tau_1, \tau_2, \tau_3, \dots, \tau_r)$.

Equation (5) is to be solved using the following steps:

1. First, a variable-form wave transformation $Q(t, \tau_1, \tau_2, \tau_3, \dots, \tau_r) = P(\phi)$ is carried out. There are several definitions to represent ϕ. Through this transformation, Equation (5) is modified to obtain the following NODE:

$$G(P, P'P, P', \dots) = 0, \tag{6}$$

where $P' = \frac{dP}{d\phi}$. On occasion, the NODE may be made to comply with the homogeneous balance requirement by using integrating Equation (6).

2. Next, we suppose a series-based solution with the help of the solution to the extended Riccati ODE to solve the NODE in (6) [36]:

$$P(\phi) = \sum_{m=0}^{p} m_j \left(\frac{\Upsilon'(\phi)}{\Upsilon(\phi)}\right)^j + \sum_{j=0}^{p-1} s_j \left(\frac{\Upsilon'(\phi)}{\Upsilon(\phi)}\right)^j \cdot \left(\frac{1}{\Upsilon(\phi)}\right). \tag{7}$$

Here, the solution to the resultant extended Riccati ODE is denoted by $\Upsilon(\phi)$ [36], and the unknown constants that need to be found later are represented by the parameters $m_j (j = 0, ..., p)$ and $s_j (j = 0, ..., p-1)$.

$$\Upsilon'(\phi) = J + K\Upsilon(\phi) + L(\Upsilon(\phi))^2, \tag{8}$$

where J, K and L are constants.

3. The positive integer p needed in Equation (7) is obtained by homogeneously balancing the highest-order derivative term and the largest nonlinear term in Equation (6).

4. Afterwards, all the terms with the same powers of $\Upsilon(\phi)$ are combined when (7) is substituted into (6) or in the equation that emerges from the integration of (6). This procedure results in an equation in terms of $\Upsilon(\phi)$. By setting the coefficients in this equation to zero, we generate an algebraic system of equations representing the parameters $m_j (j = 0, ..., p)$ and $s_j (j = 0, ..., p-1)$ along with the further associated parameters.

5. Using MAPLE, an analytical treatment of the resultant set of nonlinear algebraic equations is carried out.

6. Afterwards, analytical soliton solutions for (5) are produced by computing and entering the unknown values in Equation (7) together with $\Upsilon(\phi)$ (Equation (8) solution).

Several families of soliton solutions may be derived by using the general solution of (8). These are displayed in Table 1 [36].

Table 1. The particular solutions $\Upsilon(\phi)$ of the Riccati equation in (8) and the formation of $\left(\frac{\Upsilon'(\phi)}{\Upsilon(\phi)}\right)$, where $n_1, n_2 \in \mathrm{R}$, $\varpi = K^2 - 4LJ$ and $\zeta = \cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\sinh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)$.

S. No.	Family	Constraint(s)	$\Upsilon(\phi)$	$\left(\frac{\Upsilon'(\phi)}{\Upsilon(\phi)}\right)$
1	Trigonometric Solutions	$\varpi < 0, \quad L \neq 0$	$-\frac{K}{2L} + \frac{\sqrt{-\varpi}\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)}{2L}$,	$-\frac{\varpi\left(1+\left(\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)\right)^2\right)}{2(-K+\sqrt{-\varpi}\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right))}$,
			$-\frac{K}{2L} - \frac{\sqrt{-\varpi}\cot\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)}{2L}$,	$\frac{\left(1+\left(\cot\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)\right)^2\right)\varpi}{2(K+\sqrt{-\varpi}\cot\left(\frac{1}{2}\sqrt{-\varpi}\phi\right))}$,
			$-\frac{K}{2L} + \frac{\sqrt{-\varpi}\left(\tan\left(\sqrt{-\varpi}\phi\right)+\left(\sec\left(\sqrt{-\varpi}\phi\right)\right)\right)}{2L}$,	$\frac{-\varpi\left(1+\sin\left(\sqrt{-\varpi}\phi\right)\right)\sec\left(\sqrt{-\varpi}\phi\right)}{-K\cos\left(\sqrt{-\varpi}\phi\right)+\sqrt{-\varpi}\sin\left(\sqrt{-\varpi}\phi\right)+\sqrt{-\varpi}}$,
			$-\frac{K}{2L} + \frac{\sqrt{-\varpi}\left(\tan\left(\sqrt{-\varpi}\phi\right)-\left(\sec\left(\sqrt{-\varpi}\phi\right)\right)\right)}{2L}$.	$\frac{\varpi\left(\sin\left(\sqrt{-\varpi}\phi\right)-1\right)\sec\left(\sqrt{-\varpi}\phi\right)}{-K\cos\left(\sqrt{-\varpi}\phi\right)+\sqrt{-\varpi}\sin\left(\sqrt{-\varpi}\phi\right)-\sqrt{-\varpi}}$.
2	Hyperbolic Solutions	$\varpi > 0, \quad L \neq 0$	$-\frac{K}{2L} - \frac{\sqrt{\varpi}\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right)}{2L}$,	$-\frac{\left(-1+\left(\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right)\right)^2\right)\varpi}{2(K+\sqrt{\varpi}\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right))}$,
			$-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\tanh\left(\sqrt{\varpi}\phi\right)+i\left(\mathrm{sech}\left(\sqrt{\varpi}\phi\right)\right)\right)}{2L}$,	$\frac{-\varpi\left(-1+i\sinh\left(\sqrt{\varpi}\phi\right)\right)\mathrm{sech}\left(\sqrt{\varpi}\phi\right)}{K\cosh\left(\sqrt{\varpi}\phi\right)+\sqrt{\varpi}\sinh\left(\sqrt{\varpi}\phi\right)+i\sqrt{\varpi}}$,
			$-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\tanh\left(\sqrt{\varpi}\phi\right)-i\left(\mathrm{sech}\left(\sqrt{\varpi}\phi\right)\right)\right)}{2L}$,	$\frac{-\varpi\left(1+i\sinh\left(\sqrt{\varpi}\phi\right)\right)\mathrm{sech}\left(\sqrt{\varpi}\phi\right)}{-K\cosh\left(\sqrt{\varpi}\phi\right)-\sqrt{\varpi}\sinh\left(\sqrt{\varpi}\phi\right)+i\sqrt{\varpi}}$,
			$-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\coth\left(\sqrt{\varpi}\phi\right)+\left(\mathrm{csch}\left(\sqrt{\varpi}\phi\right)\right)\right)}{2L}$.	$-\frac{\varpi\left(2\left(\cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\right)^2-1\right)}{4\zeta\left(-2K\zeta+\sqrt{\varpi}\right)}$.
3	Rational Solutions	$\varpi = 0$	$\frac{-2J(K\phi+2)}{K^2\phi}$,	$\frac{-2}{\phi\,(K\phi+2)}$,
		$\varpi = 0$, & $K = L = 0$	$\phi\, J$,	$\frac{1}{\phi}$,
		$\varpi = 0$, & $K = J = 0$	$-\frac{1}{\phi\, L}$.	$-\frac{1}{\phi}$.
4	Exponential Solutions	$L = 0$, & $K = \epsilon$, $J = \kappa\epsilon$	$e^{\epsilon\,\phi} - \kappa$,	$\frac{\epsilon\, e^{\epsilon\,\phi}}{e^{\epsilon\,\phi}-\kappa}$,
		$J = 0$, & $K = \epsilon$, $L = \kappa\epsilon$	$\frac{e^{\epsilon\,\phi}}{1-\kappa e^{\epsilon\phi}}$.	$-\frac{\epsilon}{-1+\kappa e^{\epsilon\,\phi}}$.
5	Rational-Hyperbolic Solutions	$J = 0$, & $K \neq 0$, $L \neq 0$	$-\frac{K\,n_1}{L(\cosh(K\phi)-\sinh(K\phi)+n_2)}$,	$\frac{K(\sinh(K\phi)-\cosh(K\phi))}{-\cosh(K\phi)+\sinh(K\phi)-n_2}$,
			$-\frac{K(\cosh(K\phi)+\sinh(K\phi))}{L(\cosh(K\phi)+\sinh(K\phi)+n_2)}$.	$\frac{Kn_2}{\cosh(K\phi)+\sinh(K\phi)+n_2}$.

3. The Construction of Kink Soliton Solutions for WBBM

In this section, we apply our suggested approach, RMESEM, to acquire families of the soliton solutions for WBBM given in (4). We begin with the application of the transformation that follows:

$$Q(t,x,y,z) = P(\phi), \quad \text{where} \quad \phi = \lambda x + \mu y + \nu z - \omega t. \tag{9}$$

By appending (9) to it, (4) now becomes

$$(\lambda - \omega)P' + \mu(P^3)' + \nu\lambda\omega P''' = 0, \tag{10}$$

where $P' = \frac{dP}{d\phi}$. According to the homogeneous balancing principle, $p = 1$ is suggested by balancing the greatest derivative $\nu\lambda\omega P'''$ and the largest nonlinear term $\mu(P^3)'$. Putting $p = 1$ in (7) yields the following solution for (10):

$$P(\phi) = \sum_{m=0}^{1} m_j \left(\frac{\Upsilon'(\phi)}{\Upsilon(\phi)}\right)^j + s_0\left(\frac{1}{\Upsilon(\phi)}\right), \tag{11}$$

where the constants s_1, m_0, and m_1 are unknown. When we insert (11) into (10), we obtain an expression in $\Upsilon(\phi)$. An algebraic system of seven equations may be obtained by gathering all terms with the same powers of $\Upsilon(\phi)$ and setting the coefficient of each term to zero. We solve the system with Maple. And as a result, the system of algebraic equations yields the following five cases of results.
Case. 1

$$m_0 = \frac{s_0 K}{2J}, m_1 = 0, s_0 = s_0, \lambda = \lambda, \mu = \frac{-4J^2(-\omega+\lambda)}{\varpi\, {s_0}^2}, \nu = \frac{2(\lambda-\omega)}{\lambda\,\omega\,\varpi}, \omega = \omega. \tag{12}$$

Case. 2

$$m_0 = m_0, m_1 = m_1, s_0 = s_0, \lambda = \omega, \mu = 0, \nu = 0, \omega = \omega. \tag{13}$$

Case. 3

$$m_0 = -\frac{m_1 K}{2}, m_1 = m_1, s_0 = 0, \lambda = \lambda, \mu = \frac{-4(\lambda-\omega)}{(8\,JL+K^2){m_1}^2}, \nu = \frac{2(\lambda-\omega)}{\lambda\,\omega\,(8\,JL+K^2)}, \omega = \omega. \tag{14}$$

Case. 4

$$m_0 = -\frac{m_1 K}{2}, m_1 = m_1, s_0 = -m_1 J, \lambda = \lambda, \mu = \frac{-4(\lambda-\omega)}{\varpi\, {m_1}^2}, \nu = \frac{2(\lambda-\omega)}{\lambda\,\omega\,\varpi}, \omega = \omega. \tag{15}$$

Taking into consideration **Case. 1** and utilizing (9) and (11) with the corresponding solution of (8), we acquire the ensuing families of kink soliton solutions for (4):
Family. 1.1: When $\varpi < 0 \quad L \neq 0$,

$$Q_{1,1}(t,x,y,z) = \frac{s_0 K}{2J} + s_0\left(-\frac{K}{2L} + \frac{\sqrt{-\varpi}\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)}{2L}\right)^{-1}, \tag{16}$$

$$Q_{1,2}(t,x,y,z) = \frac{s_0 K}{2J} + s_0\left(-\frac{K}{2L} - \frac{\sqrt{-\varpi}\cot\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)}{2L}\right)^{-1}, \tag{17}$$

$$Q_{1,3}(t,x,y,z) = \frac{s_0 K}{2J} + s_0\left(-\frac{K}{2L} + \frac{\sqrt{-\varpi}\left(\tan\left(\sqrt{-\varpi}\phi\right) + \sec\left(\sqrt{-\varpi}\phi\right)\right)}{2L}\right)^{-1}, \tag{18}$$

and

$$Q_{1,4}(t,x,y,z) = \frac{s_0 K}{2J} + s_0\left(-\frac{K}{2L} + \frac{\sqrt{-\varpi}\left(\tan\left(\sqrt{-\varpi}\phi\right) - \sec\left(\sqrt{-\varpi}\phi\right)\right)}{2L}\right)^{-1}. \tag{19}$$

Family. 1.2: When $\varpi > 0 \quad L \neq 0$,

$$Q_{1,5}(t,x,y,z) = \frac{s_0 K}{2J} + s_0\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right)}{2L}\right)^{-1}, \tag{20}$$

$$Q_{1,6}(t,x,y,z) = \frac{s_0 K}{2J} + s_0\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\tanh\left(\sqrt{\varpi}\phi\right) + i\,\mathrm{sech}\left(\sqrt{\varpi}\phi\right)\right)}{2L}\right)^{-1}, \tag{21}$$

$$Q_{1,7}(t,x,y,z) = \frac{s_0 K}{2J} + s_0\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\tanh\left(\sqrt{\varpi}\phi\right) - i\,\mathrm{sech}\left(\sqrt{\varpi}\phi\right)\right)}{2L}\right)^{-1}, \tag{22}$$

and

$$Q_{1,8}(t,x,y,z) = \frac{s_0 K}{2J} + s_0\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\tanh\left(\frac{1}{4}\sqrt{\varpi}\phi\right) - \coth\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\right)}{4L}\right)^{-1}. \tag{23}$$

Family. 1.3: When $K = \epsilon$, $J = \kappa\epsilon(\kappa \neq 0)$ and $L = 0$,

$$Q_{1,9}(t,x,y,z) = \frac{s_0}{2\kappa} + \frac{s_0}{e^{\epsilon\phi} - \kappa}. \tag{24}$$

In the above kink soliton solutions, $\phi = \lambda x + (\frac{J^2(-4(\lambda-\omega))}{\varpi {s_0}^2})y + (\frac{2(\lambda-\omega)}{\lambda\omega\varpi})z - \omega t$.

Taking into consideration **Case. 2** and utilizing (9) and (11) with the corresponding solution of (8), we acquire the ensuing families of kink soliton solutions for (4):
Family. 2.1: When $\varpi < 0 \quad L \neq 0$,

$$Q_{2,1}(t,x,y,z) = m_0 + \frac{m_1\varpi\left(1 + \left(\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)\right)^2\right)}{2(K - \sqrt{-\varpi}\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right))} + s_0\left(-\frac{K}{2L} + \frac{\sqrt{-\varpi}\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)}{2L}\right)^{-1}, \tag{25}$$

$$Q_{2,2}(t,x,y,z) = m_0 + \frac{m_1\varpi\left(1 + \left(\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)\right)^2\right)}{2(K - \sqrt{-\varpi}\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right))} + s_0\left(-\frac{K}{2L} + \frac{\sqrt{-\varpi}\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)}{2L}\right)^{-1}, \tag{26}$$

$$\begin{aligned} Q_{2,3}(t,x,y,z) =& \frac{m_1\varpi\left(1 + \sin\left(\sqrt{-\varpi}\phi\right)\right)}{\cos\left(\sqrt{-\varpi}\phi\right)\left(K\cos\left(\sqrt{-\varpi}\phi\right) - \sqrt{-\varpi}\sin\left(\sqrt{-\varpi}\phi\right) - \sqrt{-\varpi}\right)} \\ &+ s_0\left(-\frac{K}{2L} + \frac{\sqrt{-\varpi}\left(\tan\left(\sqrt{-\varpi}\phi\right) + \sec\left(\sqrt{-\varpi}\phi\right)\right)}{2L}\right)^{-1} + m_0, \end{aligned} \tag{27}$$

and

$$\begin{aligned} Q_{2,4}(t,x,y,z) =& -\frac{m_1\varpi\left(\sin\left(\sqrt{-\varpi}\phi\right) - 1\right)}{\cos\left(\sqrt{-\varpi}\phi\right)\left(K\cos\left(\sqrt{-\varpi}\phi\right) - \sqrt{-\varpi}\sin\left(\sqrt{-\varpi}\phi\right) + \sqrt{-\varpi}\right)} \\ &+ s_0\left(-\frac{K}{2L} + \frac{\sqrt{-\varpi}\left(\tan\left(\sqrt{-\varpi}\phi\right) - \sec\left(\sqrt{-\varpi}\phi\right)\right)}{2L}\right)^{-1} + m_0. \end{aligned} \tag{28}$$

Family. 2.2: When $\varpi > 0 \quad L \neq 0$,

$$Q_{2,5}(t,x,y,z) = m_0 - \frac{1}{2}\frac{m_1\varpi\left(-1+\left(\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right)\right)^2\right)}{K+\sqrt{\varpi}\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right)} + s_0\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right)}{2L}\right)^{-1}, \tag{29}$$

$$\begin{aligned} Q_{2,6}(t,x,y,z) =& -\frac{m_1\varpi\left(-1+i\sinh\left(\sqrt{\varpi}\phi\right)\right)}{\cosh\left(\sqrt{\varpi}\phi\right)\left(K\cosh\left(\sqrt{\varpi}\phi\right)+\sqrt{\varpi}\sinh\left(\sqrt{\varpi}\phi\right)+i\sqrt{\varpi}\right)} \\ &+ s_0\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\tanh\left(\sqrt{\varpi}\phi\right)+i\,\mathrm{sech}\left(\sqrt{\varpi}\phi\right)\right)}{2L}\right)^{-1} + m_0, \end{aligned} \tag{30}$$

$$\begin{aligned} Q_{2,7}(t,x,y,z) =& \frac{m_1\varpi\left(1+i\sinh\left(\sqrt{\varpi}\phi\right)\right)}{\cosh\left(\sqrt{\varpi}\phi\right)\left(K\cosh\left(\sqrt{\varpi}\phi\right)+\sqrt{\varpi}\sinh\left(\sqrt{\varpi}\phi\right)-i\sqrt{\varpi}\right)} \\ &+ s_0\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\tanh\left(\sqrt{\varpi}\phi\right)-i\,\mathrm{sech}\left(\sqrt{\varpi}\phi\right)\right)}{2L}\right)^{-1} + m_0, \end{aligned} \tag{31}$$

and

$$\begin{aligned} Q_{2,8}(t,x,y,z) =& \frac{1}{4}\frac{m_1\varpi\left(2\left(\cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\right)^2-1\right)}{\cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\sinh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\left(2K\cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\sinh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)-\sqrt{\varpi}\right)} \\ &+ s_0\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\tanh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)-\coth\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\right)}{4L}\right)^{-1} + m_0. \end{aligned} \tag{32}$$

Family. 2.3: When $\varpi = 0, \quad K \neq 0$,

$$Q_{2,9}(t,x,y,z) = m_0 - \frac{2m_1}{\phi\left(K\phi+2\right)} - \frac{s_0K^2\phi}{2J(K\phi+2)}. \tag{33}$$

Family. 2.4: When $\varpi = 0$, in case when $K = L = 0$,

$$Q_{2,10}(t,x,y,z) = m_0 + \frac{m_1}{\phi} + \frac{s_0}{J\phi}. \tag{34}$$

Family. 2.5: When $\varpi = 0$, in case when $K = J = 0$,

$$Q_{2,11}(t,x,y,z) = m_0 - \frac{m_1}{\phi} - s_0L\phi. \tag{35}$$

Family. 2.6: When $K = \epsilon$, $J = \kappa\epsilon(\kappa \neq 0)$ and $L = 0$,

$$Q_{2,12}(t,x,y,z) = m_0 + \frac{m_1\epsilon\,\mathrm{e}^{\epsilon\phi}}{\mathrm{e}^{\epsilon\phi}-\kappa} + \frac{s_0}{\mathrm{e}^{\epsilon\phi}-\kappa}. \tag{36}$$

Family. 2.7: When $K = \epsilon$, $L = \kappa\epsilon(\kappa \neq 0)$ and $J = 0$,

$$Q_{2,13}(t,x,y,z) = m_0 - \frac{m_1\epsilon}{-1+\kappa\,\mathrm{e}^{\epsilon\phi}} + \frac{s_0\left(1-\kappa\,\mathrm{e}^{\epsilon\phi}\right)}{\mathrm{e}^{\epsilon\phi}}. \tag{37}$$

Family. 2.8: When $J = 0$, $L \neq 0$ and $K \neq 0$,

$$Q_{2,14}(t,x,y,z) = m_0 + \frac{m_1 K(\sinh(K\phi) - \cosh(K\phi))}{-\cosh(K\phi) + \sinh(K\phi) - n_2} - \frac{s_0 L(\cosh(K\phi) - \sinh(K\phi) + n_2)}{n_1 K}, \tag{38}$$

and

$$Q_{2,15}(t,x,y,z) = m_0 + \frac{m_1 K n_2}{\cosh(K\phi) + \sinh(K\phi) + n_2} - \frac{s_0 L(\cosh(K\phi) + \sinh(K\phi) + n_2)}{K(\cosh(K\phi) + \sinh(K\phi))}. \tag{39}$$

In the above kink soliton solutions, $\phi = \lambda(x - t)$.

Taking into consideration **Case. 3** and utilizing (9) and (11) with the corresponding solution of (8), we acquire the ensuing families of kink soliton solutions for (4):
Family. 3.1: When $\varpi < 0 \quad L \neq 0$,

$$Q_{3,1}(t,x,y,z) = -\frac{m_1 K}{2} + \frac{m_1 \varpi \left(1 + \left(\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)\right)^2\right)}{2(K - \sqrt{-\varpi}\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right))}, \tag{40}$$

$$Q_{3,2}(t,x,y,z) = -\frac{m_1 K}{2} + \frac{m_1 \varpi \left(1 + \left(\cot\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)\right)^2\right)}{2(K + \sqrt{-\varpi}\cot\left(\frac{1}{2}\sqrt{-\varpi}\phi\right))}, \tag{41}$$

$$Q_{3,3}(t,x,y,z) = -\frac{m_1 K}{2} + \frac{m_1 \varpi \left(1 + \sin\left(\sqrt{-\varpi}\phi\right)\right)}{\cos\left(\sqrt{-\varpi}\phi\right)\left(K\cos\left(\sqrt{-\varpi}\phi\right) - \sqrt{-\varpi}\sin\left(\sqrt{-\varpi}\phi\right) - \sqrt{-\varpi}\right)}, \tag{42}$$

and

$$Q_{3,4}(t,x,y,z) = -\frac{m_1 K}{2} - \frac{m_1 \varpi \left(\sin\left(\sqrt{-\varpi}\phi\right) - 1\right)}{\cos\left(\sqrt{-\varpi}\phi\right)\left(K\cos\left(\sqrt{-\varpi}\phi\right) - \sqrt{-\varpi}\sin\left(\sqrt{-\varpi}\phi\right) + \sqrt{-\varpi}\right)}. \tag{43}$$

Family. 3.2: When $\varpi > 0 \quad L \neq 0$,

$$Q_{3,5}(t,x,y,z) = -\frac{m_1 K}{2} - \frac{m_1 \varpi \left(-1 + \left(\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right)\right)^2\right)}{2(K + \sqrt{\varpi}\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right))}, \tag{44}$$

$$Q_{3,6}(t,x,y,z) = -\frac{m_1 K}{2} - \frac{m_1 \varpi \left(-1 + i\sinh\left(\sqrt{\varpi}\phi\right)\right)}{\cosh\left(\sqrt{\varpi}\phi\right)\left(K\cosh\left(\sqrt{\varpi}\phi\right) + \sqrt{\varpi}\sinh\left(\sqrt{\varpi}\phi\right) + i\sqrt{\varpi}\right)}, \tag{45}$$

$$Q_{3,7}(t,x,y,z) = -\frac{m_1 K}{2} + \frac{m_1 \varpi \left(1 + i\sinh\left(\sqrt{\varpi}\phi\right)\right)}{\cosh\left(\sqrt{\varpi}\phi\right)\left(K\cosh\left(\sqrt{\varpi}\phi\right) + \sqrt{\varpi}\sinh\left(\sqrt{\varpi}\phi\right) - i\sqrt{\varpi}\right)}, \tag{46}$$

and

$$Q_{3,8}(t,x,y,z) = -\frac{m_1 K}{2} + \frac{m_1 \varpi \left(2\left(\cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\right)^2 - 1\right)}{4(\cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\sinh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\left(2K\cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\sinh\left(\frac{1}{4}\sqrt{\varpi}\phi\right) - \sqrt{\varpi}\right))}. \tag{47}$$

Family. 3.3: When $\varpi = 0, \quad K \neq 0$,

$$Q_{3,9}(t,x,y,z) = -\frac{m_1 K}{2} - 2\frac{m_1}{\phi\,(K\phi + 2)}. \tag{48}$$

Family. 3.4: When $\varpi = 0$, in case when $K = L = 0$,

$$Q_{3,10}(t,x,y,z) = \frac{m_1}{\phi}. \tag{49}$$

Family. 3.5: When $\varpi = 0$, in case when $K = J = 0$,

$$Q_{3,11}(t,x,y,z) = -\frac{m_1}{\phi}. \tag{50}$$

Family. 3.6: When $K = \epsilon$, $J = \kappa\epsilon(\kappa \neq 0)$ and $L = 0$,

$$Q_{3,12}(t,x,y,z) = -\frac{m_1\epsilon}{2} + \frac{m_1\epsilon\, e^{\epsilon\phi}}{e^{\epsilon\phi} - \kappa}. \tag{51}$$

Family. 3.7: When $K = \epsilon$, $L = \kappa\epsilon(\kappa \neq 0)$ and $J = 0$,

$$Q_{3,13}(t,x,y,z) = -\frac{m_1\epsilon}{2} - \frac{m_1\epsilon}{-1 + \kappa\, e^{\epsilon\phi}}. \tag{52}$$

Family. 3.8: When $J = 0$, $L \neq 0$ and $K \neq 0$,

$$Q_{3,14}(t,x,y,z) = -\frac{m_1 K}{2} + \frac{m_1 K(\sinh(K\phi) - \cosh(K\phi))}{-\cosh(K\phi) + \sinh(K\phi) - n_2}, \tag{53}$$

and

$$Q_{3,15}(t,x,y,z) = -\frac{m_1 K}{2} + \frac{m_1 K n_2}{\cosh(K\phi) + \sinh(K\phi) + n_2}. \tag{54}$$

In the above kink soliton solutions, $\phi = \lambda x + (\frac{-4(\lambda-\omega)}{(8JL+K^2){m_1}^2})y + (\frac{2(\lambda-\omega)}{\lambda\,\omega\,(8JL+K^2)})z - \omega t$.

Taking into consideration **Case. 4** and utilizing (9) and (11) with the corresponding solution of (8), we acquire the ensuing families of kink soliton solutions for (4):
Family. 4.1: When $\varpi < 0 \quad L \neq 0$,

$$Q_{4,1}(t,x,y,z) = -\frac{m_1 K}{2} + \frac{m_1 \varpi \left(1 + \left(\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)\right)^2\right)}{2(K - \sqrt{-\varpi}\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right))} - m_1 J\left(-\frac{K}{2L} + \frac{\sqrt{-\varpi}\tan\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)}{2L}\right)^{-1}, \tag{55}$$

$$Q_{4,2}(t,x,y,z) = -\frac{m_1 K}{2} + \frac{m_1\varpi\left(1+\left(\cot\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)\right)^2\right)}{2(K+\sqrt{-\varpi}\cot\left(\frac{1}{2}\sqrt{-\varpi}\phi\right))} - m_1 J\left(-\frac{K}{2L} - \frac{\sqrt{-\varpi}\cot\left(\frac{1}{2}\sqrt{-\varpi}\phi\right)}{2L}\right)^{-1}, \tag{56}$$

$$Q_{4,3}(t,x,y,z) = \frac{m_1\varpi\left(1+\sin\left(\sqrt{-\varpi}\phi\right)\right)}{\cos\left(\sqrt{-\varpi}\phi\right)\left(K\cos\left(\sqrt{-\varpi}\phi\right)-\sqrt{-\varpi}\sin\left(\sqrt{-\varpi}\phi\right)-\sqrt{-\varpi}\right)} - m_1 J\left(-\frac{K}{2L} + \frac{\sqrt{-\varpi}\left(\tan\left(\sqrt{-\varpi}\phi\right)+\sec\left(\sqrt{-\varpi}\phi\right)\right)}{2L}\right)^{-1} - \frac{m_1 K}{2}, \tag{57}$$

and

$$Q_{4,4}(t,x,y,z) = -\frac{m_1\varpi\left(\sin\left(\sqrt{-\varpi}\phi\right)-1\right)}{\cos\left(\sqrt{-\varpi}\phi\right)\left(K\cos\left(\sqrt{-\varpi}\phi\right)-\sqrt{-\varpi}\sin\left(\sqrt{-\varpi}\phi\right)+\sqrt{-\varpi}\right)} - m_1 J\left(-\frac{K}{2L} + \frac{\sqrt{-\varpi}\left(\tan\left(\sqrt{-\varpi}\phi\right)-\sec\left(\sqrt{-\varpi}\phi\right)\right)}{2L}\right)^{-1} - \frac{m_1 K}{2}. \tag{58}$$

Family. 4.2: When $\varpi > 0 \quad L \neq 0$,

$$Q_{4,5}(t,x,y,z) = -\frac{m_1 K}{2} - \frac{m_1\varpi\left(-1+\left(\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right)\right)^2\right)}{2(K+\sqrt{\varpi}\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right))} - m_1 J\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\tanh\left(\frac{1}{2}\sqrt{\varpi}\phi\right)}{2L}\right)^{-1}, \tag{59}$$

$$Q_{4,6}(t,x,y,z) = -\frac{m_1\varpi\left(-1+i\sinh\left(\sqrt{\varpi}\phi\right)\right)}{\cosh\left(\sqrt{\varpi}\phi\right)\left(K\cosh\left(\sqrt{\varpi}\phi\right)+\sqrt{\varpi}\sinh\left(\sqrt{\varpi}\phi\right)+i\sqrt{\varpi}\right)} - m_1 J\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\tanh\left(\sqrt{\varpi}\phi\right)+i\mathrm{sech}\left(\sqrt{\varpi}\phi\right)\right)}{2L}\right)^{-1} - \frac{m_1 K}{2}, \tag{60}$$

$$Q_{4,7}(t,x,y,z) = \frac{m_1\varpi\left(1+i\sinh\left(\sqrt{\varpi}\phi\right)\right)}{\cosh\left(\sqrt{\varpi}\phi\right)\left(K\cosh\left(\sqrt{\varpi}\phi\right)+\sqrt{\varpi}\sinh\left(\sqrt{\varpi}\phi\right)-i\sqrt{\varpi}\right)} - m_1 J\left(-\frac{K}{2L} - \frac{\sqrt{\varpi}\left(\tanh\left(\sqrt{\varpi}\phi\right)-i\mathrm{sech}\left(\sqrt{\varpi}\phi\right)\right)}{2L}\right)^{-1} - \frac{m_1 K}{2}, \tag{61}$$

and

$$Q_{4,8}(t,x,y,z) = \frac{m_1\varpi\left(2\left(\cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\right)^2-1\right)}{4(\cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\sinh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\left(2K\cosh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\sinh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)-\sqrt{\varpi}\right))} - m_1 J\left(-\frac{K}{2L}-\frac{\sqrt{\varpi}\left(\tanh\left(\frac{1}{4}\sqrt{\varpi}\phi\right)-\coth\left(\frac{1}{4}\sqrt{\varpi}\phi\right)\right)}{4L}\right)^{-1} - \frac{m_1K}{2}. \tag{62}$$

Family. 4.3: When $K=\epsilon$, $J=\kappa\epsilon(\kappa\neq 0)$ and $L=0$,

$$Q_{4,9}(t,x,y,z) = -\frac{m_1\epsilon}{2} + \frac{m_1\epsilon\, e^{\epsilon\phi}}{e^{\epsilon\phi}-\kappa} - \frac{m_1\kappa\epsilon}{e^{\epsilon\phi}-\kappa}. \tag{63}$$

Family. 4.4: When $K=\epsilon$, $L=\kappa\epsilon(\kappa\neq 0)$ and $J=0$,

$$Q_{4,10}(t,x,y,z) = -\frac{m_1\epsilon}{2} - \frac{m_1\epsilon}{-1+\kappa\, e^{\epsilon\phi}}. \tag{64}$$

Family. 4.5: When $J=0$, $L\neq 0$ and $K\neq 0$,

$$Q_{4,11}(t,x,y,z) = -\frac{m_1K}{2} + \frac{m_1K(\sinh(K\phi)-\cosh(K\phi))}{-\cosh(K\phi)+\sinh(K\phi)-n_2}, \tag{65}$$

and

$$Q_{4,12}(t,x,y,z) = -\frac{m_1K}{2} + \frac{m_1Kn_2}{\cosh(K\phi)+\sinh(K\phi)+n_2}. \tag{66}$$

In the above kink soliton solutions, $\phi = \lambda x + (\frac{-4(\lambda-\omega)}{\varpi\, {m_1}^2})y + (\frac{2(\lambda-\omega)}{\lambda\,\omega\,\varpi})z - \omega t$.

4. Results and Discussion

In this section, numerous traveling soliton structures discovered in the model under investigation are visually displayed. In the framework of WBBM, we use the innovative RMESEM to extract and depict a number of wave structures in 3D, 2D, and contour forms, exposing the presence of multiple kink solitons, including anti-kink, twinning kink, bright kink, bifurcated kink, lump-like kink, and other multiple kinks. Traveling wave solutions such as soliton types are crucial for understanding the dynamic behavior of many physical systems modeled by NPDEs. Self-reinforcing solitary waves, or soliton waves, are able to maintain their form during propagation. Their localized character and stability set them apart.

Prominently, we obtain kink solitons such as solitary kink, anti-kink, twinning kink, bright kink, bifurcated kink, lump-like kink and other multiple kinks in the realm of WBBM. A single, localized wave that keeps its form while moving at a steady pace is called a solitary kink. It symbolizes a single, steady crest or depression that travels across the water surface without altering shape in water ripples. A reversed version of an isolated kink, an anti-kink symbolizes a falling wave in a situation where a solitary kink represents a rising wave. It shows a depression or dip that moves steadily across the surface in the case of water surges. When two kinks move in tandem, they form a twinning kink, which frequently takes the form of two linked waves. This is a representation of two parallel troughs and crests moving across the surface of the water. A localized wave that is larger in frequency than the medium it surrounds is called a bright kink. It denotes a prominent, bright crest that moves with noticeable movement in surges of water. As it spreads, a bifurcated kink divides into multiple tendrils. When it comes to water crests, it is a situation in which an individual wave peak or depression splits, resulting in many wave fronts or pathways. A

localized wave which looks a lump and bump, frequently in a three-dimensional in nature situation, is called a lump-like kink. This is an additional curved localized rise or dip in the waves of water as opposed to the crispness of a single kink. Finally, a sequence of kink oscillations engaging or passing through a medium is referred to as many kinks. When it comes to water gestures, they symbolize a complicated arrangement of several cresting and crashing points that move in tandem to form a complex waveform formation. Each of these kinds of waves demonstrates the many kinetics of the water propagation of waves and shows how the WBBM model may be used to predict and comprehend a wide range of complicated and diverse wave behaviors. Moreover, the graph for $Q_{1,5}$ in (20) displays the profile of the solitary kink soliton as seen in Figure 1. The graph for $Q_{1,9}$ in (24) is shown in Figure 2, which displays a bright kink soliton profile. Similarly, the graph for $Q_{2,5}$ given in (29) is displayed in Figure 3, which shows the anti-kink soliton profile. The graph for $Q_{2,11}$ given in (35) is displayed in Figure 4, which exhibits the bifurcated kink soliton profile. Figure 5 represents the graph for $Q_{2,14}$ shown in (38), which displays the soliton profile of a solitary kink. Likewise, Figure 6 shows the graph for $Q_{3,3}$ reported in (42), which displays a lump-like kink soliton profile. The graph for $Q_{3,8}$, which is a bright kink soliton profile, is shown in Figure 7. The graph of the twinning kink soliton profile is shown in Figure 8 for $Q_{3,9}$ as reported in (48). Figure 9 is plotted for $Q_{4,5}$ articulated in (59), which shows a solitary kink soliton profile. Finally, the graph for $Q_{4,10}$ given in (64) displays the kink soliton profile in Figure 10.

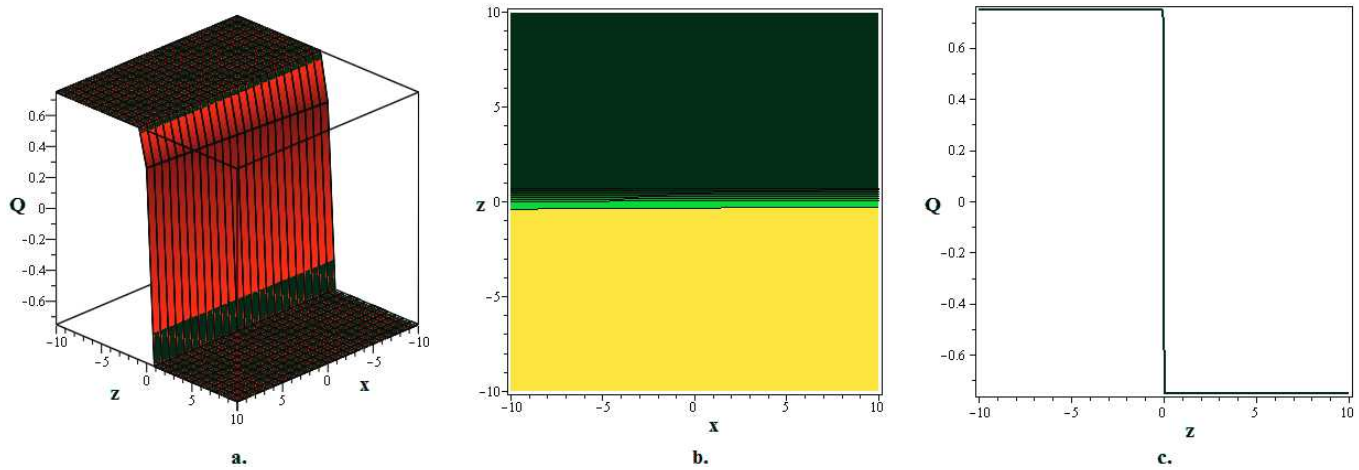


Figure 1. (**a**–**c**) The 3D, contour, and 2D (when $x = 1$) graphs for the solitary kink soliton solution $Q_{1,5}$ stated in (20) are plotted for $J = 2, K = 5, L = 2, \lambda = 0.005, \omega = 0.025, s_0 = 1, t = 0, y = 1$.

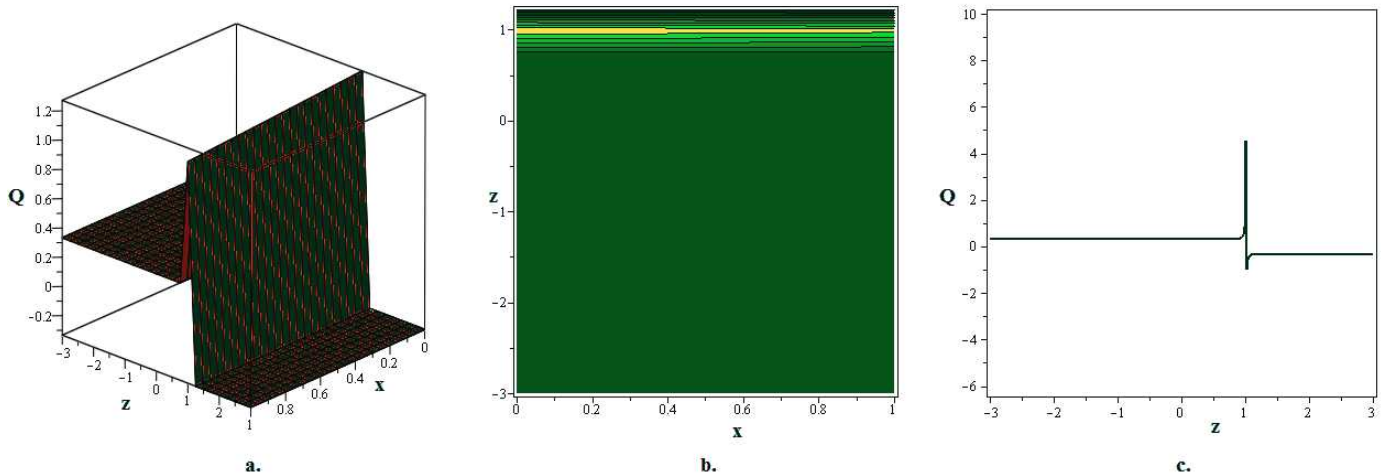


Figure 2. (**a**–**c**) The 3D, contour, and 2D (when $x = 2$) graphs for the bright kink soliton solution $Q_{1,9}$ stated in (24) are plotted for $J = 9, K = 3, L = 0, \lambda = 0.015, \omega = 0.35, s_0 = 2, t = 1, y = 5, \kappa = 3, \epsilon = 3$.

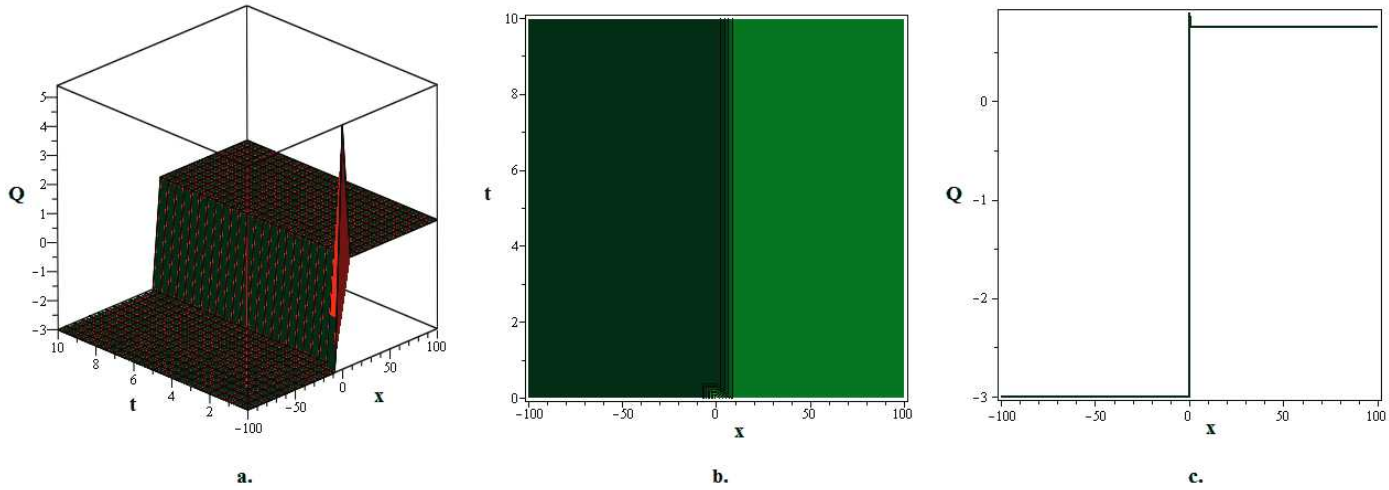


Figure 3. (**a**–**c**) The 3D, contour, and 2D (when $t = 1$) graphs for the anti-kink soliton solution $Q_{2,5}$ stated in (29) are plotted for $J = 8, K = 10, L = 2, \lambda = 50, \omega = 15, s_0 = 5, m_0 = 2, m_1 = 3$.

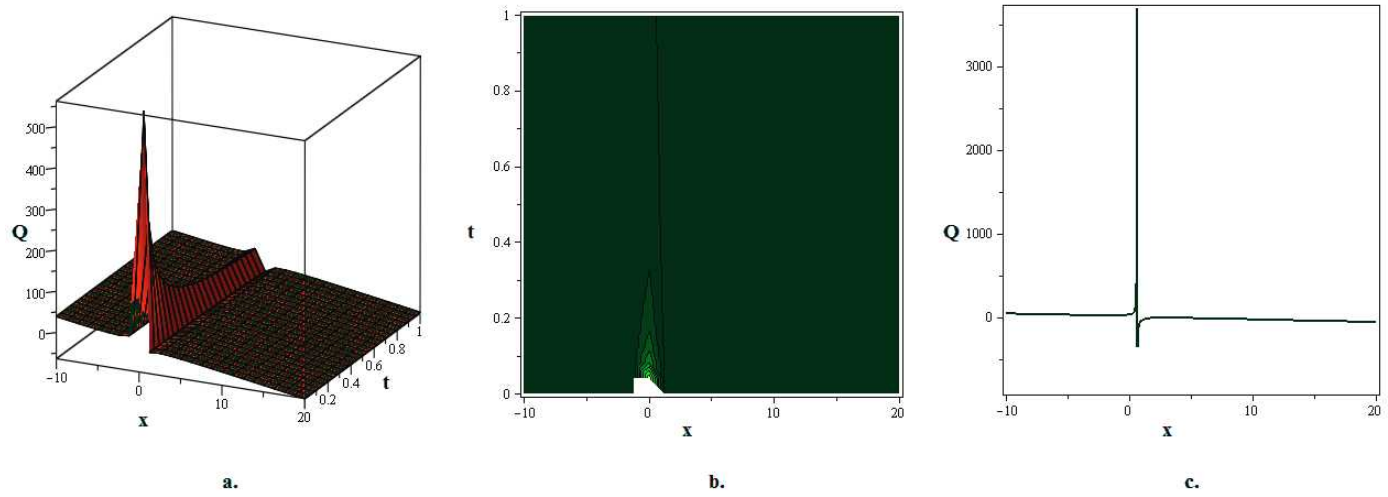


Figure 4. (**a**–**c**) The 3D, contour, and 2D (when $t = 1$) graphs for the bifurcated kink soliton solution $Q_{2,11}$ stated in (35) are plotted for $J = 0, K = 0, L = 1, \lambda = 0.335, \omega = 0.215, s_0 = 10, m_0 = 4, m_1 = 5$. In this profile, the kink interacts with shock wave.

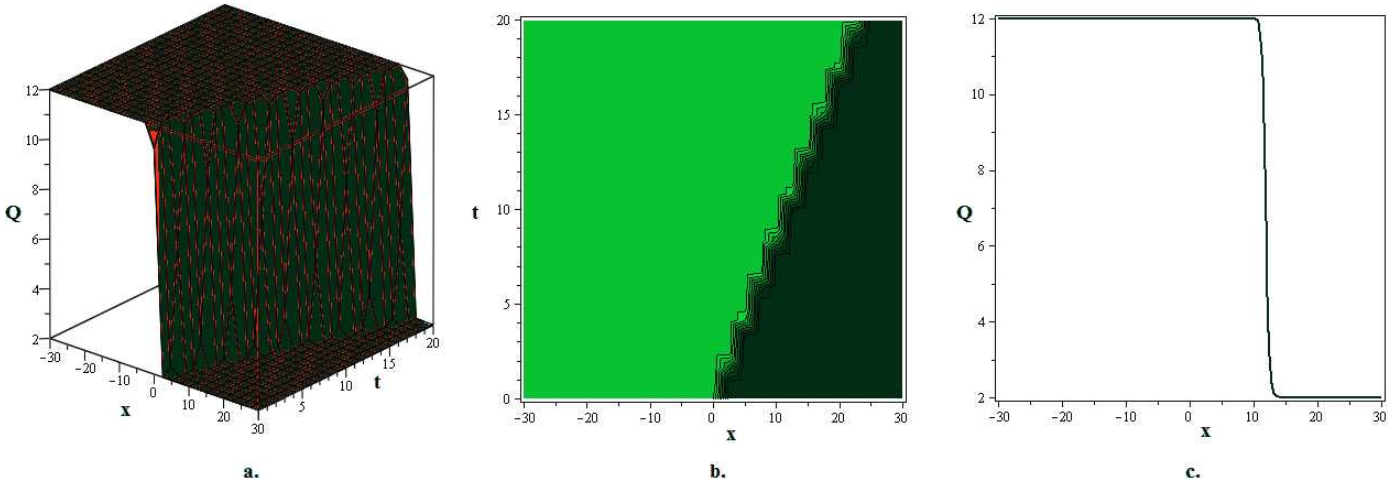


Figure 5. (**a**–**c**) The 3D, contour, and 2D (when $t = 10$) graphs for the solitary kink soliton solution $Q_{2,14}$ stated in (38) are plotted for $J = 0, K = 5, L = 0, \lambda = 0.75, \omega = 0.85, s_0 = 15, m_0 = 2, m_1 = 2, n_1 = 0.5, n_2 = 0.10$.

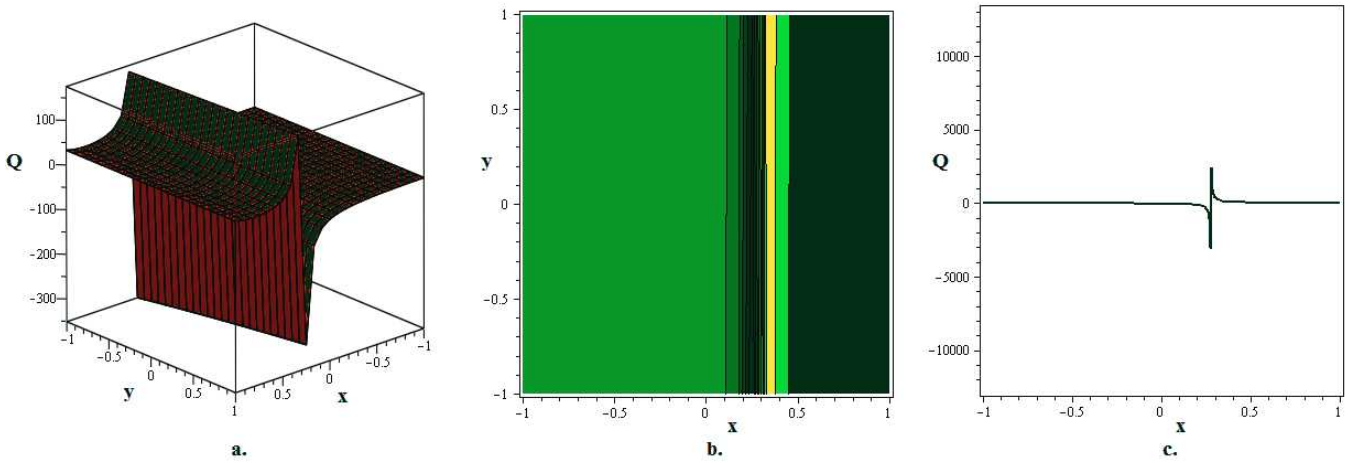


Figure 6. (**a**–**c**) The 3D, contour, and 2D (when $y = 0$) graphs for the lump-type kink soliton solution $Q_{3,3}$ stated in (42) are plotted for $J = 1, K = \sqrt{(}7), L = 4, \lambda = 0.545, \omega = 0.235, m_1 = 5, t = 10, z = 0.5$.

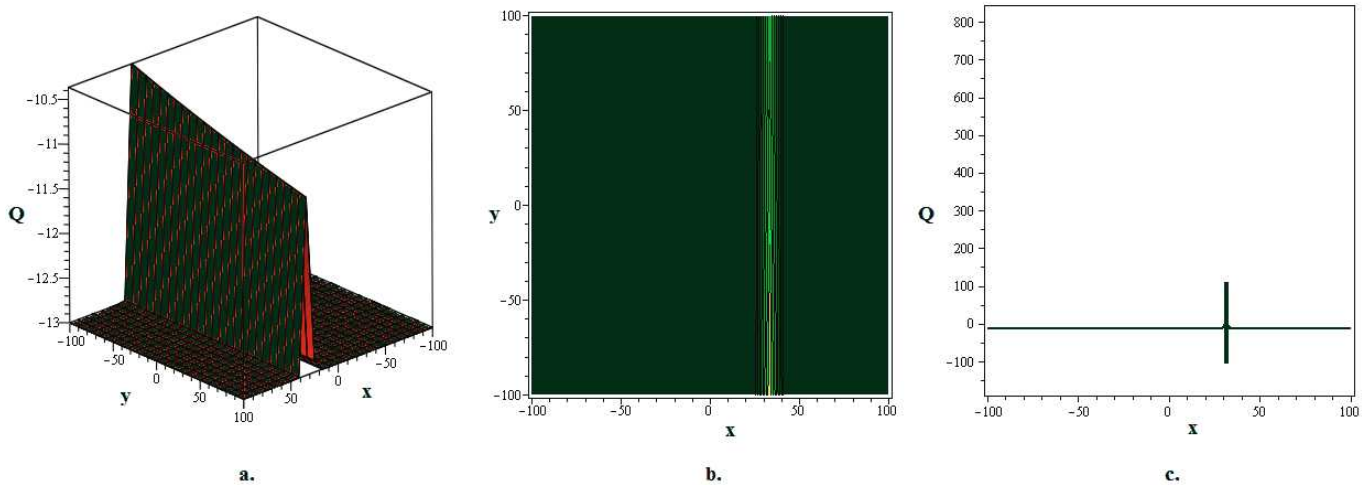


Figure 7. (**a**–**c**) The 3D, contour, and 2D (when $y = 20$) graphs for the bright kink soliton solution $Q_{3,8}$ stated in (47) are plotted for $J = 3, K = 13, L = 12, \lambda = 0.221, \omega = 0.345, m_1 = 2, t = 20, z = 1$.

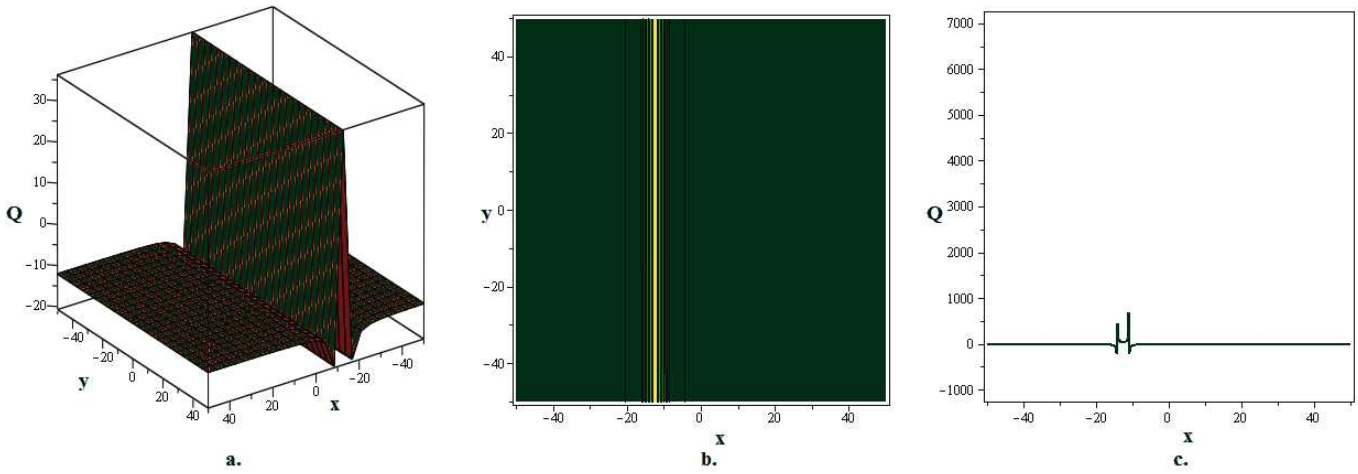


Figure 8. (**a**–**c**) The 3D, contour, and 2D (when $y = 0$) graphs for the twinning kink soliton solution $Q_{3,9}$ stated in (48) are plotted for $J = 1, K = 2, L = 1, \lambda = 0.315, \omega = 0.115, m_1 = 12, t = 10, z = 5$.

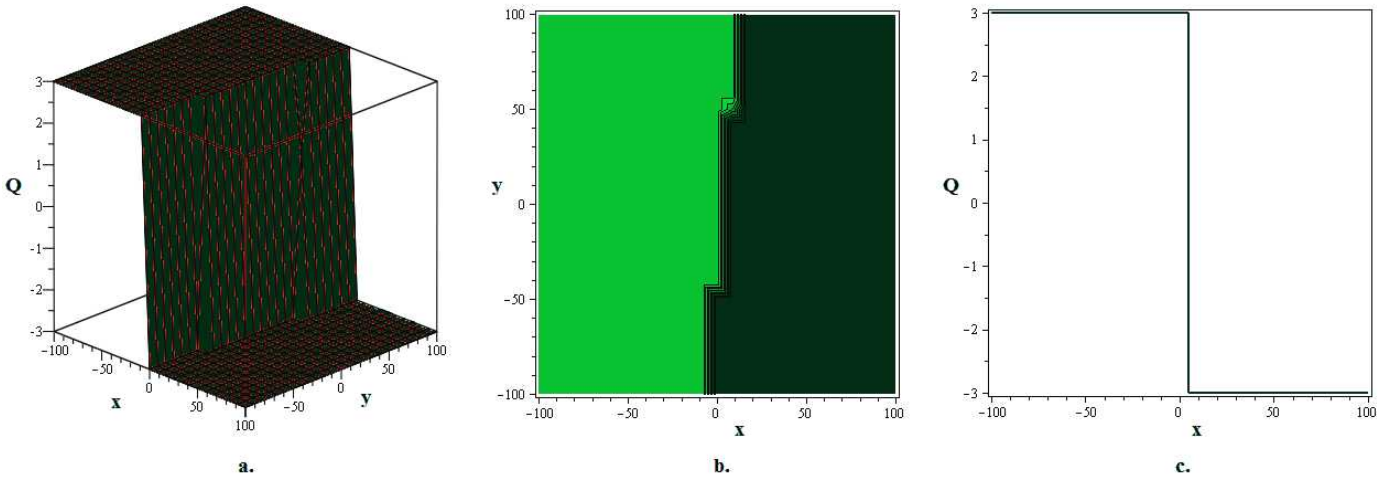


Figure 9. (**a**–**c**) The 3D, contour, and 2D (when $y = 10$) graphs for the multiple kink soliton solution $Q_{4,5}$ stated in (59) are plotted for $J = 4, K = 10, L = 4, \lambda = 5, \omega = 1, m_1 = 1, t = 20, z = 10$.

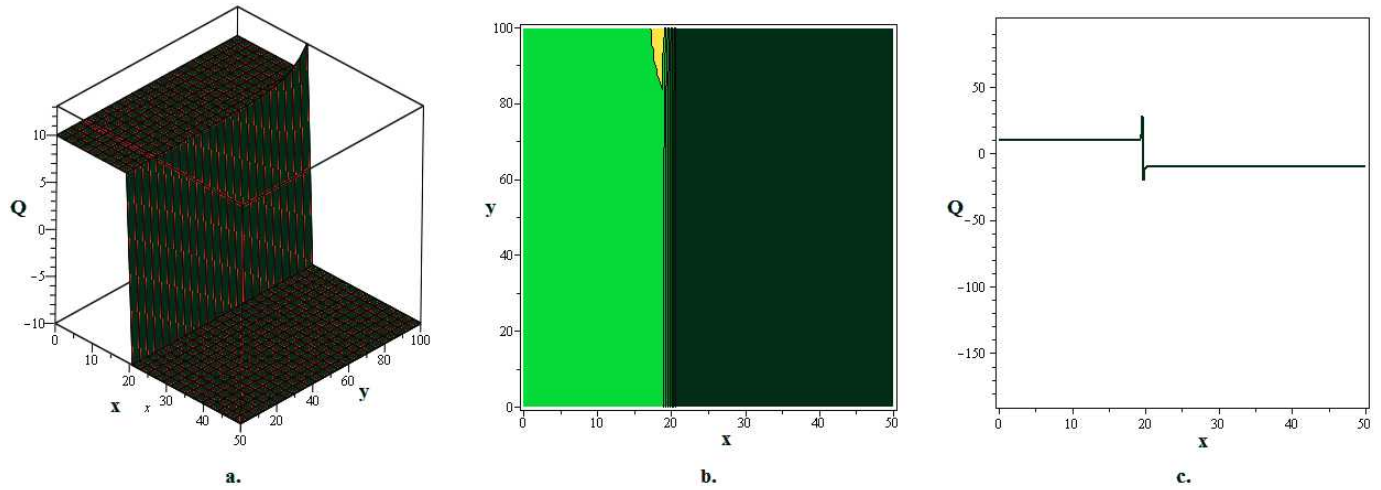


Figure 10. (**a**–**c**) The 3D, contour, and 2D (when $y = 30$) graphs for the kink soliton solution $Q_{4,10}$ stated in (64) are plotted for $J = 0, K = 10, L = 20, \lambda = 1, \omega = 2, m_1 = 2, t = 10, z = 2, \epsilon = 10, \kappa = 2$.

5. Conclusions

In conclusion, an innovative technique known as RMESEM is utilized in this study to produce and evaluate precise families of soliton solutions for the (3+1)-dimensional WBBM. The propagating behavior of acquired solitons is graphically represented by the 3D, 2D, and contour graphics, which clearly demonstrated the existence of several types of kink solitons in the setting of WBBM, including anti-kink, twinning kink, bright kink, bifurcated kink, lump-like kink, and other multiple kinks. The obtained soliton solutions provide insightful information on the WBBM model behavior. Moreover, it is demonstrated that solving NPDEs with the RMESEM approach is a flexible and effective strategy. Our understanding of the underlying physical processes and the relationships between different wave phenomena in a nonlinear setting are advanced by its ability to create a range of soliton families. Understanding the stability, interactions, and soliton dispersion aspects of the WBBM model is further made easier with the help of the derived soliton solutions. Nonetheless, the approach fails when the highest derivative term and nonlinear terms are not homogeneously balanced. Notwithstanding this limitation, the present investigation demonstrates that the methodology employed in this work is highly portable and efficacious for nonlinear problems in a variety of scientific domains. Moreover, further exploration into the stability analysis of solitons as well as the inclusion and effects of

fractional derivatives on solitons within the context of targeted model is the future objective of this study.

Author Contributions: Conceptualization, M.Y.A.; Methodology, M.Y.A.; Validation, H.A.; Data curation, H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Acknowledgments: The authors gratefully acknowledge the funding of the Deanship of Graduate Studies and Scientific Research, Jazan University, Saudi Arabia, through project number: RG24-L02.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1. Comparison with Other Analytical Methods

The findings of the other analytical procedures can be derived from our method. For example, setting $m_1 = 0$ in (11) results in

$$P(\phi) = s_0\left(\frac{1}{\Upsilon(\phi)}\right). \tag{A1}$$

This displays the EDAM closed-form solution's structure [29]. Thus, by setting $m_1 = 0$ from our findings, the results obtained by EDAM may also be achieved. For example, if we put $m_0 = a_0$, $s_0 = b_1$, $J = \Omega$, $K = 0$, and $L = 1$ as shown in (8), we receive some of the findings from our results gained under **Case. 1** that Shafqat-Ur-Rahman [37] achieved. In a similar manner, we obtain (11) if we set $s_0 = 0$:

$$P(\phi) = \sum_{m=0}^{1} m_j\left(\frac{\Upsilon'(\phi)}{\Upsilon(\phi)}\right)^j, \tag{A2}$$

which, using the Riccati-NODE, is the closed-form solution given by the (G′/G)-expansion technique [38]. Therefore, the results of EDAM and the (G′/G)-expansion approach may be more broadly obtained from the generalized results in our study.

Appendix A.2. Fractional Generalization

We may generalize our findings to the fractional form. As an example, let us consider the (4) with conformable derivatives of δ-order, i.e.,

$$D_t^\delta Q + D_x^\delta Q + D_y^\delta Q^3 - D_{xzt}^{3\delta} Q = 0, \tag{A3}$$

then, all of our solutions are also those of Conformable WBBM (CWBBM) in (A3), where ϕ is defined as follows:

$$\phi = \frac{\lambda x^\delta}{\delta} + \frac{\mu y^\delta}{\delta} + \frac{\nu z^\delta}{\delta} - \frac{\omega t^\delta}{\delta}. \tag{A4}$$

References

1. El-Ajou, A.; Arqub, O.A.; Momani, S. Approximate analytical solution of the nonlinear fractional KdV-Burgers equation: A new iterative algorithm. *J. Comput. Phys.* **2015**, *293*, 81–95. [CrossRef]
2. Alyobi, S.; Shah, R.; Khan, A.; Shah, N.A.; Nonlaopon, K. Fractional Analysis of Nonlinear Boussinesq Equation under Atangana-Baleanu-Caputo Operator. *Symmetry* **2022**, *14*, 2417. [CrossRef]
3. Khader, M.M.; Saad, K.M. A numerical study by using the Chebyshev collocation method for a problem of biological invasion: Fractional Fisher equation. *Int. J. Biomath.* **2018**, *11*, 1850099. [CrossRef]
4. Ma, H.C.; Peng, X.F.; Yao, D.D. Improved hyperbolic function method and exact solutions for variable coefficient Benjamin-Bona-Mahony-Burgers equation. *Therm. Sci.* **2015**, *19*, 1183–1187. [CrossRef]
5. Klein, C.; Sparber, C.; Markowich, P. Numerical study of fractional nonlinear Schrödinger equations. *Proc. R. Soc. Math. Phys. Eng. Sci.* **2014**, *470*, 20140364. [CrossRef] [PubMed]

6. Zhao, X.; Xu, W. Travelling wave solutions for a class of the generalized Benjamin-Bona-Mahoney equations. *Appl. Math. Comput.* **2007**, *192*, 507–519. [CrossRef]
7. Nagatani, T. Modified KdV equation for jamming transition in the continuum models of traffic. *Phys. Stat. Mech. Its Appl.* **1998**, *261*, 599–607. [CrossRef]
8. Yusufoğlu, E.; Bekir, A. Exact solutions of coupled nonlinear evolution equations. *Chaos Solitons Fractals* **2008**, *37*, 842–848. [CrossRef]
9. Yavuz, M.; Abdeljawad, T. Nonlinear regularized long-wave models with a new integral transformation applied to the fractional derivative with power and Mittag-Leffler kernel. *Adv. Differ. Equ.* **2020**, *2020*, 367. [CrossRef]
10. Benjamin, T.B. *Lectures on Nonlinear Wave Motion*; Lecture Notes in Applied Mathematics; American Mathematical Society: Providence, RI, USA, 1974; Volume 15, pp. 3–47.
11. Kuru, S. Compactons and kink-like solutions of BBM-like equations by means of factorization. *Chaos Solitons Fractals* **2009**, *42*, 626–633. [CrossRef]
12. Wazzan, L. A modified tanh-coth method for solving the KdV and the KdV-Burgers' equations. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, *14*, 443–450. [CrossRef]
13. Martynenko, A.; Chen, Y. Degradation kinetics of total anthocyanins and formation of polymeric color in blueberry hydrothermodynamic (HTD) processing. *J. Food Eng.* **2016**, *171*, 44–51. [CrossRef]
14. Davidson, P.A. An Introduction to Magnetohydrodynamics. *Am. J. Phys.* **2002**, *70*, 781. [CrossRef]
15. Bona, J.L.; Dougalis, V.A. An initial-and boundary-value problem for a model equation for propagation of long waves. *J. Math. Anal. Appl.* **1980**, *75*, 503–522. [CrossRef]
16. Bhimani, D.G.; Haque, S. Norm Inflation for Benjamin-Bona-Mahony Equation in Fourier Amalgam and Wiener Amalgam Spaces with Negative Regularity. *Mathematics* **2021**, *9*, 3145. [CrossRef]
17. Yokus, A.; Sulaiman, T.A.; Bulut, H. On the analytical and numerical solutions of the Benjamin-Bona-Mahony equation. *Opt. Quantum Electron.* **2018**, *50*, 31. [CrossRef]
18. Sahoo, S.; Saha Ray, S. A novel approach for stochastic solutions of wick-type stochastic time-fractional Benjamin-Bona-Mahony equation for modeling long surface gravity waves of small amplitude. *Stoch. Anal. Appl.* **2019**, *37*, 377–387. [CrossRef]
19. Rizvi, S.T.; Seadawy, A.R.; Ahmed, S.; Younis, M.; Ali, K. Study of multiple lump and rogue waves to the generalized unstable space time fractional nonlinear Schrödinger equation. *Chaos Solitons Fractals* **2021**, *151*, 111251. [CrossRef]
20. Darvishi, M.T.; Najafi, M.; Wazwaz, A.M. Soliton solutions for Boussinesq-like equations with spatio-temporal dispersion. *Ocean. Eng.* **2017**, *130*, 228–240. [CrossRef]
21. Ali, R.; Barak, S.; Altalbe, A. Analytical study of soliton dynamics in the realm of fractional extended shallow water wave equations. *Phys. Scr.* **2024**, *99*, 065235. [CrossRef]
22. Shah, R.; Hyder, A.A.; Iqbal, N.; Botmart, T. Fractional view evaluation system of Schrödinger-KdV equation by a comparative analysis. *AIMS Math.* **2022**, *7*, 19846–19864. [CrossRef]
23. Hammad, M.M.A.; Shah, R.; Alotaibi, B.M.; Alotiby, M.; Tiofack, C.G.L.; Alrowaily, A.W.; El-Tantawy, S.A. On the modified versions of (G′/G)-expansion technique for analyzing the fractional coupled Higgs system. *AIP Adv.* **2023**, *13*, 105131. [CrossRef]
24. Lou, S.Y.; Chen, L.L. Formal variable separation approach for nonintegrable models. *J. Math. Phys.* **1999**, *40*, 6491–6500. [CrossRef]
25. Alipour, M.M.; Domairry, G.; Davodi, A.G. An application of exp-function method to approximate general and explicit solutions for nonlinear Schrödinger equations. *Numer. Methods Partial. Differ. Equ.* **2011**, *27*, 1016–1025. [CrossRef]
26. Yang, X.J. A new integral transform method for solving steady heat-transfer problem. *Therm. Sci.* **2016**, *20* (Suppl. S3), 639–642. [CrossRef]
27. Alshehry, A.S.; Yasmin, H.; Shah, R. A homotopy perturbation method with Elzaki transformation for solving the fractional Biswas-Milovic model. *Open Phys.* **2023**, *21*, 20230147. [CrossRef]
28. Odibat, Z.; Momani, S.; Erturk, V.S. Generalized differential transform method: Application to differential equations of fractional order. *Appl. Math. Comput.* **2008**, *197*, 467–477. [CrossRef]
29. Noor, S.; Alyousef, H.A.; Shafee, A.; Shah, R.; El-Tantawy, S.A. A novel analytical technique for analyzing the (3+1)-dimensional fractional calogero-bogoyavlenskii-schiff equation: Investigating solitary/shock waves and many others physical phenomena. *Phys. Scr.* **2024**, *99*, 065257. [CrossRef]
30. Ali, R.; Kumar, D.; Akgül, A.; Altalbe, A. On the periodic soliton solutions for fractional schrodinger equations. *Fractals* **2024**, 32, 2440033. [CrossRef]
31. Bilal, M.; Iqbal, J.; Ali, R.; Awwad, F.A.; Ismail, E.A.A. Exploring Families of Solitary Wave Solutions for the Fractional Coupled Higgs System Using Modified Extended Direct Algebraic Method. *Fractal Fract.* **2023**, *7*, 653. [CrossRef]
32. Ali, R.; Hendy, A.S.; Ali, M.R.; Hassan, A.M.; Awwad, F.A.; Ismail, E.A. Exploring Propagating Soliton Solutions for the Fractional Kudryashov-Sinelshchikov Equation in a Mixture of Liquid-Gas Bubbles under the Consideration of Heat Transfer and Viscosity. *Fractal Fract.* **2023**, *7*, 773. [CrossRef]
33. Mohammed, F.A.; Elboree, M.K. Soliton solutions and periodic solutions for two models arises in mathematical physics. *AIMS Math.* **2021**, *7*, 4439–4458. [CrossRef]
34. Asjad, M.I.; Majid, S.Z.; Faridi, W.A.; Eldin, S.M. Sensitive analysis of soliton solutions of nonlinear Landau-Ginzburg-Higgs equation with generalized projective Riccati method. *AIMS Math.* **2023**, *8*, 10210–10227. [CrossRef]

35. Kuo, C.K.; Kumar, D.; Juan, C.J. A study of resonance Y-type multi-soliton solutions and soliton molecules for new (2+1)-dimensional nonlinear wave equations. *AIMS Math.* **2022**, *7*, 20740–20751. [CrossRef]
36. Xiao, Y.; Barak, S.; Hleili, M.; Shah, K. Exploring the dynamical behaviour of optical solitons in integrable kairat-II and kairat-X equations. *Phys. Scr.* **2024**, *99*, 095261. [CrossRef]
37. Bilal, M.; Ahmad, J. New exact solitary wave solutions for the 3D-FWBBM model in arising shallow water waves by two analytical methods. *Results Phys.* **2021**, *25*, 104230.
38. Akbar, M.A.; Ali, N.H.M.; Mohyud-Din, S.T. The alternative (G′/G)-expansion method with generalized Riccati equation: Application to fifth order (1+ 1)-dimensional Caudrey-Dodd-Gibbon equation. *Int. J. Phys. Sci.* **2012**, *7*, 743–752.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

mathematics

MDPI

Article

Exploring Kink Solitons in the Context of Klein–Gordon Equations via the Extended Direct Algebraic Method

Saleh Alshammari [1,*], Othman Abdullah Almatroud [1], Mohammad Alshammari [1], Hamzeh Zureigat [2] and M. Mossa Al-Sawalha [1,*]

[1] Deparment of Mathematics, College of Science, University of Ha'il, Ha'il 2440, Saudi Arabia; o.almatroud@uoh.edu.sa (O.A.A.); dar.alshammari@uoh.edu.sa (M.A.)
[2] Department of Mathematics, Faculty of Science and Technology, Jadara University, Irbid 21110, Jordan; hamzeh.zu@jadara.edu.jo
* Correspondence: saleh.alshammari@uoh.edu.sa (S.A.); m.alswalha@uoh.edu.sa (M.M.A.-S.)

Abstract: This work employs the Extended Direct Algebraic Method (EDAM) to solve quadratic and cubic nonlinear Klein–Gordon Equations (KGEs), which are standard models in particle and quantum physics that describe the dynamics of scaler particles with spin zero in the framework of Einstein's theory of relativity. By applying variables-based wave transformations, the targeted KGEs are converted into Nonlinear Ordinary Differential Equations (NODEs). The resultant NODEs are subsequently reduced to a set of nonlinear algebraic equations through the assumption of series-based solutions for them. New families of soliton solutions are obtained in the form of hyperbolic, trigonometric, exponential and rational functions when these systems are solved using Maple. A few soliton solutions are considered for certain values of the given parameters with the help of contour and 3D plots, which indicate that the solitons exist in the form of dark kink, hump kink, lump-like kink, bright kink and cuspon kink solitons. These soliton solutions are relevant to actual physics, for instance, in the context of particle physics and theories of quantum fields. These solutions are useful also for the enhancement of our understanding of the basic particle interactions and wave dynamics at all levels of physics, including but not limited to cosmology, compact matter physics and nonlinear optics.

Keywords: Klein–Gordon equations; nonlinear partial differential equations; extended direct algebraic method; kink solitons; explicit solutions

MSC: 33B15; 34A34; 35A20; 35A22; 44A10

check for updates

Citation: Alshammari, S.; Almatroud, O.A.; Alshammari, M.; Zureigat, H.; Al-Sawalha, M.M. Exploring Kink Solitons in the Context of Klein–Gordon Equations via the Extended Direct Algebraic Method. *Mathematics* **2024**, *12*, 3433. https://doi.org/10.3390/math12213433

Academic Editors: Teng Huang, Qiong Wang, Yan Pang and Xiangmin Jiao

Received: 29 August 2024
Revised: 30 October 2024
Accepted: 1 November 2024
Published: 2 November 2024

1. Introduction

A more generalised form of the nonlinear Klein–Gordon Equation (KGE) with different nonlinearities can be articulated as follows:

$$u_{tt} - \nu u_{xx} + P'(u) = 0, \tag{1}$$

where $u = u(x,t)$, ν is a constant and $P(u)$ is a suitable nonlinear function that is frequently selected as the potential energy [1]. Choosing $P'(u)$ results in the generation of a number of equations that can be used to simulate a wide range of physical processes, from quantum physics to wave motion, including sine-Gordon, Landau–Ginzburg–Higgs, PHI-4 and reaction Duffing equations [2]. Kragh discusses the KGE's inception, stating that prominent physicists such as Klein, Fock and Schrödinger were involved in revealing the connection between general relativity theory and some of the older forms of the KGE, which is one of the most significant equations, from which the canonical version of the nonlinear Schrödinger equation can be derived [3,4]. Galehouse [5] used suitable gauge transformations to obtain the KGE geometrically. Moreover, Schechter provided proof

of the scattering theory for the more general form of the KGE [6]. The KGE's equivalent theory is created by adhering to the traditional process comprising a comparable equation in a certain finite-energy normed Hilbert space that contains first-order time [7]. In this article, Weder also provided evidence for the invariance principle, interweaving relations and the existence and completeness of the wave operators. Thus, similar eigenfunction expansions can be applied to a certain field case in order to construct the spectral and scattering theory of the KGE, and the strong results are similar to the Schrödinger case [8]. Tsukanov discusses the translation motion of a KGE in an external field [9].

Since its introduction, the KGE has yielded a wide range of solutions that describe many physical phenomena, including kinks and solitary waves. Tariq et al., in [10], developed a new plethora of exact solutions such as bright and dark solitary wave solutions, kink solitary wave solutions, anti-kink solitary wave solutions, periodic solitary wave solutions and hyperbolic function solutions using the Sardar sub-equation and extended Fan sub-equation methods with stability analysis of the observed solutions. Onyenegecha constructed analytical solutions of a D-dimensional KGE with modified Mobius squared potential [11]. By using the reduced differential transform method, Belayeh et al. found approximate analytic solutions to a two-dimensional nonlinear KGE [12]. Ahmadov et al. used the Nikiforov–Uvarov method to construct analytical solutions for the KGE with combined exponential-type and ring-shaped potentials [13]. Abdeljabbar et al. explored bright and dark soliton solutions in the realm of quadratic KGEs [14]. Applying the extended first kind elliptic sub-equation approach, certain bell- and kink-type solitary wave solutions produced by applying some hyperbolic functions for the KGE were constructed [15]. Using the modified simple equation method, Akter and Akbar derived similar solutions [16]. Adomian applied the decomposition technique to the KGE's initial boundary value problems [17]. Kudryavtsev [18] studied the interaction of two solitons that are weakly oscillatory, self-localised and damped for a specific form of the KGE. In addition to the analytical solutions previously outlined, a range of numerical techniques have been used to solve KGE-related problems. A few difficulties based on the KGE were resolved by Strauss and Vazquez [19] using the most straightforward central second difference approach. The energy-saving scheme is better suited to model the long-term behaviour of the solutions than the other two, according to the numerical integration of the problem using four explicit finite-difference methods [20]. Another way to solve the KGE numerically is via a collocation method based on radial basis functions of thin plate splines [21]. Integrated radial basis function network methods and multi-quadratic quasi-interpolation are used to approximate the solutions of the non-homogenous version of the KGE [22]. In order to generate numerical solutions of the KGE, classical polynomial cubic B-spline functions have also been adapted for the collocation method [23–27].

Motivated by the ongoing research on KGEs, in this work, we use the EDAM to construct and analyse new families of kink soliton solutions for two KGEs, i.e., the quadratic KGE and the cubic KGE. These models are articulated as follows:

a. When $\nu = 1$ and $P'(u) = mu - nu^2$ are substituted into (1), where m and n are non-zero constants, then the quadratic KGE takes the following shape [28]:

$$u_{tt} - u_{xx} + mu - nu^2 = 0. \tag{2}$$

b. When $P'(u) = mu - nu^3$ is substituted into (1), then the cubic KGE takes the following shape [29]:

$$u_{tt} - \nu u_{xx} + mu - nu^3 = 0. \tag{3}$$

KGEs are prominent models in particle and quantum physics which articulate the dynamics of scaler particles (particles with spin zero) in the context of special relativity. The unfinished task of the current study is to use the EDAM to generate kink soliton solutions for (2) and (3). The KGEs are converted into NODEs using variables-based wave transformation. From the resultant NODEs, numerous soliton solutions can be obtained by further reducing them into an assortment of nonlinear algebraic equations under the

presumption that they have a series-based solution. With the use of illustrated contour and 3D graphs, the dynamics of a few soliton solutions are investigated for certain values of the supplied parameters. This shows that the solitons take on the structures of several kink wave types, including dark, hump, lump-like, bright and cuspon kink solitons. These graphics also confirm the presence of kink waves in both KGE models. The found soliton solutions have significant physical implications, especially in the context of quantum field theories and particle physics.

The format of the rest of the article is as follows: The Section 2 goes into further detail about the EDAM's methodology. For its induction to the KGEs, we present Section 3. In Section 4, we provide illustrations of particular kink soliton solutions along with an explanation, while the Section 5 summarises our investigation.

2. The Working Methodology of EDAM

In this section, we present a general description of the EDAM [30–32]. Look into the general Nonlinear Partial Differential Equation (NPDE) in the following form:

$$E(\varrho, \varrho_t, \varrho_{\sigma_1}, \varrho_{\sigma_2}, \varrho\varrho_{\sigma_1}, \dots) = 0, \tag{4}$$

where $\varrho = \varrho(t, \sigma_1, \sigma_2, \sigma_3, \dots, \sigma_n)$. In order to solve Equation (4), the following steps are followed:

Step 1. Equation (4) is first subjected to a variable transformation of the form $\varrho(t, \sigma_1, \sigma_2, \sigma_3, \dot{,}\sigma_n) = \Pi(\eta)$, where $\eta = \eta(t, \sigma_1, \sigma_2, \sigma_3, \dot{,}\sigma_n)$, which may be expressed in a many ways. This transformation converts (4) into an NODE with the following structure:

$$F(\Pi, \Pi', \Pi\ \Pi', \dots) = 0, \tag{5}$$

where $\Pi' = \frac{d\Pi}{d\eta}$.

Step 2. Next, we suppose the subsequent series-based solution for (5):

$$\Pi(\eta) = \sum_{m=0}^{N} C_m (G(\eta))^m \quad \text{or} \quad \Pi(\eta) = \sum_{m=-N}^{N} C_m (G(\eta))^m. \tag{6}$$

Here, C_m symbolises parameters which need to be estimated, while $G(\eta)$ satisfies the Ricatti equation of the following form:

$$G'(\eta) = p + qG(\eta) + r(G(\eta))^2, \tag{7}$$

where p, q, r are constants.

Step 3. We obtain a positive integer N (as expressed in Equation (6)) whenever we try to find a homogeneous balance between the dominant nonlinear term and highest-order derivative in Equation (5). By using the ensuing formulae, we obtain the balance number N more precisely:

$$N + s = N\zeta + r(s + N),$$

such that

$$D(\frac{d^\eta \Pi}{d\eta^s}) = N + s,$$
$$D(\Pi^\zeta (\frac{d^s \Pi}{d\eta^s})^r) = N\zeta + r(s + N),$$

where D shows the degree of $\Pi(\eta)$ and s, ζ and r are positive integers.

Step 4. We then substitute (6) into (5) or integrate (5), developing the equation, and place all the terms in order of $G(\eta)$ to form a polynomial in $G(\eta)$. When all the coefficients

of the derived polynomial are set to zero, an algebraic system of equations for $C_m s$ and other parameters are generated.

Step 5. Then, Maple is utilised to solve this system of nonlinear algebraic equations.

Step 6. The travelling wave solutions, which can be obtained from Equation (4), are found by solving for the unknown parameters and substituting the resultant solution into Equation (5) together with the appropriate $G(\eta)$ solution from Equation (7). Using the general solution of Equation (7), we can generate the families of travelling soliton solutions shown below:

Family 1. *For* $\Lambda < 0$ *and* $r \neq 0$,

$$G_1(\eta) = -\frac{q}{2r} + \frac{\sqrt{-\Lambda}\tan\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{2r},$$

$$G_2(\eta) = -\frac{q}{2r} - \frac{\sqrt{-\Lambda}\cot\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{2r},$$

$$G_3(\eta) = -\frac{q}{2r} + \frac{\sqrt{-\Lambda}\left(\tan\left(\sqrt{-\Lambda}\eta\right) + \left(\sec\left(\sqrt{-\Lambda}\eta\right)\right)\right)}{2r},$$

$$G_4(\eta) = -\frac{q}{2r} - \frac{\sqrt{-\Lambda}\left(\cot\left(\sqrt{-\Lambda}\eta\right) + \left(\csc\left(\sqrt{-\Lambda}\eta\right)\right)\right)}{2r},$$

and

$$G_5(\eta) = -\frac{q}{2r} + \frac{\sqrt{-\Lambda}\left(\tan\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right) - \cot\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right)\right)}{4r}.$$

Family 2. *For* $\Lambda > 0$ *and* $r \neq 0$,

$$G_6(\eta) = -\frac{q}{2r} - \frac{\sqrt{\Lambda}\tanh\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{2r},$$

$$G_7(\eta) = -\frac{q}{2r} - \frac{\sqrt{\Lambda}\coth\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{2r},$$

$$G_8(\eta) = -\frac{q}{2r} - \frac{\sqrt{\Lambda}\left(\tanh\left(\sqrt{\Lambda}\eta\right) + i\left(\operatorname{sech}\left(\sqrt{\Lambda}\eta\right)\right)\right)}{2r},$$

$$G_9(\eta) = -\frac{q}{2r} - \frac{\sqrt{\Lambda}\left(\coth\left(\sqrt{\Lambda}\eta\right) + \left(\operatorname{csch}\left(\sqrt{\Lambda}\eta\right)\right)\right)}{2r},$$

and

$$G_{10}(\eta) = -\frac{q}{2r} - \frac{\sqrt{\Lambda}\left(\tanh\left(\frac{1}{4}\sqrt{\Lambda}\eta\right) - \coth\left(\frac{1}{4}\sqrt{\Lambda}\eta\right)\right)}{4r}.$$

Family 3. *For* $pr > 0$ *and* $q = 0$,

$$G_{11}(\eta) = \sqrt{\frac{p}{r}}\tan(\sqrt{pr}\eta),$$

$$G_{12}(\eta) = -\sqrt{\frac{p}{r}}\cot(\sqrt{pr}\eta),$$

$$G_{13}(\eta) = \sqrt{\frac{p}{r}}(\tan(2\sqrt{pr}\eta) + (\sec(2\sqrt{pr}\eta))),$$

$$G_{14}(\eta) = -\sqrt{\frac{p}{r}}\left(\cot\left(2\sqrt{df}\eta\right) + (\csc(2\sqrt{pr}\eta))\right),$$

and

$$G_{15}(\eta) = \frac{1}{2}\sqrt{\frac{p}{r}}\left(\tan\left(\frac{1}{2}\sqrt{pr}\eta\right) - \cot\left(\frac{1}{2}\sqrt{pr}\eta\right)\right).$$

Family 4. *For $pr < 0$ and $q = 0$,*

$$G_{16}(\eta) = -\sqrt{-\frac{p}{r}}\tanh(\sqrt{-pr}\eta),$$

$$G_{17}(\eta) = -\sqrt{-\frac{p}{r}}\coth(\sqrt{-pr}\eta),$$

$$G_{18}(\eta) = -\sqrt{-\frac{p}{r}}(\tanh(2\sqrt{-pr}\eta) + (i\mathrm{sech}(2\sqrt{-pr}\eta))),$$

$$G_{19}(\eta) = -\sqrt{-\frac{p}{r}}(\coth(2\sqrt{-pr}\eta) + (\mathrm{csch}(2\sqrt{-pr}\eta))),$$

and

$$G_{20}(\eta) = -\frac{1}{2}\sqrt{-\frac{p}{r}}\left(\tanh\left(\frac{1}{2}\sqrt{pr}\eta\right) + \coth\left(\frac{1}{2}\sqrt{-pr}\eta\right)\right).$$

Family 5. *For $r = p$ and $q = 0$,*

$$G_{21}(\eta) = \tan(p\eta),$$

$$G_{22}(\eta) = -\cot(p\eta),$$

$$G_{23}(\eta) = \tan(2\,p\eta) + (\sec(2\,p\eta)),$$

$$G_{24}(\eta) = -\cot(2\,p\eta) + (\csc(2\,p\eta)),$$

and

$$G_{25}(\eta) = \frac{1}{2}\tan\left(\frac{1}{2}\,p\eta\right) - \frac{1}{2}\cot\left(\frac{1}{2}\,p\eta\right).$$

Family 6. *For $r = -p$ and $q = 0$,*

$$G_{26}(\eta) = -\tanh(p\eta),$$

$$G_{27}(\eta) = -\coth(p\eta),$$

$$G_{28}(\eta) = -\tanh(2\,p\eta) + (i\mathrm{sech}(2\,p\eta)),$$

$$G_{29}(\eta) = -\coth(2\,p\eta) + (\mathrm{csch}(2\,p\eta)),$$

and

$$G_{30}(\eta) = -\frac{1}{2}\tanh\left(\frac{1}{2}\,p\eta\right) - \frac{1}{2}\coth\left(\frac{1}{2}\,p\eta\right).$$

Family 7. *For $\Lambda = 0$,*

$$G_{31}(\eta) = -2\,\frac{p(q\eta\ + 2)}{q^2\eta}.$$

Family 8. *For $r = 0$, $q = \lambda$ and $d = s\lambda$ (with $s \neq 0$),*

$$G_{32}(\eta) = \mathrm{e}^{\lambda\eta} - s.$$

Family 9. *For $q = r = 0$,*

$$G_{33}(\eta) = p\eta.$$

Family 10. *For* $q = p = 0$,

$$G_{34}(\eta) = -\frac{1}{r\eta}.$$

Family 11. *For* $q \neq 0$, $r \neq 0$ *and* $p = 0$,

$$G_{35}(\eta) = -\frac{q}{r(\cosh(q\eta) - \sinh(q\eta) + 1)},$$

and

$$G_{36}(\eta) = -\frac{q(\cosh(q\eta) + \sinh(q\eta))}{r(\cosh(q\eta) + \sinh(q\eta) + 1)}.$$

Family 12. *For* $q = \lambda$, $r = s\lambda$ *(with* $s \neq 0$*) and* $p = 0$,

$$G_{37}(\eta) = \frac{e^{\lambda\eta}}{1 - se^{\lambda\eta}}.$$

In the above solutions, $\Lambda = q^2 - 4rp$.

3. Construction of Soliton Solutions for KGEs

In this section, we evaluate soliton solutions using the EDAM for the nonlinear KGEs.

3.1. Example 1

First, we evaluate soliton solutions for the quadratic KGE stated in (2) as follows:

$$u_{tt} - u_{xx} + mu - nu^2 = 0. \tag{8}$$

Taking the travelling wave transformation of the form

$$u(x,t) = U(\eta), \eta = bx - at, \tag{9}$$

where a and b denote non-zero constants, we obtain the following by the chain rule:

$$\begin{aligned}
u_t(x,t) &= \frac{\partial}{\partial t}U(\eta) = U'(\eta)\frac{\partial}{\partial t}\eta = -aU'(\eta),\\
u_{tt}(x,t) &= \frac{\partial}{\partial t}(-aU'(\eta)) = -aU''(\eta)\frac{\partial}{\partial t}\eta = a^2U''(\eta),\\
u_x(x,t) &= \frac{\partial}{\partial x}U(\eta) = U'(\eta)\frac{\partial}{\partial x}\eta = bU'(\eta),\\
u_{xx}(x,t) &= \frac{\partial}{\partial x}(bU'(\eta)) = bU''(\eta)\frac{\partial}{\partial x}\eta = b^2U''(\eta).
\end{aligned}$$

Equation (2) can be transformed into the following NODE:

$$(a^2 - b^2)U'' + mU - nU^2 = 0. \tag{10}$$

To find the balance number N presented in (7), we balance the highest-order derivative term U'' with the highest-power nonlinear term U^2 in the above equation, which determines $N + 2 = 2N$. Thus, $N = 2$. Putting $N = 2$ into (7), starting from $N = 0$, suggests the following closed-form solution for (10):

$$U(\eta) = C_0 + C_1G(\eta) + C_2(G(\eta))^2. \tag{11}$$

The arbitrary constants C_0, C_1 and C_2 are placeholders that need to be found. A substitution of (11) with (10) yields an expression in $G(\eta)$. After gathering terms with the same

powers of $G(\eta)$ and equating the coefficients to zero, we obtain the following system of nonlinear algebraic equations:

$$-b^2C_1qp-2\,b^2C_2p^2+mC_0-n{C_0}^2+a^2C_1qp+2\,a^2C_2p^2=0,$$

$$a^2C_1q^2+mC_1-b^2C_1q^2-6\,b^2C_2pq+2\,a^2C_1rp+6\,a^2C_2pq-2\,b^2C_1rp-2\,nC_0C_1=0,$$

$$3\,a^2C_1qr-4\,b^2C_2q^2+4\,a^2C_2q^2+8\,a^2C_2pr-8\,b^2C_2pr-3\,b^2C_1qr-2\,nC_0C_2+mC_2-n{C_1}^2=0,$$

$$-10\,b^2C_2qr-2\,nC_1C_2+2\,a^2C_1r^2+10\,a^2C_2qr-2\,b^2C_1r^2=0,$$

and

$$-n{C_2}^2-6\,b^2C_2r^2+6\,a^2C_2r^2=0.$$

When this system is solved using the Maple tool, it leads to the following three cases of solutions:

Case 1.

$$C_0=C_0,C_1=\frac{C_0q}{p},C_2=\frac{C_0r}{p},a=a,b=b,m=(b^2-a^2)\Lambda,n=6\,\frac{r\left(a^2-b^2\right)p}{C_0}. \tag{12}$$

Case 2.

$$C_0=\frac{1}{6}\,\frac{C_2\left(q^2+2\,pr\right)}{r^2},C_1=\frac{C_2q}{r},C_2=C_2,a=a,b=b,m=(a^2-b^2)\Lambda,n=6\,\frac{r^2\left(a^2-b^2\right)}{C_2}. \tag{13}$$

Case 3.

$$C_0=C_0,C_1=C_1,C_2=C_2,a=\pm b,b=b,m=0,n=0. \tag{14}$$

Considering Case 1 and utilizing Equations (11), (9) and the corresponding general solution of (7), we construct the ensuing families of soliton solutions for the quadratic KGE in (2):

Family 13. *For* $\Lambda<0$ *and* $r\neq0$,

$$u_{1,1,1}(x,t)=C_0+\frac{C_0q}{p}\left(-\frac{1}{2}\,\frac{q}{r}+\frac{1}{2}\,\frac{\sqrt{-\Lambda}\tan\left(\frac{1}{2}\,\sqrt{-\Lambda}\eta\right)}{r}\right)+\frac{C_0r}{p}\left(-\frac{1}{2}\,\frac{q}{r}+\frac{1}{2}\,\frac{\sqrt{-\Lambda}\tan\left(\frac{1}{2}\,\sqrt{-\Lambda}\eta\right)}{r}\right)^2, \tag{15}$$

$$u_{1,1,2}(x,t)=C_0+\frac{C_0q}{p}\left(-\frac{1}{2}\,\frac{q}{r}-\frac{1}{2}\,\frac{\sqrt{-\Lambda}\cot\left(\frac{1}{2}\,\sqrt{-\Lambda}\eta\right)}{r}\right)+\frac{C_0r}{p}\left(-\frac{1}{2}\,\frac{q}{r}-\frac{1}{2}\,\frac{\sqrt{-\Lambda}\cot\left(\frac{1}{2}\,\sqrt{-\Lambda}\eta\right)}{r}\right)^2, \tag{16}$$

$$\begin{aligned}u_{1,1,3}(x,t)=&C_0+\frac{C_0q}{p}\left(-\frac{1}{2}\,\frac{q}{r}+\frac{1}{2}\,\frac{\sqrt{-\Lambda}\left(\tan\left(\sqrt{-\Lambda}\eta\right)+\sec\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right)\\&+\frac{C_0r}{p}\left(-\frac{1}{2}\,\frac{q}{r}+\frac{1}{2}\,\frac{\sqrt{-\Lambda}\left(\tan\left(\sqrt{-\Lambda}\eta\right)+\sec\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^2,\end{aligned} \tag{17}$$

$$u_{1,1,4}(x,t) = C_0 + \frac{C_0 q}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\cot\left(\sqrt{-\Lambda}\eta\right) + \csc\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right) + \frac{C_0 r}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\cot\left(\sqrt{-\Lambda}\eta\right) + \csc\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^2, \tag{18}$$

$$u_{1,1,5}(x,t) = C_0 + \frac{C_0 q}{p}\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{4}\frac{\sqrt{-\Lambda}\left(\tan\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right) - \cot\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right)\right)}{r}\right) + \frac{C_0 r}{p}\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{4}\frac{\sqrt{-\Lambda}\left(\tan\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right) + \cot\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^2. \tag{19}$$

Family 14. *For* $\Lambda > 0$ *and* $r \neq 0$,

$$u_{1,1,6}(x,t) = C_0 + \frac{C_0 q}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\tanh\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right) + \frac{C_0 r}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\tanh\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right)^2, \tag{20}$$

$$u_{1,1,7}(x,t) = C_0 + \frac{C_0 q}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\coth\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right) + \frac{C_0 r}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\coth\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right)^2, \tag{21}$$

$$u_{1,1,8}(x,t) = C_0 + \frac{C_0 q}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\tanh\left(\sqrt{\Lambda}\eta\right) + \mathrm{i}\,\mathrm{sech}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right) + \frac{C_0 r}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\tanh\left(\sqrt{\Lambda}\eta\right) + \mathrm{i}\,\mathrm{sech}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right)^2, \tag{22}$$

$$u_{1,1,9}(x,t) = C_0 + \frac{C_0 q}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\coth\left(\sqrt{\Lambda}\eta\right) + \mathrm{csch}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right) + \frac{C_0 r}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\coth\left(\sqrt{\Lambda}\eta\right) + \mathrm{csch}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right)^2, \tag{23}$$

$$u_{1,1,10}(x,t) = C_0 + \frac{C_0 q}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{4}\frac{\sqrt{\Lambda}\left(\tanh\left(\frac{1}{4}\sqrt{\Lambda}\eta\right) - \coth\left(\frac{1}{4}\sqrt{\Lambda}\eta\right)\right)}{r}\right) + \frac{C_0 r}{p}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{4}\frac{\sqrt{\Lambda}\left(\tanh\left(\frac{1}{4}\sqrt{\Lambda}\eta\right) - \coth\left(\frac{1}{4}\sqrt{\Lambda}\eta\right)\right)}{r}\right)^2. \tag{24}$$

Family 15. *For* $pr > 0$ *and* $q = 0$,

$$u_{1,1,11}(x,t) = C_0 + C_0(\tan(\sqrt{rp}\eta))^2, \tag{25}$$

$$u_{1,1,12}(x,t) = C_0 + C_0(\cot(\sqrt{pr}\eta))^2, \tag{26}$$

$$u_{1,1,13}(x,t) = C_0 + C_0(\tan(2\sqrt{pr}\eta) + \sec(2\sqrt{pr}\eta))^2, \tag{27}$$

$$u_{1,1,14}(x,t) = C_0 + C_0(\cot(2\sqrt{pr}\eta) + \csc(2\sqrt{pr}\eta))^2, \tag{28}$$

$$u_{1,1,15}(x,t) = C_0 + \frac{1}{4}\,C_0\left(\tan\left(\frac{1}{2}\,\sqrt{pr}\eta\right) - \cot\left(\frac{1}{2}\,\sqrt{pr}\eta\right)\right)^2. \tag{29}$$

Family 16. *For $pr < 0$ and $q = 0$,*

$$u_{1,1,16}(x,t) = C_0 - C_0(\tan(\sqrt{rp}\eta))^2, \tag{30}$$

$$u_{1,1,17}(x,t) = C_0 - C_0(\coth(\sqrt{rp}\eta))^2, \tag{31}$$

$$u_{1,1,18}(x,t) = C_0 - C_0\big(\tanh\big(2\,\sqrt{-pr}\eta\big) + \big(i\mathrm{sech}\big(2\,\sqrt{-pr}\eta\big)\big)\big)^2, \tag{32}$$

$$u_{1,1,19}(x,t) = C_0 - C_0\big(\coth\big(2\,\sqrt{-pr}\eta\big) + \big(\mathrm{csch}\big(2\,\sqrt{-pr}\eta\big)\big)\big)^2, \tag{33}$$

$$u_{1,1,20}(x,t) = C_0 - \frac{1}{4}\,C_0\left(\frac{1}{2}\,\tanh\big(\sqrt{-rp}\eta\big) + \frac{1}{2}\,\coth\big(\sqrt{-rp}\eta\big)\right)^2. \tag{34}$$

Family 17. *For $r = p$ and $q = 0$,*

$$u_{1,1,21}(x,t) = C_0 + C_0(\tan(p\eta))^2, \tag{35}$$

$$u_{1,1,22}(x,t) = C_0 + C_0(\cot(p\eta))^2, \tag{36}$$

$$u_{1,1,23}(x,t) = C_0 + C_0(\tan(2\,p\eta) + \sec(2\,p\eta))^2, \tag{37}$$

$$u_{1,1,24}(x,t) = C_0 + C_0(-\cot(2\,p\eta) + \csc(2\,p\eta))^2, \tag{38}$$

$$u_{1,1,25}(x,t) = C_0 + C_0\left(\frac{1}{2}\tan\left(\frac{1}{2}\,p\eta\right) - \frac{1}{2}\,\cot\left(\frac{1}{2}\,p\eta\right)\right)^2. \tag{39}$$

Family 18. *For $r = -p$ and $q = 0$,*

$$u_{1,1,26}(x,t) = C_0 - C_0(\tanh(p\eta))^2, \tag{40}$$

$$u_{1,1,27}(x,t) = C_0 - C_0(\coth(p\eta))^2, \tag{41}$$

$$u_{1,1,28}(x,t) = C_0 - C_0(-\tanh(2\,p\eta) + i\,\mathrm{sech}(2\,p\eta))^2, \tag{42}$$

$$u_{1,1,29}(x,t) = C_0 - C_0(-\coth(2\,p\eta) + \mathrm{csch}(2\,p\eta))^2, \tag{43}$$

$$u_{1,1,30}(x,t) = C_0 - C_0\left(-\frac{1}{2}\,\tanh\left(\frac{1}{2}\,p\eta\right) - \frac{1}{2}\,\coth\left(\frac{1}{2}\,p\eta\right)\right)^2. \tag{44}$$

Family 19. *For $\Lambda = 0$,*

$$u_{1,1,31}(x,t) = C_0 - 2\,\frac{C_0 p(q\eta + 2)}{qr\eta} + 4\,\frac{C_0 rp(q\eta + 2)^2}{q^4\eta^2}. \tag{45}$$

Family 20. *For $r = 0$, $q = \lambda$ and $p = s\lambda (s \neq= 0)$,*

$$u_{1,1,32}(x,t) = C_0 + \frac{C_0(\mathrm{e}^{\lambda\eta} - s)}{s}. \tag{46}$$

In the above solutions, $\eta = bx - at$.

Considering Case 2 and utilizing Equations (11), (9) and the corresponding general solution of (7), we construct the ensuing families of soliton solutions for the quadratic KGE in (2):

Family 21. *For $\Lambda < 0$ and $r \neq 0$,*

$$u_{1,2,1}(x,t) = \frac{C_2 q}{r}\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\tan\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\tan\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right)^2 + \frac{1}{6}\frac{C_2(q^2+2pr)}{r^2}, \tag{47}$$

$$u_{1,2,2}(x,t) = \frac{C_2 q}{r}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\cot\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\cot\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right)^2 + \frac{1}{6}\frac{C_2(q^2+2pr)}{r^2}, \tag{48}$$

$$\begin{aligned} u_{1,2,3}(x,t) = {} & \frac{C_2 q}{r}\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\tan\left(\sqrt{-\Lambda}\eta\right) + \sec\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right) \\ & + C_2\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\tan\left(\sqrt{-\Lambda}\eta\right) + \sec\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^2 + \frac{1}{6}\frac{C_2(q^2+2pr)}{r^2}, \end{aligned} \tag{49}$$

$$\begin{aligned} u_{1,2,4}(x,t) = {} & \frac{C_2 q}{r}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\cot\left(\sqrt{-\Lambda}\eta\right) + \csc\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right) \\ & + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\cot\left(\sqrt{-\Lambda}\eta\right) + \csc\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^2 + \frac{1}{6}\frac{C_2(q^2+2pr)}{r^2}, \end{aligned} \tag{50}$$

$$\begin{aligned} u_{1,2,5}(x,t) = {} & \frac{C_2 q}{r}\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{4}\frac{\sqrt{-\Lambda}\left(\tan\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right) - \cot\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right)\right)}{r}\right) \\ & + C_2\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{4}\frac{\sqrt{-\Lambda}\left(\tan\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right) - \cot\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^2 + \frac{1}{6}\frac{C_2(q^2+2pr)}{r^2}. \end{aligned} \tag{51}$$

Family 22. *For* $\Lambda > 0$ *and* $r \neq 0$,

$$u_{1,2,6}(x,t) = \frac{C_2 q}{r}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\tanh\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\tanh\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right)^2 + \frac{1}{6}\frac{C_2(q^2+2pr)}{r^2}, \tag{52}$$

$$u_{1,2,7}(x,t) = \frac{C_2 q}{r}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\coth\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\coth\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right)^2 + \frac{1}{6}\frac{C_2(q^2+2pr)}{r^2}, \tag{53}$$

$$\begin{aligned} u_{1,2,8}(x,t) = {} & \frac{C_2 q}{r}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\tanh\left(\sqrt{\Lambda}\eta\right) + i\,\mathrm{sech}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right) \\ & + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\tanh\left(\sqrt{\Lambda}\eta\right) + i\,\mathrm{sech}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right)^2 + \frac{1}{6}\frac{C_2(q^2+2pr)}{r^2}, \end{aligned} \tag{54}$$

$$\begin{aligned} u_{1,2,9}(x,t) = {} & \frac{C_2 q}{r}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\coth\left(\sqrt{\Lambda}\eta\right) + \mathrm{csch}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right) \\ & + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\coth\left(\sqrt{\Lambda}\eta\right) + \mathrm{csch}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right)^2 + \frac{1}{6}\frac{C_2(q^2+2pr)}{r^2}, \end{aligned} \tag{55}$$

$$u_{1,2,10}(x,t) = \frac{C_2 q}{r}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{4}\frac{\sqrt{\Lambda}\left(\tanh\left(\frac{1}{4}\sqrt{\Lambda}\eta\right) - \coth\left(\frac{1}{4}\sqrt{\Lambda}\eta\right)\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{4}\frac{\sqrt{\Lambda}\left(\tanh\left(\frac{1}{4}\sqrt{\Lambda}\eta\right) - \coth\left(\frac{1}{4}\sqrt{\Lambda}\eta\right)\right)}{r}\right)^2 + \frac{1}{6}\frac{C_2(q^2+2pr)}{r^2}. \tag{56}$$

Family 23. *For $pr > 0$ and $q = 0$,*

$$u_{1,2,11}(x,t) = \frac{1}{3}\frac{C_2 p}{r} + \frac{C_2 p\left(\tan\left(\sqrt{pr}\eta\right)\right)^2}{r}, \tag{57}$$

$$u_{1,2,12}(x,t) = \frac{1}{3}\frac{C_2 p}{r} + \frac{C_2 p\left(\cot\left(\sqrt{pr}\eta\right)\right)^2}{r}, \tag{58}$$

$$u_{1,2,13}(x,t) = \frac{1}{3}\frac{C_2 p}{r} + \frac{C_2 p\left(\tan\left(2\sqrt{pr}\eta\right) + \sec\left(2\sqrt{\mathrm{rp}}\eta\right)\right)^2}{r} \tag{59}$$

$$u_{1,2,14}(x,t) = \frac{1}{3}\frac{C_2 p}{r} + \frac{C_2 p\left(\cot\left(2\sqrt{pr}\eta\right) + \csc\left(2\sqrt{pr}\eta\right)\right)^2}{r}, \tag{60}$$

$$u_{1,2,15}(x,t) = \frac{1}{3}\frac{C_2 p}{r} + \frac{1}{4}\frac{C_2 p\left(\tan(1/2\sqrt{pr}\eta) - \cot\left(\frac{1}{2}\sqrt{pr}\eta\right)\right)^2}{r}. \tag{61}$$

Family 24. *For $pr < 0$ and $q = 0$,*

$$u_{1,2,16}(x,t) = \frac{1}{3}\frac{C_2 p}{r} - \frac{C_2 p(\tanh(\sqrt{-pr}\eta))^2}{r}, \tag{62}$$

$$u_{1,2,17}(x,t) = \frac{1}{3}\frac{C_2 p}{r} - \frac{C_2 p(\coth(\sqrt{-pr}\eta))^2}{r}, \tag{63}$$

$$u_{1,2,18}(x,t) = \frac{1}{3}\frac{C_2 p}{r} - \frac{C_2 p(\coth(2\sqrt{-pr}\eta) + \operatorname{csch}(2\sqrt{-pr}\eta))^2}{r}, \tag{64}$$

$$u_{1,2,19}(x,t) = \frac{1}{3}\frac{C_2 p}{r} - \frac{C_2 p(\tanh(2\sqrt{-pr}\eta) + \mathrm{i}\operatorname{sech}(2\sqrt{-pr}\eta))^2}{r}, \tag{65}$$

$$u_{1,2,20}(x,t) = \frac{1}{3}\frac{C_2 p}{r} - \frac{1}{4}\frac{C_2 p\left(\tanh\left(\frac{1}{2}\sqrt{-pr}\eta\right) + \coth\left(\frac{1}{2}\sqrt{-pr}\eta\right)\right)^2}{r}. \tag{66}$$

Family 25. *For $r = p$ and $q = 0$,*

$$u_{1,2,21}(x,t) = \frac{1}{3}C_2 + C_2(\tan(r\eta))^2, \tag{67}$$

$$u_{1,2,22}(x,t) = \frac{1}{3}C_2 + C_2(\cot(r\eta))^2, \tag{68}$$

$$u_{1,2,23}(x,t) = \frac{1}{3}C_2 + C_2(\tan(2r\eta) + \sec(2r\eta))^2, \tag{69}$$

$$u_{1,2,24}(x,t) = \frac{1}{3}C_2 + C_2(-\cot(2r\eta) + \csc(2r\eta))^2, \tag{70}$$

$$u_{1,2,25}(x,t) = \frac{1}{3}C_2 + C_2\left(\frac{1}{2}\tan\left(\frac{1}{2}r\eta\right) - \frac{1}{2}\cot\left(\frac{1}{2}r\eta\right)\right)^2. \tag{71}$$

Family 26. *For $r = -p$ and $q = 0$,*

$$u_{1,2,26}(x,t) = -\frac{1}{3}C_2 + C_2(\tanh(p\eta))^2, \tag{72}$$

$$u_{1,2,27}(x,t) = -\frac{1}{3}C_2 + C_2(\coth(p\eta))^2, \tag{73}$$

$$u_{1,2,28}(x,t) = -\frac{1}{3}C_2 + C_2(-\tanh(2\,p\eta) + \mathrm{i}\,\mathrm{sech}(2\,p\eta))^2, \tag{74}$$

$$u_{1,2,29}(x,t) = -\frac{1}{3}C_2 + C_2(-\coth(2\,p\eta) + \mathrm{csch}(2\,p\eta))^2, \tag{75}$$

$$u_{1,2,30}(x,t) = -\frac{1}{3}C_2 + C_2\left(-\frac{1}{2}\tanh\left(\frac{1}{2}\,p\eta\right) - \frac{1}{2}\coth\left(\frac{1}{2}\,p\eta\right)\right)^2. \tag{76}$$

Family 27. *For $\Lambda = 0$,*

$$u_{1,2,31}(x,t) = \frac{1}{6}\,\frac{C_2(q^2+2\,pr)}{r^2} + \frac{C_2 q}{r}\left(-2\,\frac{p(q\eta\,+2)}{q^2\eta}\right) + C_2\left(-2\,\frac{p(q\eta\,+2)}{q^2\eta}\right)^2. \tag{77}$$

Family 28. *For $q = p = 0$,*

$$u_{1,2,32}(x,t) = \frac{C_2}{r^2\eta^2}. \tag{78}$$

Family 29. *For $q \neq 0$, $r \neq 0$ and $p = 0$,*

$$u_{1,2,33}(x,t) = \frac{1}{6}\,\frac{C_2 q^2}{r^2} - \frac{C_2 q^2}{r^2(\cosh(q\eta) - \sinh(q\eta) + 1)} + \frac{C_2 q^2}{r^2(\cosh(q\eta) - \sinh(q\eta) + 1)^2}. \tag{79}$$

and

$$u_{1,2,34}(x,t) = \frac{1}{6}\,\frac{C_2 q^2}{r^2} + \frac{C_2 q}{r}\left(-\frac{q(\cosh(q\eta) + \sinh(q\eta))}{r(\cosh(q\eta) + \sinh(q\eta) + 1)}\right) + C_2\left(-\frac{q(\cosh(q\eta) + \sinh(q\eta))}{r(\cosh(q\eta) + \sinh(q\eta) + 1)}\right)^2. \tag{80}$$

Family 30. *For $q = \lambda$, $r = s\lambda$ (with $s \neq 0$) and $p = 0$,*

$$u_{1,2,35}(x,t) = \frac{1}{6}\,\frac{C_2\lambda^2}{s\lambda^2} + \frac{C_2\lambda\,\mathrm{e}^{\lambda\,\eta}}{s\lambda\,(1 - s\mathrm{e}^{\lambda\,\eta})} + \frac{C_2\left(\mathrm{e}^{\lambda\,\eta}\right)^2}{\left(1 - s\mathrm{e}^{\lambda\,\eta}\right)^2}. \tag{81}$$

In the above solutions, $\eta = bx - at$.

Considering Case 3 and utilizing Equations (11), (9) and the corresponding general solution of (7), we construct the ensuing families of soliton solutions for the quadratic KGE in (2):

Family 31. *For $\Lambda < 0$ and $r \neq 0$,*

$$u_{1,3,1}(x,t) = C_0 + C_1\left(-\frac{1}{2}\,\frac{q}{r} + \frac{1}{2}\,\frac{\sqrt{-\Lambda}\tan\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right) + C_2\left(-\frac{1}{2}\,\frac{q}{r} + \frac{1}{2}\,\frac{\sqrt{-\Lambda}\tan\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right)^2, \tag{82}$$

$$u_{1,3,2}(x,t) = C_0 + C_1\left(-\frac{1}{2}\,\frac{q}{r} - \frac{1}{2}\,\frac{\sqrt{-\Lambda}\cot\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right) + C_2\left(-\frac{1}{2}\,\frac{q}{r} - \frac{1}{2}\,\frac{\sqrt{-\Lambda}\cot\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right)^2, \tag{83}$$

$$u_{1,3,3}(x,t) = C_0 + C_1\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\tan\left(\sqrt{-\Lambda}\eta\right) + \sec\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\tan\left(\sqrt{-\Lambda}\eta\right) + \sec\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^2, \tag{84}$$

$$u_{1,3,4}(x,t) = C_0 + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\cot\left(\sqrt{-\Lambda}\eta\right) + \csc\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\cot\left(\sqrt{-\Lambda}\eta\right) + \csc\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^2, \tag{85}$$

$$u_{1,3,5}(x,t) = C_0 + C_1\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{4}\frac{\sqrt{-\Lambda}\left(\tan\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right) - \cot\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right)\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{4}\frac{\sqrt{-\Lambda}\left(\tan\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right) - \cot\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^2. \tag{86}$$

Family 32. *For* $\Lambda > 0$ *and* $r \neq 0$,

$$u_{1,3,6}(x,t) = C_0 + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\tanh\left(\sqrt{\Lambda}\eta\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\tanh\left(\sqrt{\Lambda}\eta\right)}{r}\right)^2, \tag{87}$$

$$u_{1,3,7}(x,t) = C_0 + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\coth\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\coth\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right)^2, \tag{88}$$

$$u_{1,3,8}(x,t) = C_0 + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\tanh\left(\sqrt{\Lambda}\eta\right) + i\,\mathrm{sech}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\tanh\left(\sqrt{\Lambda}\eta\right) + i\,\mathrm{sech}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right)^2, \tag{89}$$

$$u_{1,3,9}(x,t) = C_0 + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\coth\left(\sqrt{\Lambda}\eta\right) + \mathrm{csch}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\coth\left(\sqrt{\Lambda}\eta\right) + \mathrm{csch}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right)^2, \tag{90}$$

$$u_{1,3,10}(x,t) = C_0 + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{4}\frac{\sqrt{\Lambda}\left(\tanh\left(\frac{1}{4}\sqrt{\Lambda}\eta\right) - \coth\left(\frac{1}{4}\sqrt{\Lambda}\eta\right)\right)}{r}\right) + C_2\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{4}\frac{\sqrt{\Lambda}\left(\tanh\left(\frac{1}{4}\sqrt{\Lambda}\eta\right) - \coth\left(\frac{1}{4}\sqrt{\Lambda}\eta\right)\right)}{r}\right)^2. \tag{91}$$

Family 33. *For* $pr > 0$ *and* $q = 0$,

$$u_{1,3,11}(x,t) = C_0 + C_1\sqrt{\frac{p}{r}}\tan(\sqrt{rp}\eta) + \frac{C_2 p\left(\tan\left(\sqrt{rp}\eta\right)\right)^2}{r}, \tag{92}$$

$$u_{1,3,12}(x,t) = C_0 - C_1\sqrt{\frac{p}{r}}\cot(\sqrt{rp}\eta) + \frac{C_2 p\left(\cot\left(\sqrt{rp}\eta\right)\right)^2}{r}, \tag{93}$$

$$u_{1,3,13}(x,t) = C_0 + C_1\sqrt{\frac{p}{r}}\left(\tan(2\sqrt{rp}\eta) + \sec\left(2\sqrt{\mathrm{rp}}\eta\right)\right) + \frac{C_2 p\left(\tan\left(2\sqrt{rp}\eta\right) + \sec\left(2\sqrt{\mathrm{rp}}\eta\right)\right)^2}{r}, \tag{94}$$

$$u_{1,3,14}(x,t) = C_0 - C_1\sqrt{\frac{p}{r}}(\cot(2\sqrt{rp}\eta) + \csc(2\sqrt{rp}\eta)) + \frac{C_2 p\left(\cot\left(2\sqrt{rp}\eta\right) + \csc\left(2\sqrt{rp}\eta\right)\right)^2}{r}, \tag{95}$$

$$u_{1,3,15}(x,t) = C_0 + \frac{1}{2}C_1\sqrt{\frac{p}{r}}\left(\tan\left(\frac{1}{2}\sqrt{rp}\eta\right) - \cot\left(\frac{1}{2}\sqrt{rp}\eta\right)\right) + \frac{1}{4}\frac{C_2 p\left(\tan\left(\frac{1}{2}\sqrt{rp}\eta\right) - \cot\left(\frac{1}{2}\sqrt{rp}\eta\right)\right)^2}{r}. \tag{96}$$

Family 34. *For $pr < 0$ and $q = 0$,*

$$u_{1,3,16}(x,t) = C_0 - C_1\sqrt{-\frac{p}{r}}\tanh\left(\sqrt{-rp}\eta\right) - \frac{C_2 p(\tanh(\sqrt{-rp}\eta))^2}{r}, \tag{97}$$

$$u_{1,3,17}(x,t) = C_0 - C_1\sqrt{-\frac{p}{r}}\coth\left(\sqrt{-rp}\eta\right) - \frac{C_2 p(\coth(\sqrt{-rp}\eta))^2}{r} \tag{98}$$

$$\begin{aligned} u_{1,3,18}(x,t) =& C_0 - C_1\sqrt{-\frac{p}{r}}\left(\tanh\left(2\sqrt{-pr}\eta\right) + \mathrm{isech}\left(2\sqrt{-pr}\eta\right)\right) \\ & - \frac{C_2 p(\tanh(2\sqrt{-pr}\eta) + \mathrm{isech}(2\sqrt{-pr}\eta))^2}{r}, \end{aligned} \tag{99}$$

$$\begin{aligned} u_{1,3,19}(x,t) =& C_0 - C_1\sqrt{-\frac{p}{r}}\left(\coth\left(2\sqrt{-rp}\eta\right) + \mathrm{csch}\left(2\sqrt{-pr}\eta\right)\right) \\ & - \frac{C_2 p(\coth(2\sqrt{-rp}\eta) + \mathrm{csch}(2\sqrt{-pr}\eta))^2}{r}, \end{aligned} \tag{100}$$

$$\begin{aligned} u_{1,3,20}(x,t) =& C_0 - \frac{1}{2}C_1\sqrt{-\frac{p}{r}}\left(\tanh\left(\frac{1}{2}\sqrt{-pr}\eta\right) + \coth\left(\frac{1}{2}\sqrt{-pr}\eta\right)\right) \\ & - \frac{1}{4}\frac{C_2 p\left(\tanh\left(\frac{1}{2}\sqrt{-pr}\eta\right) + \coth\left(\frac{1}{2}\sqrt{-pr}\eta\right)\right)^2}{r}. \end{aligned} \tag{101}$$

Family 35. *For $r = p$ and $q = 0$,*

$$u_{1,3,21}(x,t) = C_0 + C_1\tan(r\eta) + C_2(\tan(r\eta))^2, \tag{102}$$

$$u_{1,3,22}(x,t) = C_0 - C_1\cot(r\eta) + C_2(\cot(r\eta))^2, \tag{103}$$

$$u_{1,3,23}(x,t) = C_0 + C_1(\tan(2r\eta) + \sec(2r\eta)) + C_2(\tan(2r\eta) + \sec(2r\eta))^2, \tag{104}$$

$$u_{1,3,24}(x,t) = C_0 + C_1(-\cot(2r\eta) + \csc(2r\eta)) + C_2(-\cot(2r\eta) + \csc(2r\eta))^2, \tag{105}$$

$$u_{1,3,25}(x,t) = C_0 + C_1\left(\frac{1}{2}\tan\left(\frac{1}{2}r\eta\right) - \frac{1}{2}\cot\left(\frac{1}{2}r\eta\right)\right) + C_2\left(\frac{1}{2}\tan\left(\frac{1}{2}r\eta\right) - \frac{1}{2}\cot\left(\frac{1}{2}r\eta\right)\right)^2. \tag{106}$$

Family 36. *For $r = -p$ and $q = 0$,*

$$u_{1,3,26}(x,t) = C_0 - C_1\tanh(p\eta) + C_2(\tanh(p\eta))^2, \tag{107}$$

$$u_{1,3,27}(x,t) = C_0 - C_1 \coth(p\eta) + C_2(\coth(p\eta))^2, \tag{108}$$

$$u_{1,3,28}(x,t) = C_0 + C_1(-\tanh(2\,p\eta) + i\,\mathrm{sech}(2\,p\eta)) + C_2(-\tanh(2\,p\eta) + i\,\mathrm{sech}(2\,p\eta))^2, \tag{109}$$

$$u_{1,3,29}(x,t) = C_0 + C_1(-\coth(2\,p\eta) + \mathrm{csch}(2\,p\eta)) + C_2(-\coth(2\,p\eta) + \mathrm{csch}(2\,p\eta))^2, \tag{110}$$

$$u_{1,3,30}(x,t) = C_0 + C_1\left(-\frac{1}{2}\tanh\left(\frac{1}{2}\,p\eta\right) - \frac{1}{2}\coth\left(\frac{1}{2}\,p\eta\right)\right) + C_2\left(-\frac{1}{2}\tanh\left(\frac{1}{2}\,p\eta\right) - \frac{1}{2}\coth\left(\frac{1}{2}\,p\eta\right)\right)^2. \tag{111}$$

Family 37. *For $r = 0$, $q = \lambda$ and $p = s\lambda$ (with $s \neq 0$),*

$$u_{1,3,31}(x,t) = C_0 + C_1\left(-2\,\frac{p(q\eta+2)}{q^2\eta}\right) + C_2\left(-2\,\frac{p(q\eta+2)}{q^2\eta}\right)^2. \tag{112}$$

Family 38. *For $r = 0$, $q = \lambda$ and $p = s\lambda$ (with $s \neq 0$),*

$$u_{1,3,32}(x,t) = C_0 + C_1\left(\mathrm{e}^{\lambda\,\Lambda} - s\right) + C_2\left(\mathrm{e}^{\lambda\,\Lambda} - s\right)^2. \tag{113}$$

Family 39. *For $q = r = 0$,*

$$u_{1,3,33}(x,t) = C_0 + C_1 p\eta + C_2 p^2\eta^2. \tag{114}$$

Family 40. *For $q = p = 0$,*

$$u_{1,3,34}(x,t) = C_0 - \frac{C_1}{r\eta} + \frac{C_2}{r^2\eta^2}. \tag{115}$$

Family 41. *For $q \neq 0$, $r \neq 0$ and $p = 0$,*

$$u_{1,3,35}(x,t) = C_0 - \frac{C_1 q}{r(\cosh(q\eta) - \sinh(q\eta) + 1)} + \frac{C_2 q^2}{r^2(\cosh(q\eta) - \sinh(q\eta) + 1)^2}, \tag{116}$$

and

$$u_{1,3,36}(x,t) = C_0 - \frac{C_1 q}{r}\left(-\frac{(\cosh(q\eta) + \sinh(q\eta))}{(\cosh(q\eta) + \sinh(q\eta) + 1)}\right) + \frac{C_2 q^2}{r^2}\left(-\frac{(\cosh(q\eta) + \sinh(q\eta))}{(\cosh(q\eta) + \sinh(q\eta) + 1)}\right)^2. \tag{117}$$

Family 42. *For $q = \lambda$, $r = s\lambda$ (with $s \neq 0$) and $p = 0$,*

$$u_{1,3,37}(x,t) = C_0 + \frac{C_1\mathrm{e}^{\lambda\,\eta}}{1 - s\mathrm{e}^{\lambda\,\eta}} + \frac{C_2\left(\mathrm{e}^{\lambda\,\eta}\right)^2}{\left(1 - s\mathrm{e}^{\lambda\,\eta}\right)^2}. \tag{118}$$

In the above solutions, $\eta = b(x \mp t)$.

3.2. Example 2

Now, we evaluate the soliton solutions for the cubic KGE stated in (3) as follows:

$$u_{tt} - \nu u_{xx} + mu - nu^3 = 0. \tag{119}$$

Taking the travelling wave transformation of the form

$$u(x,t) = u\eta, \quad \eta = x + \omega t, \tag{120}$$

where ω is a non-zero constant, we obtain the following by the chain rule:

$$u_t(x,t) = \frac{\partial}{\partial t}U(\eta) = U'(\eta)\frac{\partial}{\partial t}\eta = \omega U'(\eta),$$
$$u_{tt}(x,t) = \frac{\partial}{\partial t}(\omega U'(\eta)) = \omega U''(\eta)\frac{\partial}{\partial t}\eta = \omega^2 U''(\eta),$$
$$u_x(x,t) = \frac{\partial}{\partial x}U(\eta) = U'(\eta)\frac{\partial}{\partial x}\eta = U'(\eta),$$
$$u_{xx}(x,t) = \frac{\partial}{\partial x}(U'(\eta)) = U''(\eta)\frac{\partial}{\partial x}\eta = U''(\eta).$$

The above transformation reduces (3) to the ensuing nonlinear NODE:

$$(\omega^2 - \nu)U'' + mU - nU^3 = 0. \tag{121}$$

To find the balance number N presented in (7), we balance the highest-order derivative term U'' with the highest-power nonlinear term U^3 in the above equation. Putting $N = 1$ into (7), starting from $N = -1$, suggests the following closed-form solution for (121):

$$U(\eta) = C_{-1}G(\eta)^{-1} + C_0 + C_1 G(\eta). \tag{122}$$

The arbitrary constants C_{-1}, C_0 and C_1 are placeholders that need to be found. The substitution of (122) into (121) yields an expression in $G(\eta)$. After gathering terms with the same powers of $G(\eta)$ and equating the coefficients to zero, we obtain the following system of nonlinear algebraic equations:

$$-nC_1{}^3 - 2\,\nu\,C_1 r^2 + 2\,\omega^2 C_1 r^2 = 0,$$

$$3\,\omega^2 C_1 qr - 3\,\nu\,C_1 qr - 3\,nC_0 C_1{}^2 = 0,$$

$$-3\,nC_0{}^2 C_1 + mC_1 - 3\,nC_{-1}C_1{}^2 - 2\,\nu\,C_1 rp - \nu\,C_1 q^2 + 2\,\omega^2 C_1 rp + \omega^2 C_1 q^2 = 0,$$

$$-6\,nC_{-1}C_0 C_1 - \nu\,C_1 qp + \omega^2 C_{-1} qr - \nu\,C_{-1} qr + \omega^2 C_1 qp - nC_0{}^3 + mC_0 = 0,$$

$$2\,\omega^2 C_{-1} pr - 3\,nC_{-1}{}^2 C_1 - 2\,\nu\,C_{-1} pr + \omega^2 C_{-1} q^2 - 3\,nC_{-1}C_0{}^2 - \nu\,C_{-1} q^2 + mC_{-1} = 0,$$

$$-3\,nC_{-1}{}^2 C_0 - 3\,\nu\,C_{-1} pq + 3\,\omega^2 C_{-1} pq = 0,$$

and

$$-2\,\nu\,C_{-1} p^2 - nC_{-1}{}^3 + 2\,\omega^2 C_{-1} p^2 = 0.$$

When this system is solved using the Maple tool, it leads to the following two cases of solutions:

Case 4.

$$C_{-1} = 0, C_0 = \frac{1}{2}\frac{C_1 q}{r}, C_1 = C_1, \omega = \omega, \nu = \frac{-\omega^2\Lambda + 2\,m}{-\Lambda}, n = -4\,\frac{mr^2}{(-\Lambda)C_1{}^2}. \tag{123}$$

Case 5.

$$C_{-1} = C_{-1}, C_0 = \frac{1}{2}\frac{qC_{-1}}{p}, C_1 = 0, \omega = \omega, \nu = \frac{-\omega^2\Lambda + 2\,m}{-\Lambda}, n = -4\,\frac{mp^2}{(-\Lambda)C_{-1}{}^2}. \tag{124}$$

Considering Case 4 and utilizing Equations (120), (122) and the corresponding general solution of (7) we construct the ensuing families of soliton solutions for the cubic KGE in (3):

Family 43. *For* $\Lambda < 0$ *and* $r \neq 0$,

$$u_{2,1,1}(x,t) = \frac{1}{2}\frac{C_1 q}{r} + C_1\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\tan\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right), \tag{125}$$

$$u_{2,1,2}(x,t) = \frac{1}{2}\frac{C_1 q}{r} + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\cot\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right), \tag{126}$$

$$u_{2,1,3}(x,t) = \frac{1}{2}\frac{C_1 q}{r} + C_1\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\tan\left(\sqrt{-\Lambda}\eta\right) + \sec\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right), \tag{127}$$

$$u_{2,1,4}(x,t) = \frac{1}{2}\frac{C_1 q}{r} + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\cot\left(\sqrt{-\Lambda}\eta\right) + \csc\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right), \tag{128}$$

$$u_{2,1,5}(x,t) = \frac{1}{2}\frac{C_1 q}{r} + C_1\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{4}\frac{\sqrt{-\Lambda}\left(\tan\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right) - \cot\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right)\right)}{r}\right). \tag{129}$$

Family 44. *For* $\Lambda > 0$ *and* $r \neq 0$,

$$u_{2,1,6}(x,t) = \frac{1}{2}\frac{C_1 q}{r} + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\tanh\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right), \tag{130}$$

$$u_{2,1,7}(x,t) = \frac{1}{2}\frac{C_1 q}{r} + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\coth\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right), \tag{131}$$

$$u_{2,1,8}(x,t) = \frac{1}{2}\frac{C_1 q}{r} + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\tanh\left(\sqrt{\Lambda}\eta\right) + i\,\mathrm{sech}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right), \tag{132}$$

$$u_{2,1,9}(x,t) = \frac{1}{2}\frac{C_1 q}{r} + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\coth\left(\sqrt{\Lambda}\eta\right) + \mathrm{csch}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right), \tag{133}$$

$$u_{2,1,10}(x,t) = \frac{1}{2}\frac{C_1 q}{r} + C_1\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{4}\frac{\sqrt{\Lambda}\left(\tanh\left(\frac{1}{4}\sqrt{\Lambda}\eta\right) - \coth\left(\frac{1}{4}\sqrt{\Lambda}\eta\right)\right)}{r}\right). \tag{134}$$

Family 45. *For* $pr > 0$ *and* $q = 0$,

$$u_{2,1,11}(x,t) = C_1\sqrt{\frac{p}{r}}\tan(\sqrt{rp}\eta), \tag{135}$$

$$u_{2,1,12}(x,t) = C_1\sqrt{\frac{p}{r}}(\cot(\sqrt{pr}\eta))^2, \tag{136}$$

$$u_{2,1,13}(x,t) = C_1\sqrt{\frac{p}{r}}\left(\tan(2\sqrt{rp}\eta) + \sec\left(2\sqrt{\mathrm{rp}}\eta\right)\right), \tag{137}$$

$$u_{2,1,14}(x,t) = -C_1\sqrt{\frac{p}{r}}(\cot(2\sqrt{rp}\eta) + \csc(2\sqrt{rp}\eta)), \tag{138}$$

$$u_{2,1,15}(x,t) = \frac{1}{2}C_1\sqrt{\frac{p}{r}}\left(\tan\left(\frac{1}{2}\sqrt{rp}\eta\right) - \cot\left(\frac{1}{2}\sqrt{rp}\eta\right)\right). \tag{139}$$

Family 46. *For $pr < 0$ and $q = 0$,*

$$u_{2,1,16}(x,t) = -C_1\sqrt{-\frac{p}{r}}\tanh\left(\sqrt{-rp}\eta\right), \tag{140}$$

$$u_{2,1,17}(x,t) = -C_1\sqrt{-\frac{p}{r}}\coth\left(\sqrt{-rp}\eta\right), \tag{141}$$

$$u_{2,1,18}(x,t) = -C_1\sqrt{-\frac{p}{r}}\left(\tanh\left(2\sqrt{-rp}\eta\right) + i\,\mathrm{sech}\left(2\sqrt{-rp}\eta\right)\right), \tag{142}$$

$$u_{2,1,19}(x,t) = -C_1\sqrt{-\frac{p}{r}}\left(\coth\left(2\sqrt{-rp}\eta\right) + \mathrm{csch}\left(2\sqrt{-rp}\eta\right)\right), \tag{143}$$

$$u_{2,1,20}(x,t) = -\frac{1}{2}C_1\sqrt{-\frac{p}{r}}\left(\tanh\left(\frac{1}{2}\sqrt{-rp}\eta\right) + \coth\left(\frac{1}{2}\sqrt{-rp}\eta\right)\right). \tag{144}$$

Family 47. *For $r = p$ and $q = 0$,*

$$u_{2,1,21}(x,t) = C_1\tan(r\eta), \tag{145}$$

$$u_{2,1,22}(x,t) = -C_1\cot(r\eta), \tag{146}$$

$$u_{2,1,23}(x,t) = C_1(\tan(2\,r\eta) + \sec(2\,r\eta)), \tag{147}$$

$$u_{2,1,24}(x,t) = C_1(-\cot(2\,r\eta) + \csc(2\,r\eta)), \tag{148}$$

$$u_{2,1,25}(x,t) = C_1\left(\frac{1}{2}\tan\left(\frac{1}{2}r\eta\right) - \frac{1}{2}\cot\left(\frac{1}{2}r\eta\right)\right). \tag{149}$$

Family 48. *For $r = -p$ and $q = 0$,*

$$u_{2,1,26}(x,t) = -C_1\tanh(p\eta), \tag{150}$$

$$u_{2,1,27}(x,t) = -C_1\coth(p\eta), \tag{151}$$

$$u_{2,1,28}(x,t) = C_1(-\tanh(2\,p\eta) + i\,\mathrm{sech}(2\,p\eta)), \tag{152}$$

$$u_{2,1,29}(x,t) = C_1(-\coth(2\,p\eta) + \mathrm{csch}(2\,p\eta)), \tag{153}$$

$$u_{2,1,30}(x,t) = C_1\left(-\frac{1}{2}\tanh\left(\frac{1}{2}p\eta\right) - \frac{1}{2}\coth\left(\frac{1}{2}p\eta\right)\right). \tag{154}$$

Family 49. *For $q \neq 0$, $r \neq 0$ and $p = 0$,*

$$u_{2,1,31}(x,t) = \frac{1}{2}\frac{C_1 q}{r} - \frac{C_1 q}{r(\cosh(q\eta) - \sinh(q\eta) + 1)}, \tag{155}$$

and

$$u_{2,1,32}(x,t) = \frac{1}{2}\frac{C_1 q}{r} - \frac{C_1 q}{r}\left(\frac{(\cosh(q\eta) + \sinh(q\eta))}{(\cosh(q\eta) + \sinh(q\eta) + 1)}\right). \tag{156}$$

Family 50. *For $q = \lambda$, $r = s\lambda$ (with $s \neq 0$) and $p = 0$,*

$$u_{2,1,33}(x,t) = \frac{1}{2}\frac{C_1}{s} + \frac{C_1 e^{\lambda\,\eta}}{1 - s e^{\lambda\,\eta}}. \tag{157}$$

In the above solutions, $\eta = x + \omega t$.

Considering Case 5 and utilizing Equations (120), (122) and the corresponding general solution of (7) we construct the ensuing families of soliton solutions for the cubic KGE in (3):

Family 51. *For* $\Lambda < 0$ *and* $r \neq 0$,

$$u_{2,2,1}(x,t) = C_{-1}\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\tan\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right)^{-1} + \frac{1}{2}\frac{qC_{-1}}{p}, \tag{158}$$

$$u_{2,2,2}(x,t) = C_{-1}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{-\Lambda}\cot\left(\frac{1}{2}\sqrt{-\Lambda}\eta\right)}{r}\right)^{-1} + \frac{1}{2}\frac{qC_{-1}}{p}, \tag{159}$$

$$u_{2,2,3}(x,t) = C_{-1}\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\tan\left(\sqrt{-\Lambda}\eta\right) + \sec\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^{-1} + \frac{1}{2}\frac{qC_{-1}}{p}, \tag{160}$$

$$u_{2,2,4}(x,t) = C_{-1}\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{2}\frac{\sqrt{-\Lambda}\left(\tan\left(\sqrt{-\Lambda}\eta\right) + \sec\left(\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^{-1} + \frac{1}{2}\frac{qC_{-1}}{p}, \tag{161}$$

$$u_{2,2,5}(x,t) = C_{-1}\left(-\frac{1}{2}\frac{q}{r} + \frac{1}{4}\frac{\sqrt{-\Lambda}\left(\tan\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right) - \cot\left(\frac{1}{4}\sqrt{-\Lambda}\eta\right)\right)}{r}\right)^{-1} + \frac{1}{2}\frac{qC_{-1}}{p}. \tag{162}$$

Family 52. *For* $\Lambda > 0$ *and* $r \neq 0$,

$$u_{2,2,6}(x,t) = C_{-1}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\tanh\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right)^{-1} + \frac{1}{2}\frac{qC_{-1}}{p}, \tag{163}$$

$$u_{2,2,7}(x,t) = C_{-1}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\coth\left(\frac{1}{2}\sqrt{\Lambda}\eta\right)}{r}\right)^{-1} + \frac{1}{2}\frac{qC_{-1}}{p}, \tag{164}$$

$$u_{2,2,8}(x,t) = C_{-1}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\tanh\left(\sqrt{\Lambda}\eta\right) + i\,\mathrm{sech}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right)^{-1} + \frac{1}{2}\frac{qC_{-1}}{p}, \tag{165}$$

$$u_{2,2,9}(x,t) = C_{-1}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{2}\frac{\sqrt{\Lambda}\left(\coth\left(\sqrt{\Lambda}\eta\right) + \mathrm{csch}\left(\sqrt{\Lambda}\eta\right)\right)}{r}\right)^{-1} + \frac{1}{2}\frac{qC_{-1}}{p}, \tag{166}$$

$$u_{2,2,10}(x,t) = C_{-1}\left(-\frac{1}{2}\frac{q}{r} - \frac{1}{4}\frac{\sqrt{\Lambda}\left(\tanh\left(\frac{1}{4}\sqrt{\Lambda}\eta\right) - \coth\left(\frac{1}{4}\sqrt{\Lambda}\eta\right)\right)}{r}\right)^{-1} + \frac{1}{2}\frac{qC_{-1}}{p}. \tag{167}$$

Family 53. *For* $pr > 0$ *and* $q = 0$,

$$u_{2,2,11}(x,t) = C_{-1}\sqrt{\frac{r}{p}}(\tan(\sqrt{rp}\eta))^{-1}, \tag{168}$$

$$u_{2,2,12}(x,t) = -C_{-1}\sqrt{\frac{r}{p}}(\cot(\sqrt{rp}\eta))^{-1}, \tag{169}$$

$$u_{2,2,13}(x,t) = C_{-1}\sqrt{\frac{r}{p}}\left(\tan(2\sqrt{rp}\eta) + \sec\left(2\sqrt{\mathrm{rp}}\eta\right)\right)^{-1}, \tag{170}$$

$$u_{2,2,14}(x,t) = -C_{-1}\sqrt{\frac{r}{p}}(\cot(2\sqrt{rp}\eta) + \csc(2\sqrt{rp}\eta))^{-1}, \tag{171}$$

$$u_{2,2,15}(x,t) = 2\,C_{-1}\sqrt{\frac{r}{p}}\left(\tan\left(\frac{1}{2}\sqrt{rp}\eta\right) - \cot\left(\frac{1}{2}\sqrt{rp}\eta\right)\right)^{-1}. \tag{172}$$

Family 54. *For $pr < 0$ and $q = 0$,*

$$u_{2,2,16}(x,t) = -C_{-1}\sqrt{-\frac{r}{p}}\left(\tanh\left(\sqrt{-rp}\eta\right)\right)^{-1}, \tag{173}$$

$$u_{2,2,17}(x,t) = -C_{-1}\sqrt{-\frac{r}{p}}\left(\coth\left(\sqrt{-rp}\eta\right)\right)^{-1}, \tag{174}$$

$$u_{2,2,18}(x,t) = -\,C_{-1}\sqrt{-\frac{r}{p}}\left(2\,\tanh\left(\sqrt{-rp}\eta\right) + 2\,i\mathrm{sech}\left(\sqrt{-rp}\eta\right)\right)^{-1} \tag{175}$$

$$u_{2,2,19}(x,t) = -C_{-1}\sqrt{-\frac{r}{p}}\left(\coth\left(2\sqrt{-rp}\eta\right) + \mathrm{csch}\left(2\sqrt{-rp}\eta\right)\right)^{-1}, \tag{176}$$

$$u_{2,2,20}(x,t) = -2\,C_{-1}\sqrt{-\frac{r}{p}}\left(\tanh\left(\frac{1}{2}\sqrt{-rp}\eta\right) + \coth\left(\frac{1}{2}\sqrt{-rp}\eta\right)\right)^{-1}. \tag{177}$$

Family 55. *For $r = p$ and $q = 0$,*

$$u_{2,2,21}(x,t) = \frac{C_{-1}}{\tan(r\eta)}, \tag{178}$$

$$u_{2,2,22}(x,t) = \frac{C_{-1}}{-\cot(r\eta)}, \tag{179}$$

$$u_{2,2,23}(x,t) = \frac{C_{-1}}{\tan(2\,r\eta) + \sec(2\,r\eta)}, \tag{180}$$

$$u_{2,2,24}(x,t) = \frac{C_{-1}}{-\cot(2\,r\eta) + \csc(2\,r\eta)}, \tag{181}$$

$$u_{2,2,25}(x,t) = \frac{C_{-1}}{\frac{1}{2}\,\tan\left(\frac{1}{2}\,r\eta\right) - \frac{1}{2}\,\cot\left(\frac{1}{2}\,r\eta\right)}, \tag{182}$$

Family 56. *For $r = -p$ and $q = 0$,*

$$u_{2,2,26}(x,t) = -\frac{C_{-1}}{\tanh(p\eta)}, \tag{183}$$

$$u_{2,2,27}(x,t) = -\frac{C_{-1}}{\coth(p\eta)}, \tag{184}$$

$$u_{2,2,28}(x,t) = \frac{C_{-1}}{-\tanh(2\,p\eta) + i\,\mathrm{sech}(2\,p\eta)}, \tag{185}$$

$$u_{2,2,29}(x,t) = \frac{C_{-1}}{-\coth(2\,p\eta) + \mathrm{csch}(2\,p\eta)}, \tag{186}$$

$$u_{2,2,30}(x,t) = \frac{C_{-1}}{-\frac{1}{2}\,\tanh\left(\frac{1}{2}\,p\eta\right) - \frac{1}{2}\,\coth\left(\frac{1}{2}\,p\eta\right)}. \tag{187}$$

Family 57. *For $r = 0$, $q = \lambda$ and $d = s\lambda$ (with $s \neq 0$),*

$$u_{2,2,31}(x,t) = \frac{C_{-1}}{e^{\lambda\Lambda} - s} + \frac{1}{2}\frac{C_{-1}}{s}. \tag{188}$$

In the above solutions, $\eta = x + \omega t$.

4. Discussion and Graphs

In this section, numerous kink soliton structures discovered in the framework of the two KGEs under investigation are visually displayed. Using the EDAM, we were able to identify and then demonstrate wave patterns in contour as well as in three-dimensional plots. The soliton solutions developed for the cubic and quadratic KGEs mostly include kink soliton profiles. When studying NPDEs associated with fluid dynamics, quantum field theory and optics, it is beneficial to examine solitons which are travelling waves that keep moving at a constant speed and formation. Such waveforms explain how scalar fields act within relativistic quantum physics when the KGEs are considered, like in vacuum state transition, kink solitons and sharp interfaces delineating fields with stable boundaries within scalar zones of different phases or amplitudes. By researching solitons to the KGEs, one is able to understand the basic characteristics of relativistic quantum mechanical systems and particles and the features of vacuums. Figure 1, The 3D and contour graphics of the bell-shaped dark soliton solution $u_{1,1,6}$ given in (20) are plotted for $q := 5$, $p := 1$, $r := 4$, $C_0 := 10$, $a := 2$ and $b := 4$. Figure 2, The 3D and contour graphics of the bright kink soliton solution $u_{1,1,17}$ given in (31) are plotted for $q := 0$, $p := -0.003$, $r := 0.005$, $C_0 := 0.001$, $a := 0.00005$ and $b := 0.00007$. Figure 3, The 3D and contour graphics of the bell-shaped kink soliton solution $u_{1,2,16}$ given in (62) are plotted for $q := 0$, $p := 4$, $r := -1$, $C_2 := 10$, $a := 0.005$ and $b := 0.005$. Figure 4, The 3D and contour graphics of the bright kink soliton solution $u_{1,2,31}$ given in (77) are plotted for $q := 2$, $p := 1$, $r := 1$, $C_2 := 70$, $a := 50$ and $b := 100$. Figure 5, The 3D and contour graphics of the bright kink soliton solution $u_{1,3,11}$ given in (92) are plotted for $q := 0$, $p := 1$, $r := 4$, $C_0 := 10$, $C_1 := 7$, $C_2 := 8$, $a := 2$ and $b := -2$. Kink soliton solutions, such as bright, dark, bell-shaped, cuspon and lump-like kink solutions, indicate different localised solutions or stimulations that result from nonlinear field concepts, corresponding to phenomena like topological deformity or particle-like structures in the context of KGEs and related fields like quantum field theory and quantum mechanics. Dark kinks show a localised dip in their profile, possibly signifying robust soliton-like features, whereas bright kinks show peaks or humps in their profile, possibly suggesting transitory or meta-stable excitations. Lump-type kinks compress on the field in pliable zones like lumps, whereas cuspon kinks have sharp cusp-like structures and frequently exhibit interesting dynamical behaviour. Bell-shaped kinks feature bell-like curves and can exhibit a range of behaviours depending on the circumstances. Such kinks are fascinating because of their ability to build robust or meta-stable localised frameworks, offer insight into the theory of field dynamics and play a role in breaking symmetry and phase shifts. Ultimately, these kinks will aid in our understanding of field theory, nonlinear interactions and the behaviour of physical systems in both the classical and quantum realms. Figure 6, The 3D and contour graphics of the twinning bright kink soliton solution $u_{1,3,27}$ given in (108) are plotted for $q := 0, p := 10, r := -10, C_0 := 5, C_1 := 0.8, C_2 := 0.100, a := 0.0080, b := 0.00110$. Figure 7, The 3D and contour graphics of the lump-like kink soliton solution $u_{2,1,3}$ given in (127) are plotted for $q := 1$, $p := 2$, $r := 3$, $C_1 := 0.25$ and $\omega := 0.0010$. Figure 8, The 3D and contour graphics of the bright kink soliton solution $u_{2,1,10}$ given in (134) are plotted for $q := 4$, $p := 1$, $r := 1$, $C_1 := 0.40$ and $\omega := 0.020$. Figure 9, The 3D and contour graphics of the kink soliton solution $u_{2,2,9}$ given in (166) are plotted for $q := 5$, $p := 2$, $r := 2$, $C_{-1} := 40$ and $\omega := 10$. Figure 10, The 3D and contour graphics of the cuspon kink soliton solution $u_{2,2,31}$ given in (188) are plotted for $q := 5$, $p := 10$, $r := 0$, $C_{-1} := 4$, $\omega := 20, s := 2, \lambda := 5$.

Furthermore, several analytical methods have been established in the literature to study soliton phenomena in nonlinear phenomena [33–37]. It is crucial to acknowledge that,

while these approaches greatly advance our comprehension of soliton dynamics and assist in linking them to the theories that explain phenomena, they may also have limitations and shortcomings (like the seven common mistakes) [38,39]. Moreover, several methods of this kind depend on the Riccati equation [40,41]. A general class of exact solutions to the Riccati equation (7) was systematically presented by [42]. Moreover, expression (6), for determining travelling waves, is precisely a rational function of the solution to the Riccati equation. Thus, the entire solution procedure of the study exemplifies the application of the transformed rational function method [43]. While there exist solitary solutions for the Riccati equation, these techniques are useful for examining soliton occurrences in nonlinear models [44]. Motivated by the above applications of the Riccati hypothesis, the present study used the EDAM, which incorporates the Riccati equation, to build and investigate soliton dynamics in two KGEs. This incorporation is helpful since it produces a large number of additional kink soliton solutions in five families of solutions, periodic, hyperbolic, rational, rational-hyperbolic and exponential, for the selected models. The solutions proposed here greatly expand our knowledge of soliton dynamics and allow us to relate the phenomena in the focused model to the underlying theories.

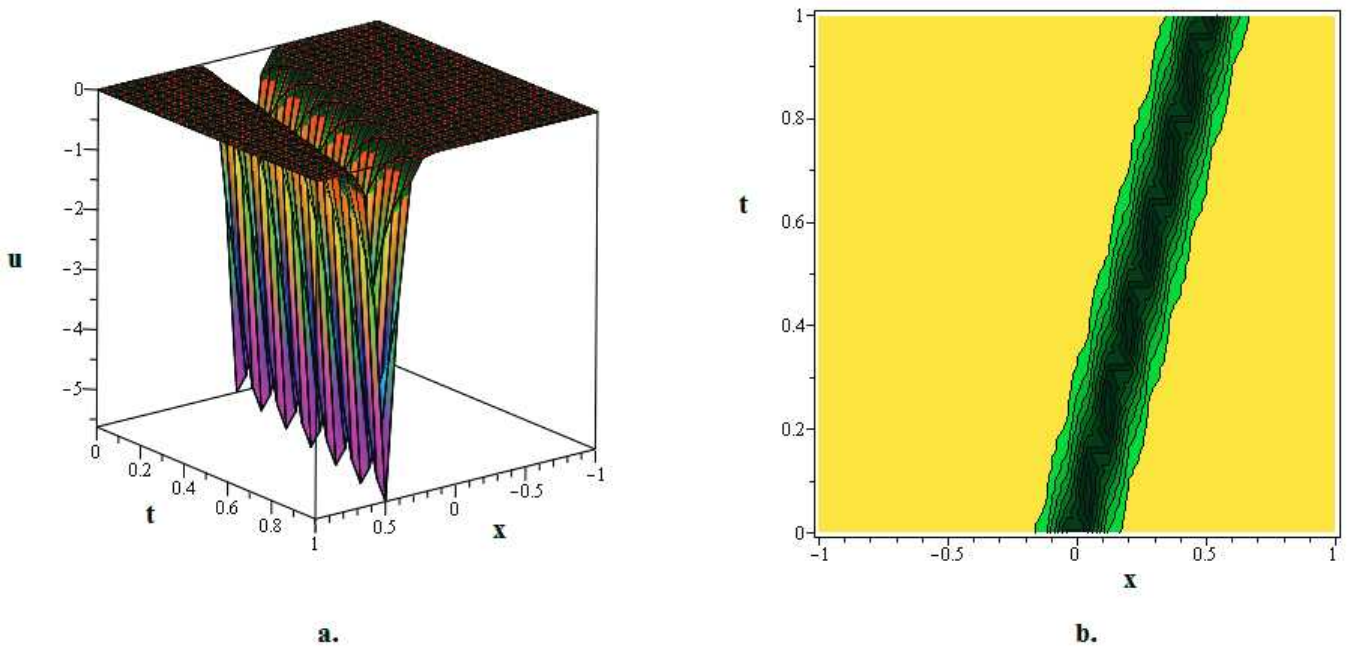


Figure 1. The 3D and contour graphics of the bell-shaped dark soliton solution $u_{1,1,6}$ given in (20) are plotted for $q := 5$, $p := 1$, $r := 4$, $C_0 := 10$, $a := 2$ and $b := 4$.

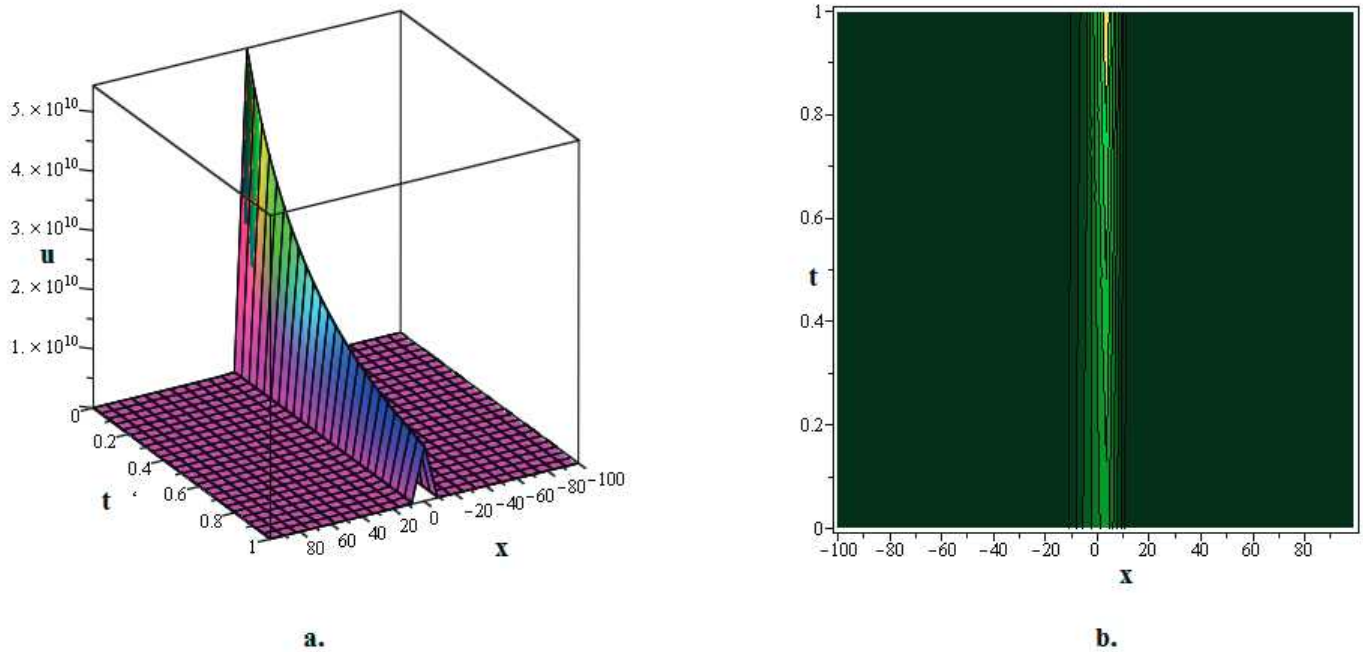


Figure 2. The 3D and contour graphics of the bright kink soliton solution $u_{1,1,17}$ given in (31) are plotted for $q := 0$, $p := -0.003$, $r := 0.005$, $C_0 := 0.001$, $a := 0.00005$ and $b := 0.00007$.

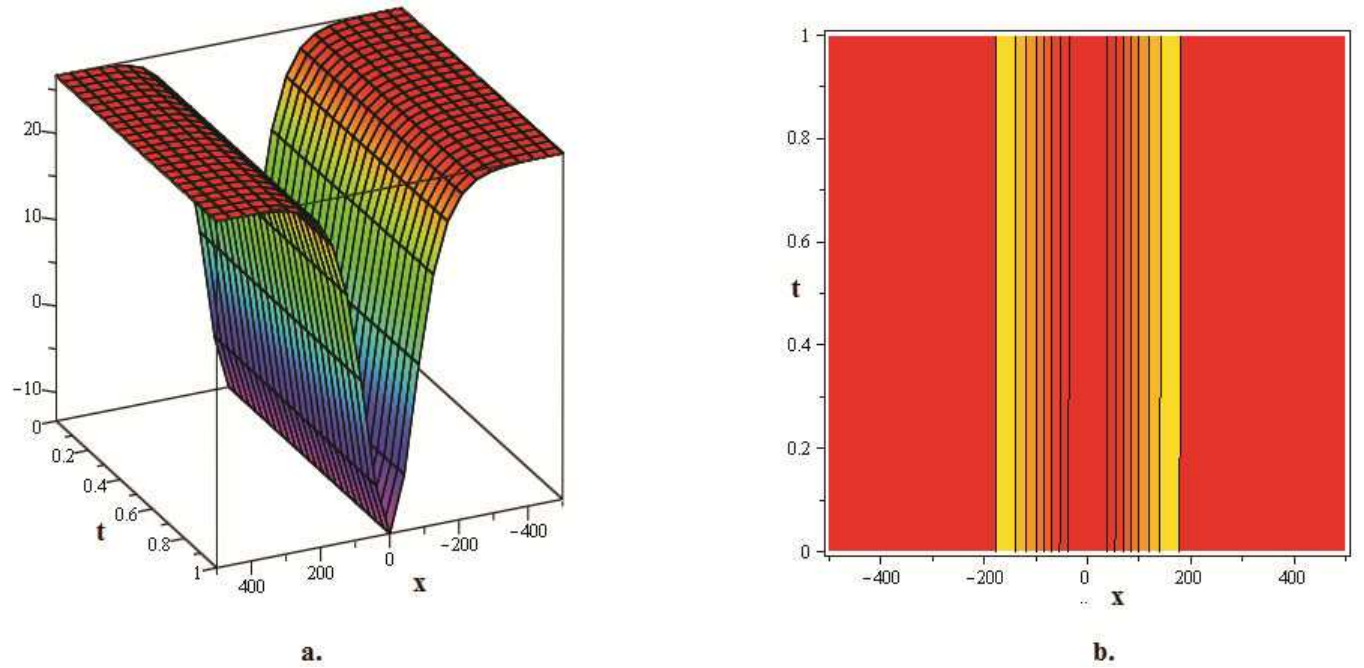


Figure 3. The 3D and contour graphics of the bell-shaped kink soliton solution $u_{1,2,16}$ given in (62) are plotted for $q := 0$, $p := 4$, $r := -1$, $C_2 := 10$, $a := 0.005$ and $b := 0.005$.

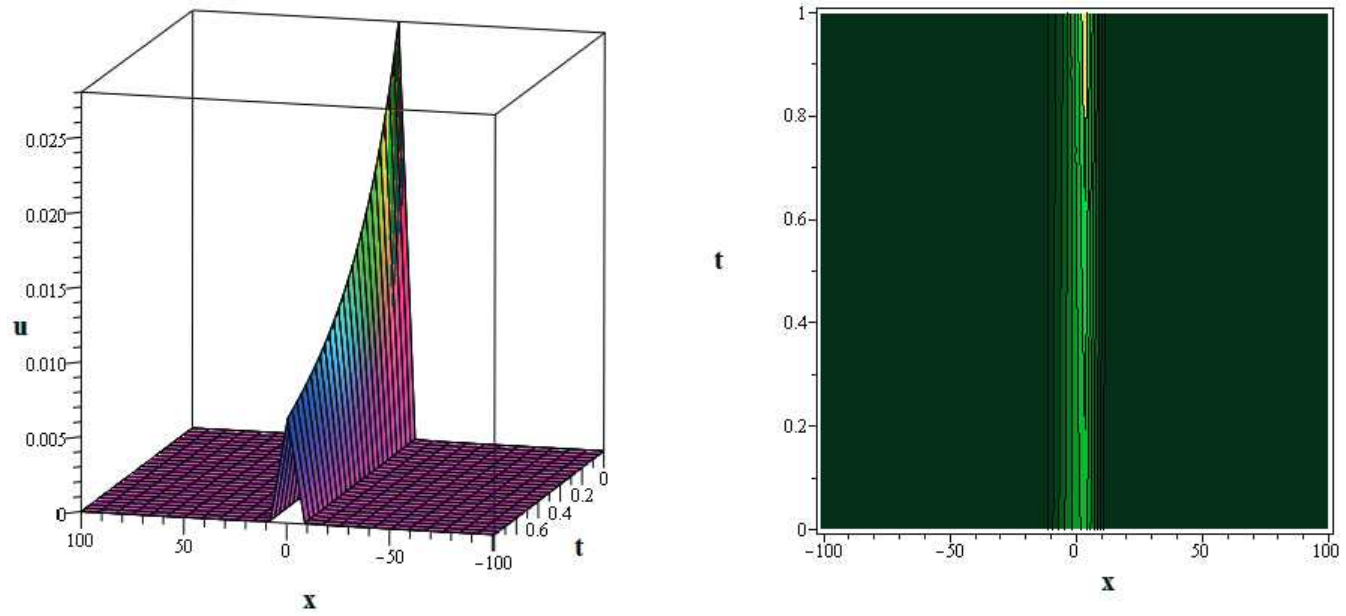


Figure 4. The 3D and contour graphics of the bright kink soliton solution $u_{1,2,31}$ given in (77) are plotted for $q := 2$, $p := 1$, $r := 1$, $C_2 := 70$, $a := 50$ and $b := 100$.

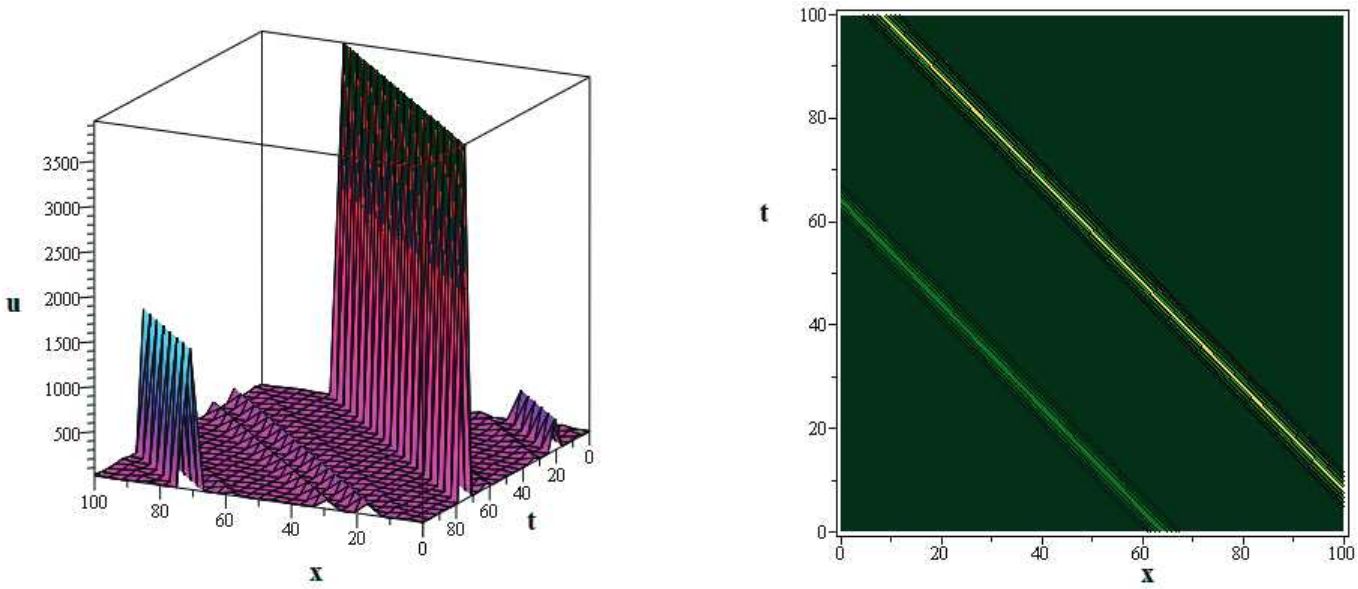


Figure 5. The 3D and contour graphics of the bright kink soliton solution $u_{1,3,11}$ given in (92) are plotted for $q := 0$, $p := 1$, $r := 4$, $C_0 := 10$, $C_1 := 7$, $C_2 := 8$, $a := 2$ and $b := -2$.

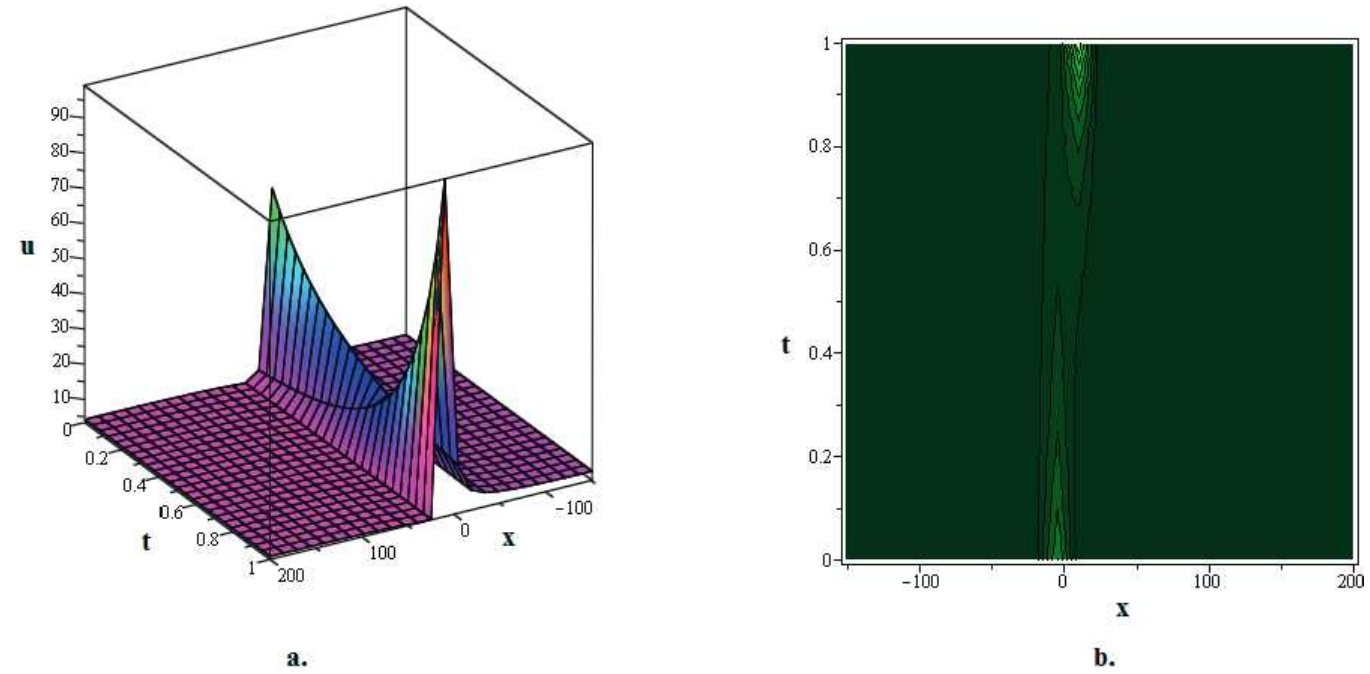


Figure 6. The 3D and contour graphics of the twinning bright kink soliton solution $u_{1,3,27}$ given in (108) are plotted for $q := 0, p := 10, r := -10, C_0 := 5, C_1 := 0.8, C_2 := 0.100, a := 0.0080, b := 0.00110$.

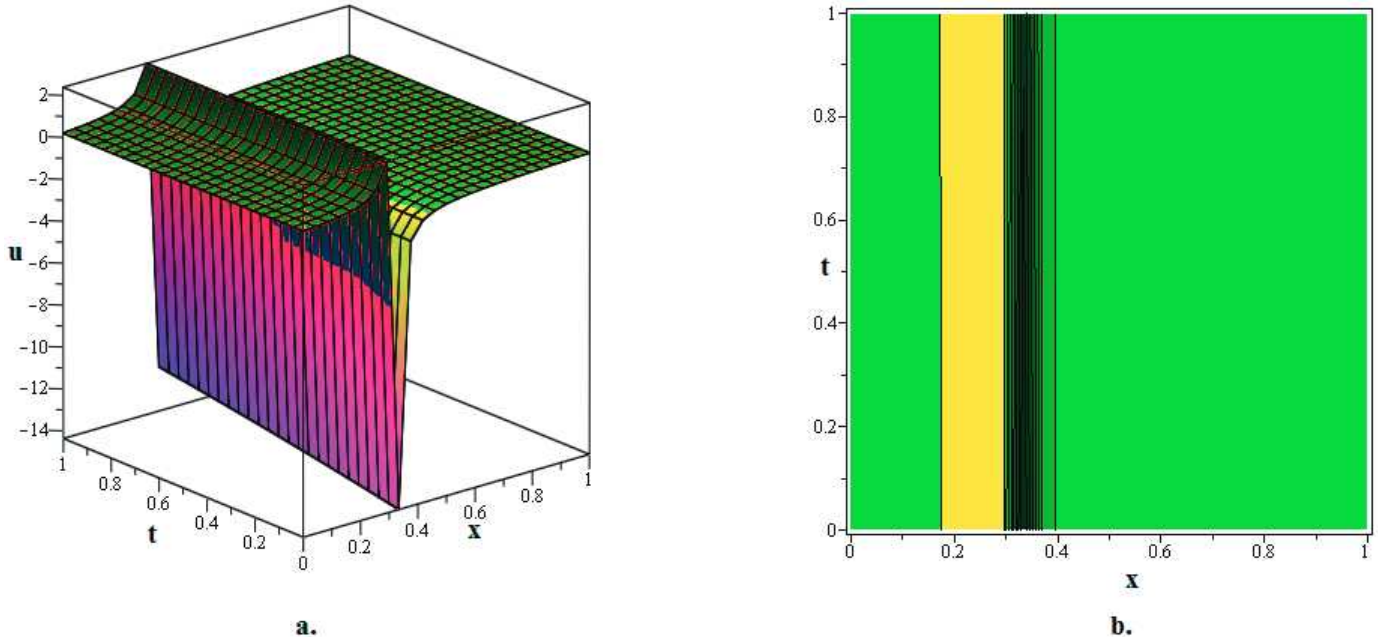


Figure 7. The 3D and contour graphics of the lump-like kink soliton solution $u_{2,1,3}$ given in (127) are plotted for $q := 1$, $p := 2$, $r := 3$, $C_1 := 0.25$ and $\omega := 0.0010$.

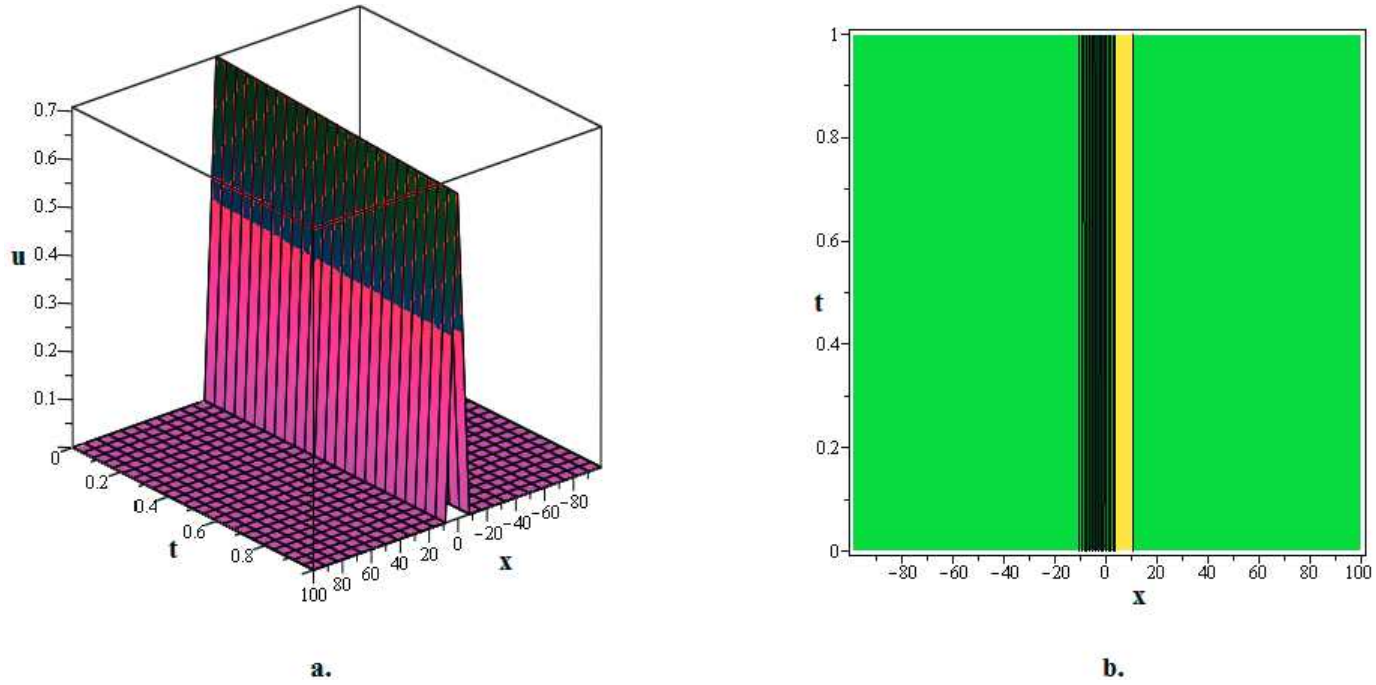


Figure 8. The 3D and contour graphics of the bright kink soliton solution $u_{2,1,10}$ given in (134) are plotted for $q := 4$, $p := 1$, $r := 1$, $C_1 := 0.40$ and $\omega := 0.020$.

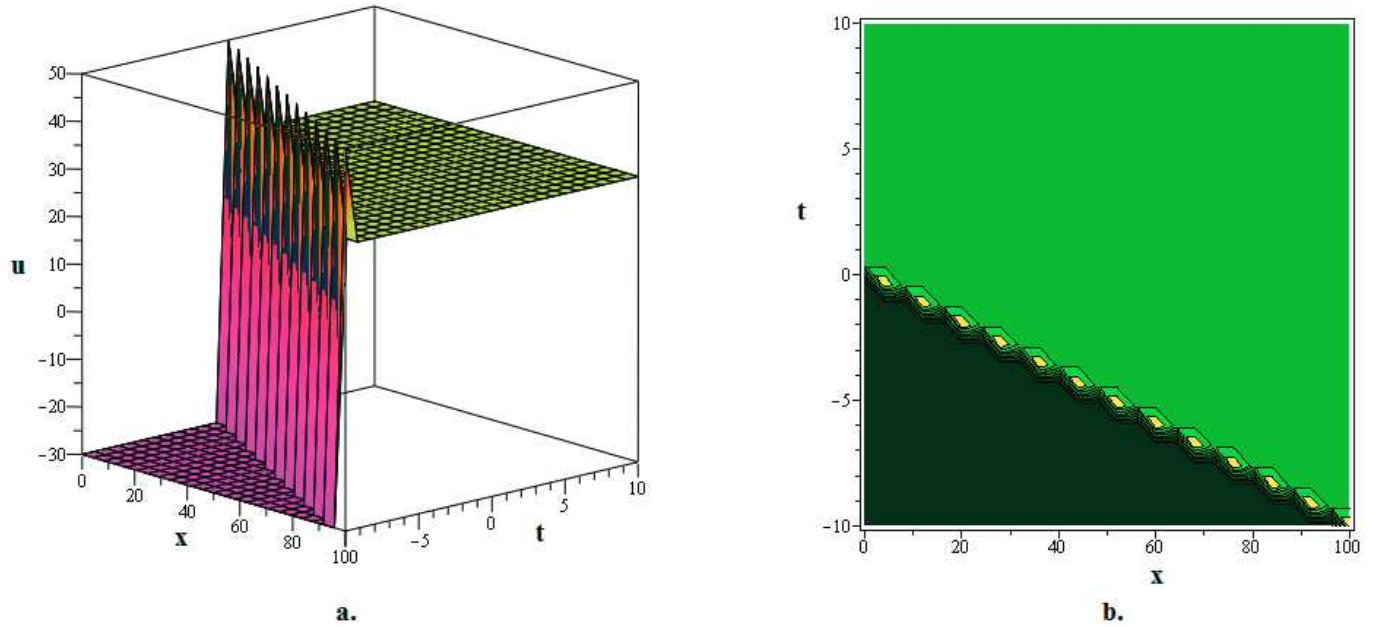


Figure 9. The 3D and contour graphics of the kink soliton solution $u_{2,2,9}$ given in (166) are plotted for $q := 5$, $p := 2$, $r := 2$, $C_{-1} := 40$ and $\omega := 10$.

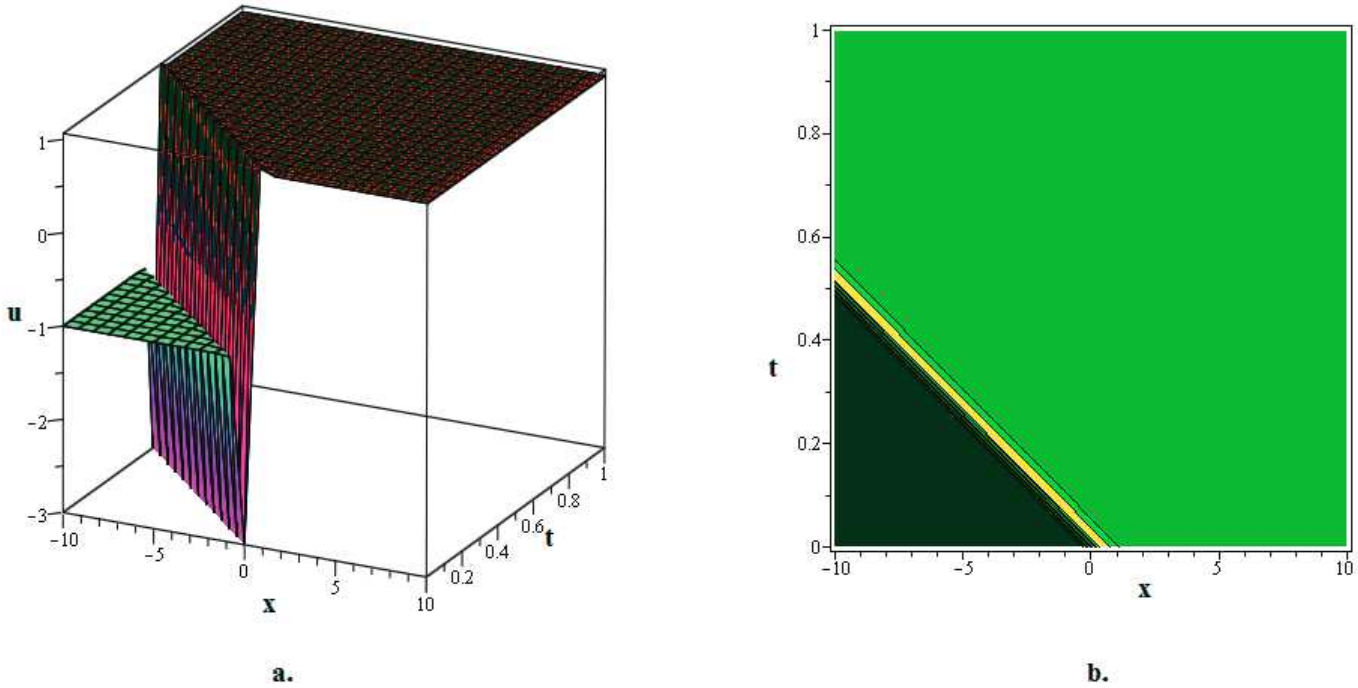


Figure 10. The 3D and contour graphics of the cuspon kink soliton solution $u_{2,2,31}$ given in (188) are plotted for $q := 5$, $p := 10$, $r := 0$, $C_{-1} := 4$, $\omega := 20$, $s := 2$, $\lambda := 5$.

5. Conclusions

We employed the proposed EDAM to construct and analyse the transmission of kink solitons within the quadratic and cubic KGEs, two models that are frequently used in solid-state physics, nonlinear optics and quantum field theory. Several soliton solutions such as the dark kink, hump kink, lump-like kink, bright kink and cuspon kink soliton solutions were obtained in terms of trigonometric, hyperbolic, exponential and rational functions by converting the targeted models into NODEs under the circumstance where solutions are assumed to be of closed form. The contour and 3D plots helped us comprehend the behaviour of the transmitting soliton processes along with various kink soliton solutions. These graphs are directly linked to areas associated with the models discussed. As a result, these results enhanced our comprehension of nonlinear dynamics and temporal evolution processes and have implications for the comprehension of some related physical phenomena. While the EDAM contributed significantly to our comprehension of soliton dynamics and their linking with the targeted models, it is important to state that the proposed method fails when the greatest derivative and the nonlinear terms are not homogenously balanced. Despite this limitation, the present research demonstrates that the method employed in this study is very effective, adaptable and useful for nonlinear models across a variety of scientific fields. Furthermore, the future goal of this study is to delve into the stability analysis of solitons and the incorporation and impact of the fractional derivatives on solitons in the realm of targeted models.

Author Contributions: Conceptualization, S.A. and O.A.A.; methodology, M.A.; software, H.Z.; validation, M.M.A.-S.; formal analysis, S.A.; investigation, O.A.A.; resources, M.A.; data curation, H.Z.; writing—original draft preparation, M.M.A.-S. and S.A.; writing—review and editing, H.Z.; visualization, M.A.; supervision, O.A.A.; project administration, M.M.A.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the scientific research deanship at the University of Hail, Saudi Arabia, through project number (RG-23122).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Whitham, G.B. *Linear and Nonlinear Waves*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
2. Baloch, S.A.; Abbas, M.; Abdullah, F.A.; Rizvi, S.T.; Althobaiti, A.; Seadawy, A.R. Multiple Soliton Solutions of Generalized Reaction Duffing Model Arising in Various Mechanical Systems. *Int. J. Theor. Phys.* **2024**, *63*, 1–15. [CrossRef]
3. Kragh, H. Equation with the many fathers. The Klein-Gordon equation in 1926. *Am. J. Phys.* **1984**, *52*, 1024–1033. [CrossRef]
4. Ablowitz, M.J. *Nonlinear Dispersive Waves: Asymptotic Analysis and Solitons*; Cambridge University Press: Cambridge, UK, 2011; Volume 47.
5. Galehouse, D.C. Geometrical derivation of the Klein-Gordon equation. *Int. J. Theor. Phys.* **1981**, *20*, 457–479. [CrossRef]
6. Schechter, M. The Klein-Gordon equation and scattering theory. *Ann. Phys.* **1976**, *101*, 601–609. [CrossRef]
7. Weder, R.A. Scattering theory for the Klein-Gordon equation. *J. Funct. Anal.* **1978**, *27*, 100–117. [CrossRef]
8. Lundberg, L.E. Spectral and scattering theory for the Klein-Gordon equation. *Commun. Math. Phys.* **1973**, *31*, 243–257. [CrossRef]
9. Tsukanov, V.D. Motion of a Klein-Gordon kink in an external field. *Theor. Math. Phys.* **1990**, *84*, 930–933. [CrossRef]
10. Tariq, K.U.; Bekir, A.; Nisar, S.; Alp, M. Construction of new wave structures and stability analysis for the nonlinear Klein-Gordon equation. *Phys. Scr.* **2024**, *99*, 055220. [CrossRef]
11. Onyenegecha, C.P.; Opara, A.I.; Njoku, I.J.; Udensi, S.C.; Ukewuihe, U.M.; Okereke, C.J.; Omame, A. Analytical solutions of D-dimensional Klein Gordon equation with modified Mobius squared potential. *Results Phys.* **2021**, *25*, 104144. [CrossRef]
12. Belayeh, W.G.; Mussa, Y.O.; Gizaw, A.K. Approximate Analytic Solutions of Two-Dimensional Nonlinear Klein Gordon Equation by Using the Reduced Differential Transform Method. *Math. Probl. Eng.* **2020**, *2020*, 5753974. [CrossRef]
13. Ahmadov, A.I.; Nagiyev, S.M.; Ikot, A.N.; Tarverdiyeva, V.A. Analytical solutions for the Klein Gordon equation with combined exponential type and ring-shaped potentials. *Sci. Rep.* **2024**, *14*, 5527. [CrossRef] [PubMed]
14. Abdeljabbar, A.; Roshid, H.O.; Aldurayhim, A. Bright, dark, and rogue wave soliton solutions of the quadratic nonlinear Klein Gordon equation. *Symmetry* **2022**, *14*, 1223. [CrossRef]
15. Huang, D.J.; Zhang, H.Q. The extended first kind elliptic sub-equation method and its application to the generalized reaction Duffing model. *Phys. Lett.* **2005**, *344*, 229–237. [CrossRef]
16. Akter, J.; Akbar, M.A. Exact solutions to the Benney-Luke equation and the Phi-4 equations by using modified simple equation method. *Results Phys.* **2015**, *5*, 125–130. [CrossRef]
17. Ebaid, A. Exact solutions for the generalized Klein-Gordon equation via a transformation and Exp-function method and comparison with Adomians method. *J. Comput. Appl. Math.* **2009**, *223*, 278–290. [CrossRef]
18. Kudryavtsev, A.E. *Solitonlike Solutions for a Higgs Scalar Field*; Institute of Theoretical and Experimental Physics: Moscow, Russia, 1975.
19. Strauss, W.; Vazquez, L. Numerical solution of a nonlinear Klein-Gordon equation. *J. Comput. Phys.* **1978**, *28*, 271–278. [CrossRef]
20. Jiménez, S.; Vzquez, L. Analysis of four numerical schemes for a nonlinear Klein-Gordon equation. *Appl. Math. Comput.* **1990**, *35*, 61–94. [CrossRef]
21. Dehghan, M.; Shokri, A. Numerical solution of the nonlinear Klein-Gordon equation using radial basis functions. *J. Comput. Appl. Math.* **2009**, *230*, 400–410. [CrossRef]
22. Sarboland, M.; Aminataei, A. Numerical solution of the nonlinear Klein-Gordon equation using multiquadric quasi-interpolation scheme. *Univers. J. Appl. Math.* **2015**, *3*, 40–49. [CrossRef]
23. Alsisi, A. Analytical and numerical solutions to the Klein Gordon model with cubic nonlinearity. *Alex. Eng. J.* **2024**, *99*, 31–37. [CrossRef]
24. Nirmala, A.N.; Kumbinarasaiah, S. Numerical solution of nonlinear Hunter-Saxton equation, Benjamin-Bona Mahony equation, and Klein-Gordon equation using Hosoya polynomial method. *Results Control. Optim.* **2024**, *14*, 100388. [CrossRef]
25. Denton, P.B. Techniques for solving static Klein-Gordon equation with self-interaction $\lambda\phi 4$ and arbitrary spherical source terms. *Phys. Lett. B* **2024**, *855*, 138860. [CrossRef]
26. Bao, W.; Lu, Y.; Zhang, Z. Convergence Rates in the Nonrelativistic Limit of the Cubic Klein Gordon Equation. *Siam J. Math. Anal.* **2024**, *56*, 6822–6860. [CrossRef]
27. Ali, I.; Ahmad, I. Applications of the nonlinear Klein/Sinh-Gordon equations in modern physics: A numerical study. *Math. Model. Control.* **2024**, *4*, 361–373. [CrossRef]
28. Sadiya, U.; Inc, M.; Arefin, M.A.; Uddin, M.H. Consistent travelling waves solutions to the ar time fractional Klein-Gordon and Sine-Gordon equations through extended tanh-function approach. *J. Taibah Univ. Sci.* **2022**, *16*, 594–607. [CrossRef]
29. Abdel-Salam, E.A.B.; Yousif, E.A. Solution of nonlinear space-time fractional differential equations using the fractional Riccati expansion method. *Math. Probl. Eng.* **2013**, *2013*, 846283. [CrossRef]
30. Aldandani, M.; Altherwi, A.A.; Abushaega, M.M. Propagation patterns of dromion and other solitons in nonlinear Phi-Four (f 4) equation. *Aims Math.* **2024**, *9*, 19786–19811. [CrossRef]

31. Yasmin, H.; Aljahdaly, N.H.; Saeed, A.M.; Shah, R. Investigating symmetric soliton solutions for the fractional coupled konno-onno system using improved versions of a novel analytical technique. *Mathematics* **2023**, *11*, 2686. [CrossRef]
32. Bilal, M.; Iqbal, J.; Ali, R.; Awwad, F.A.; Ismail, E.A. Exploring families of solitary wave solutions for the fractional coupled Higgs system using modified extended direct algebraic method. *Fractal Fract.* **2023**, *7*, 653. [CrossRef]
33. Rizvi, S.T.; Ghafoor, S.; Seadawy, A.R.; Arnous, A.H.; AL, H. Exploration of solitons and analytical solutions by sub-ODE and variational integrators to Klein-Gordon model. *Aims Math.* **2024**, *9*, 21144–21176. [CrossRef]
34. Díaz Palencia, J.L.; Roa González, J.; Sánchez Sánchez, A. Study of Solutions for a Degenerate Reaction Equation with a High Order Operator and Advection. *Mathematics* **2022**, *10*, 1729. [CrossRef]
35. Ullah, I.; Shah, K.; Barak, S.; Abdeljawad, T. Pioneering the Plethora of Soliton for the (3+1)-Dimensional Fractional Heisenberg Ferromagnetic Spin Chain Equation. *Phys. Scr.* **2024**, *99*, 095229. [CrossRef]
36. Ali, R.; Barak, S.; Altalbe, A. Analytical study of soliton dynamics in the realm of fractional extended shallow water wave equations. *Phys. Scr.* **2024**, *99*, 065235. [CrossRef]
37. Ullah, I.; Shah, K.; Abdeljawad, T.; Alam, M.M.; Hendy, A.S.; Barak, S. Dynamics Behaviours of Kink Solitons in Conformable Kolmogorov-Petrovskii-Piskunov Equation. *Qual. Theory Dyn. Syst.* **2024**, *23* (Suppl. 1), 268. [CrossRef]
38. Kudryashov, N.A. Seven common errors in finding exact solutions of nonlinear differential equations. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, *14*, 3507–3529. [CrossRef]
39. Navickas, Z.; Ragulskis, M. Comments on A new algorithm for automatic computation of solitary wave solutions to nonlinear partial differential equations based on the Exp-function method. *Appl. Math. Comput.* **2014**, *243*, 419–425. [CrossRef]
40. Antonova, A.O.; Kudryashov, N.A. Generalization of the simplest equation method for nonlinear non-autonomous differential equations. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 4037–4041. [CrossRef]
41. Xiao, Y.; Barak, S.; Hleili, M.; Shah, K. Exploring the dynamical behaviour of optical solitons in integrable kairat-II and kairat-X equations. *Phys. Scr.* **2024**, *99*, 095261. [CrossRef]
42. Ma, W.X.; Fuchssteiner, B. Explicit and exact solutions to a Kolmogorov-Petrovskii-Piskunov equation. *Int. J. -Non-Linear Mech.* **1996**, *31*, 329–338. [CrossRef]
43. Ma, W.X.; Lee, J.H. A transformed rational function method and exact solutions to the 3+ 1 dimensional Jimbo-Miwa equation. *Chaos Solitons Fractals* **2009**, *42*, 1356–1363. [CrossRef]
44. Navickas, Z.; Marcinkevicius, R.; Telksniene, I.; Telksnys, T.; Ragulskis, M. Structural stability of the hepatitis C model with the proliferation of infected and uninfected hepatocytes. *Math. Comput. Model. Dyn. Syst.* **2024**, *30*, 51–72. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Intelligent Sensor Software for Robust and Energy-Sustainable Decision-Making in Welding of Steel Reinforcement for Concrete

Javier Ferreiro-Cabello [1], Francisco Javier Martinez-de-Pison [2,*], Esteban Fraile-Garcia [1], Alpha Pernia-Espinoza [2] and Jose Divasón [3]

[1] SCoDIP Group, Department of Mechanical Engineering, University of La Rioja, 26006 Logroño, Spain; javier.ferreiro@unirioja.es (J.F.-C.); esteban.fraile@unirioja.es (E.F.-G.)
[2] SCoTIC, Scientific Computation & Technological Innovation Center, University of La Rioja, 26006 Logroño, Spain; alpha.pernia@unirioja.es
[3] PSYCOTRIP Group, Department of Mathematics and Computation, University of La Rioja, 26006 Logroño, Spain; jose.divason@unirioja.es
* Correspondence: fjmartin@unirioja.es

Abstract: In today's industrial landscape, optimizing energy consumption, reducing production times, and maintaining quality standards are critical challenges, particularly in energy-intensive processes like resistance spot welding (RSW). This study introduces an intelligent decision support system designed to optimize the RSW process for steel reinforcement bars. By creating robust machine learning models trained on limited datasets, the system generates interactive heat maps that provide real-time guidance to production engineers or intelligent systems, enabling dynamic adaptation to changing conditions and external factors such as fluctuating energy costs. These heat maps serve as a flexible and intuitive tool for identifying robust operational points that balance quality, energy efficiency, and productivity. The proposed methodology advances decision-making in welding processes by combining robust predictive modeling with innovative visualization techniques, offering a versatile solution for multiobjective optimization in real-world industrial applications.

Keywords: energy-sustainable decision; resistance spot welding; intelligent welding; machine learning; artificial intelligence (AI)

1. Introduction

Welding is a fundamental process in manufacturing [1], critical for constructing durable and high-strength structures. With the advent of Industry 4.0, many manufacturing companies have successfully implemented automated systems for monitoring and controlling welding processes, effectively closing the loop from creation to quality control [2,3]. These advancements have leveraged historical data from a variety of processes to develop artificial intelligence (AI) techniques aimed at enhancing productivity and minimizing production costs. Within this context, machine learning (ML) has become an essential tool, enabling the creation of intelligent models that can predict outcomes, define setpoints, plan, control, and even identify defects in real time. Despite the growing use of AI in Industry 4.0, sustainability concerns—particularly energy efficiency and environmental impact—remain a secondary consideration [4]. Consequently, many companies still rely on manual or semi-automated welding, highlighting the need to fully automate these processes to achieve both sustainability and productivity gains.

Among the various welding techniques available, resistance spot welding (RSW) is particularly economical and productive for joining steel in reinforced concrete structures, as it forms a strong and durable hidden joint within a short processing time [5]. RSW is widely preferred for high-production-rate applications, but it poses technical challenges due to the need for precise control of parameters like current, pressure, and time. Additionally, the high energy demands of RSW increase operational costs and impact environmental sustainability, making the optimization of energy consumption while maintaining quality and productivity a key priority for manufacturers [6].

In this study, we focus on assembly welding for reinforced steel used in concrete structures, which is critical for ensuring structural integrity and durability. This RSW process involves applying pressure and electrical current to overlapping steel bars, forming a solid-state bond. Typical configurations require welding bars of different diameters to ensure joint integrity. The equipment used in this study, PRAXAIR MPH Digital Pneumatic, allows precise parameter control, but optimization is critical to balance quality, cost, and sustainability. Two key indicators, deformation limitation and strength, were selected to evaluate the process. Deformation limitation helps mitigate damage from residual stresses, while strength ensures overall reliability. These metrics are essential for meeting structural standards.

The aim of this work is to develop an intelligent decision support system to optimize the RSW process for steel reinforcement welding. This system integrates machine learning models trained on experimental data to generate intelligent heat maps. These heat maps provide real-time guidance, enabling operators to identify robust operational points that satisfy constraints related to quality, energy consumption, and production time. The main contributions of this study are twofold:

- The development of machine learning models to estimate optimal welding parameters despite the small dataset available [7]. The proposed machine learning approach accurately predicts ideal parameters and remains robust under varying conditions.
- The introduction of intelligent heat maps as a visual and flexible tool that enables production engineers to address the challenge of balancing welding quality, production time, and energy savings while adapting to dynamic constraints and external factors like energy costs. Unlike traditional optimization methods, heat maps allow engineers to dynamically identify optimal operating points and robust regions where process parameters remain stable. These maps can also be integrated into intelligent systems for automated decision-making, providing a versatile solution for multiobjective optimization in welding and similar industrial processes.

In Section 2, Section 2.1 presents related works on RSW. Section 2.2 outlines the main issues encountered when working with small datasets, as well as background on ML techniques relevant to this research. Section 2.3 provides a detailed description of the dataset underlying this study, including the experimental procedures and methodologies employed in its collection and development. Section 2.4 explains the proposed approach. Section 3 presents the experimental results and a discussion of these findings. Finally, Section 4 concludes this paper by summarizing the main contributions and suggesting future research directions.

2. Materials and Methods

2.1. The Resistance Spot Welding Process

Artificial intelligence (AI) has significantly advanced the understanding and optimization of the resistance spot welding (RSW) process [8,9]. Techniques such as neural networks [3,10] and adaptive neuro-fuzzy inference systems (ANFISs) [9] have been em-

ployed to predict weld strength in steel sheets with high accuracy. Additionally, carbon emissions from welding processes have been modeled to assess environmental impacts [11].

The recent surge in manufacturing data has facilitated the application of machine learning (ML) in welding quality prediction. For instance, logistic regression models have been utilized to predict defects in electric resistance welded tubes, yielding satisfactory results and enabling the estimation of safe operating ranges for process variables [12]. Support vector regression has also been applied to predict quality in electron beam welding [13]. Furthermore, feature engineering-based ML approaches have been proposed for quality prediction in RSW of steel sheets, combining engineering and data science perspectives to provide insights into welding data and account for the dynamic nature of RSW due to factors like cooling time and electrode wear [14].

In recent years, there has been a growing interest in integrating AI with sensor data to enhance RSW processes. For example, a study developed a machine learning tool to predict the effect of electrode wear on weld quality, achieving an accuracy of 90% [15]. Another research utilized a convolutional neural network (CNN) combined with a long short-term memory (LSTM) network and an attention mechanism to detect welding quality online, achieving a detection accuracy of 98.5% [16]. Additionally, AI-driven interpretation of ultrasonic data has been explored for real-time quality assessment in RSW, highlighting the potential of adaptive welding using ultrasonic process monitoring backed by AI-based data interpretation [17].

Despite these advancements, the application of AI in the RSW of reinforcement bars (rebar) remains underexplored. Moreover, existing studies primarily focus on predictive modeling and do not incorporate visualization techniques to facilitate sustainable decision-making beyond the outputs of ML models. This gap underscores the need for research that integrates AI-driven predictive models with intuitive visualization tools to support informed and sustainable decisions in RSW processes.

2.2. Machine Learning-Based Strategy for Small Datasets

Automating complex industrial processes, such as resistance spot welding (RSW), often encounters the challenge of limited data availability during initial implementation phases. In many cases, comprehensive process data are scarce, necessitating the acquisition of information through meticulously designed experiments that span the entire operational state space. Typically, these datasets consist of process conditions as inputs and corresponding weld quality parameters as outputs. Machine learning (ML) models can be developed from this data to predict optimal setpoints based on predefined requirements. However, constructing and deploying these models is particularly challenging when relying on experimental datasets, primarily due to their limited size and scope.

The high economic and time costs associated with experimental data collection often result in datasets that are insufficiently large to capture the full spectrum of process variability. This limitation can lead to the development of overfitted models—models that have learned the training data too well and, consequently, fail to generalize to new, unseen conditions [18]. Overfitting is a common issue in models with high complexity, such as deep neural networks with numerous parameters or extensive regression trees, where the model effectively 'memorizes' the training data rather than learning the underlying patterns.

Conversely, employing less flexible algorithms, such as simple linear regression, may yield models with inadequate accuracy, rendering them unsuitable for capturing the complexities of the process. Therefore, selecting an appropriate algorithm and meticulously tuning its parameters are critical steps in developing accurate ML models, especially when dealing with small datasets [19]. Robustness is a key consideration to ensure that the

setpoints derived from these models remain reliable and stable, even in the presence of input perturbations.

Recent studies have explored various strategies to address the challenges associated with small datasets in ML applications. Techniques such as data augmentation, transfer learning, and the use of ensemble methods have been investigated to enhance model performance and generalization capabilities [20]. Additionally, the integration of domain knowledge into the modeling process has been shown to improve the reliability of predictions, particularly in industrial applications where expert insights can compensate for limited data [21].

In summary, effectively managing the constraints of small datasets requires a balanced approach that combines careful algorithm selection, parameter tuning, and the incorporation of domain expertise. Such strategies are essential to develop robust and accurate ML models capable of supporting automation in industrial processes like RSW.

2.2.1. The Strategy in this Study

The primary task in developing a machine learning (ML)-based decision support system involves testing various models and optimizing their hyperparameters. When working with large datasets and numerous hyperparameters, it is common practice to limit the number of models tested and use grid search for hyperparameter optimization due to the substantial computational cost of each experiment.

In this study, however, the dataset is relatively small (298 rows with five input variables and two target variables). This smaller dataset size has three key implications:

- There is an elevated risk of overfitting.
- The limited dataset size necessitates additional experimental trials.
- Feature selection is unnecessary, as there are only five input variables.

To mitigate the risk of overfitting in training, we employ a nested cross-validation approach. Given the small dataset size, this method is repeated multiple times to enhance the robustness of the results. Additionally, hyperparameter optimization is performed using Bayesian optimization rather than the more conventional grid search, which improves efficiency and allows for more refined parameter tuning.

Beyond traditional models like random forest (RF) and multilayer perceptron (MLP), we also evaluate a powerful AutoML tool, Autogluon, to further streamline and enhance the modeling process.

2.2.2. Bayesian Optimization

Bayesian optimization [22] is a popular algorithm for hyperparameter optimization in machine learning models. While alternative methods like evolutionary algorithms [23] and particle swarm optimization [24] exist, they are typically more computationally expensive and often require numerous evaluations.

In Bayesian optimization, a probabilistic model of the objective function is iteratively updated to guide the selection of hyperparameters. This process involves evaluating the model with selected hyperparameters, updating the probabilistic model based on performance, and repeating until an optimal set of hyperparameters is identified. By leveraging past iterations, Bayesian optimization focuses the search on promising regions of the hyperparameter space.

Compared with standard methods like grid and random search, Bayesian optimization provides a more efficient exploration of the hyperparameter space. Unlike grid search, which examines a fixed set of points and may miss key regions, Bayesian optimization uses its probabilistic model to adaptively explore high-potential areas, even in the presence of non-convex or noisy objective functions. This adaptability helps it avoid local optima and

manage evaluation noise more effectively. However, since Bayesian optimization relies on iterative learning from previous evaluations, it tends to have higher computational costs per iteration.

2.2.3. Nested Cross-Validation

Cross-validation (CV) is essential when developing machine learning models, as it provides a reliable estimate of model performance on unseen data and helps mitigate overfitting. The core idea of cross-validation is to split the dataset into training and testing sets multiple times. The model is trained on each training set and evaluated on the corresponding testing set. Repeating this process across different data splits yields a more robust estimate of the model's performance. The most common method, k-fold cross-validation, divides the dataset into k equally sized folds. The model is trained on $k-1$ folds and tested on the remaining fold. This process is repeated k times, so each fold is used once for testing.

The development of AI/ML systems is prone to errors, especially related to *model biases* and *dataset shifts* [25]. Model biases often stem from improper training procedures, leading to skewed inference results. Dataset shifts, on the other hand, arise from mismatches between training and testing data distributions. These issues are particularly challenging with small datasets, where the limited data volume amplifies the risk of overfitting and makes mistakes more likely.

Nested cross-validation is a technique designed to evaluate both the generalization error of a machine learning model and its hyperparameters. It is an extension of the k-fold cross-validation approach and is particularly beneficial when working with small datasets and models with numerous hyperparameters to tune [26]. Algorithm 1 presents a pseudo-code version of the process.

Algorithm 1 Pseudo-code for nested cross-validation

```
Split the data into k outer folds.
for each outer fold do
    Split the outer training set into k − 1 inner folds and one inner validation fold.
    for each combination of hyperparameters do
        Train the model on the k − 1 inner folds.
        Evaluate the model on the inner validation fold.
    end for
    Select the hyperparameters that yielded the best performance on the inner validation folds.
    Retrain the model on the full outer training set (using all inner folds) with the selected hyperparameters.
    Evaluate the model on the outer test fold.
end for
return Average performance of the model across all outer test folds.
```

The key concept in nested cross-validation is the use of two *nested loops* of cross-validation:

1. Outer loop: The outer loop splits the data into training and testing sets, similar to standard cross-validation. However, rather than using the same hyperparameters across all folds, a unique set of hyperparameters is selected for each fold.
2. Inner loop: The inner loop is dedicated to hyperparameter selection. It performs cross-validation on the training set to assess the model's performance with various hyperparameter configurations. The set of hyperparameters that performs best in the inner loop is then used to train the model on the full training set in the outer loop.

Considering the scalability challenges of nested cross-validation with large datasets, it is noted that for fully automated systems capable of generating extensive data, stan-

dard cross-validation may provide a more computationally efficient alternative while maintaining robust performance metrics, a point explicitly addressed in this study.

2.3. The RSW Process Dataset

This study utilizes a dataset on the strength of resistance spot welding (RSW), created through controlled welding experiments and described in detail by Ferreiro-Cabello et al. [7]. The following lines summarize the process of creating the dataset and expand on its main features, incorporating additional context and methodology details.

The dataset includes welding tests on rebar of structural steel B500-S, specifically in diameters commonly used in reinforced concrete structures: 8, 10, 12, and 16 mm.

This study focuses on assembly welding, specifically the fabrication of reinforcement steel (ferralla), which does not have a structural purpose.

These diameters were selected to represent a range of joint thicknesses between 16 mm and 32 mm, as shown in Table 1. For each combination, the test configuration consisted of welding three bars of diameter *B* to one of diameter *A*, as illustrated in Figure 1. This configuration ensured that the dataset would include the most frequently used joint dimensions in the production of electro-welded meshes.

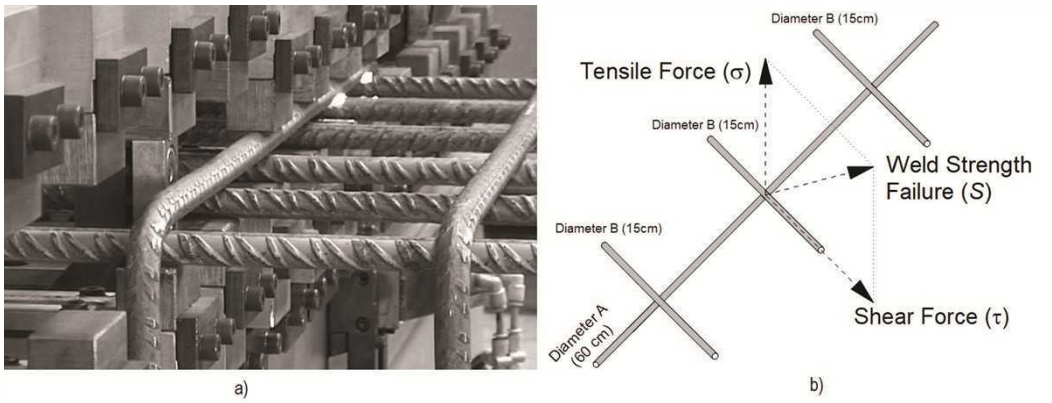


Figure 1. (**a**) Fusion welding of reinforcement bars. (**b**) Test bar configuration and weld strength failure (*S*).

Table 1. Combinations of diameters for joint Thickness.

Diameter A (mm)	Diameter B (mm)	Joint Thickness (mm)
8	8	16
8	10	18
8	12	20
10	12	22
12	12	24
10	16	26
12	16	28
16	16	32

The welding experiments were conducted using a PRAXAIR MPH Digital Pneumatic welder, equipped with a maximum power output of 50 kVA. This equipment enabled precise adjustments of key welding parameters, including time (t), current intensity (I), and pressure (p). Welding time was controlled by a digital system, which allowed for cycles of up to 2 s, split into different active time percentages to match the joint thickness. Three

discrete values of pressure (5, 6, and 7 bars) were used, and current intensity varied from 40% to 80% of the maximum power, as shown in Figure 2.

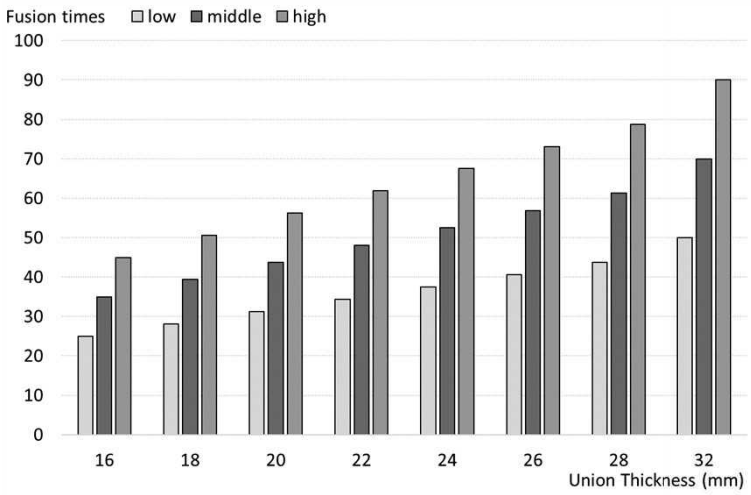


Figure 2. Active welding time according to joint thickness, represented on a 0–100 scale.

A total of 2160 welds were performed, corresponding to 360 combinations of *t-I-p* for each joint thickness. For each parameter combination, two sets of six welded bars were prepared to assess both strength and deformation characteristics. The welding parameter commands are shown in Table 2.

Table 2. The welding parameter commands for the A-t-p equipment.

Joint Thickness (mm)	Equipment Parameters		No. Specimen
	Phase-1 (Contact)	Phase-2 (Welding)	
16	A*0.5 t*1/3 5_6_7	40–50–60–70–80% 25–35–45 5_6_7	45
18	A*0.5 t*1/3 5_6_7	40–50–60–70–80% 29–40–51 5_6_7	45
20	A*0.5 t*1/3 5_6_7	40–50–60–70–80% 32–44–57 5_6_7	45
22	A*0.5 t*1/3 5_6_7	40–50–60–70–80% 35–49–62 5_6_7	45
24	A*0.5 t*1/3 5_6_7	40–50–60–70–80% 38–53–68 5_6_7	45
26	A*0.5 t*1/3 5_6_7	40–50–60–70–80% 41–57–74 5_6_7	45
28	A*0.5 t*1/3 5_6_7	40–50–60–70–80% 44–62–79 5_6_7	45
32	A*0.5 t*1/3 5_6_7	40–50–60–70–80% 50–70–90 5_6_7	45

To evaluate joint deformation, the contact area thickness was measured after welding. This measurement was conducted using a Palmer micrometer to ensure non-destructive sampling accuracy. The deformation (D) was calculated as

$$D = 1 - \frac{\frac{\sum_{n=1}^{6} e_n}{6}}{A + B}\% \tag{1}$$

where e_n represents the individual thickness measurements. An acceptable weld was defined by minimum values of $S = 3500$ N and $D = 0.85(A + B)$, filtering out any outliers due to test errors.

In addition to thickness measurements, weld strength was tested using destructive shear (τ) and tensile (σ) tests with a Servosis ME-420/20 machine. For each combination, the mean values of shear ($\bar{\tau}$) and tensile ($\bar{\sigma}$) strength were calculated. The combined strength (S) was computed using

$$S = \sqrt{\bar{\sigma}^2 + \bar{\tau}^2} \tag{2}$$

as illustrated in Figure 1. This comprehensive testing provided reliable data on joint performance across various conditions.

Finally, the dataset consists of five input variables (diameters A and B, t, I, and p) and two output variables (S and D). Extreme outliers were removed, resulting in a final dataset of 298 rows, each representing a unique welding condition.

Figure 3 shows the distribution of joint thickness across experiments, confirming the dataset's balanced coverage for different joint thicknesses. This balance ensures that the dataset represents a diverse range of welding conditions applicable to industrial RSW processes.

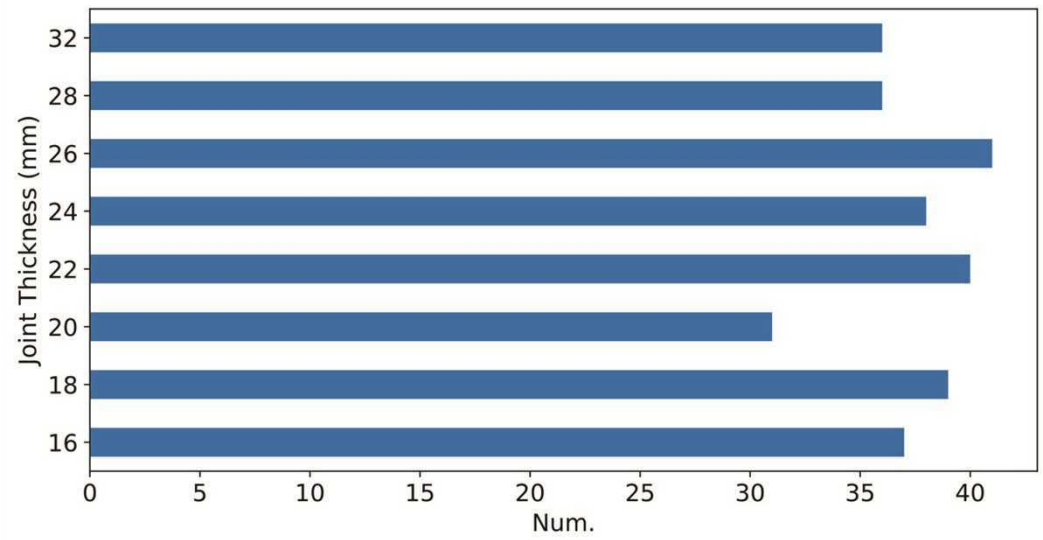


Figure 3. Distribution of experiments according to the total joint thickness, *thickness* $= A + B$.

2.4. Methodology

The proposed methodology involves a series of steps designed to optimize the machine learning (ML) models, ensure their robustness, and facilitate sustainable decision-making. The steps are as follows:

1. **Selection of ML algorithms**: Choosing the most appropriate algorithms based on the nature of the problem and the limited dataset size.
2. **Model training and evaluation**: Implementing nested cross-validation combined with Bayesian optimization for hyperparameter tuning to enhance model performance and prevent overfitting.

3. **Robustness analysis**: Assessing model robustness to ensure reliable performance across various data conditions.
4. **Heat map generation**: Developing interactive heat maps to support sustainable decision-making by visualizing optimal operational regions.

2.4.1. Selection of the ML Algorithms

The main goal was to develop machine learning (ML) models capable of making robust and accurate predictions of S (strength) in Newtons (N) and D (deformation) as a percentage (%). As described in Section 2.3, the dataset consists of 298 experimental cases with five input variables representing each test configuration: *A-B-t-I-p*, along with two output target variables: S (strength) and D (deformation).

To enhance model training, the distribution of the target variables was improved by applying a square root transformation. This adjustment aimed to reduce skewness and improve the stability of the predictions. Finally, the entire dataset was normalized to ensure uniformity in the input scale across all variables.

The following algorithms were selected for experimentation due to their suitability for small datasets: ridge regression (`Ridge`), kernel ridge regression (`KRidge`), decision tree regressor (`DTR`), random forest regressor (`RF`), support vector regression (`SVR`), and multilayer perceptron (`MLP`). All of these algorithms are available in the `scikit-learn` Python library and are well suited for the dataset size and structure. Table 3 summarizes the hyperparameter optimization ranges for each model.

In addition to these established ML models, the `AutoGluon` AutoML tool was also included in the experiments to explore potential performance gains through automated model selection and tuning. Other algorithms, such as Lasso regression, K-Nearest Neighbors regression, and Gaussian process regression, were excluded from the final experiments due to suboptimal performance with this dataset.

Table 3. ML models and ranges of the optimization of each hyperparameter.

Algorithm	Tuning Hyperparameters
Ridge	$alpha = 10^{[-7,3]}$
Kridge	$alpha = 10^{[-7,0]}$
	$gamma = 10^{[-7,0]}$
	kernel = rbf
DTR	$max_depth \in [1, 30]$
	$min_imp \in [0, 1]$
RF	$max_depth \in [1, 30]$
	$n_estimators \in [1, 50]$
	$min_imp \in [0, 1]$
SVR	$C = 10^{[-7,3]}$
	$gamma = 10^{[-7,0]}$
	kernel = rbf
MLP	$hidden_size \in [1, 25]$
	$alpha = 10^{[-7,0]}$
	activation = logistic

2.4.2. Training and Evaluation

Once the models were selected, we proceeded with their training and evaluation. Given the small dataset size, which increases the risk of overfitting but allows for multiple experimental trials, we incorporated two advanced techniques: Bayesian optimization and nested cross-validation. Although widely recognized, these techniques are not typically

part of standard AI model development, where simpler approaches such as grid search and basic cross-validation are more common.

Bayesian optimization was particularly suited to this problem due to its balance between performance and computational efficiency, making it ideal for the dataset size and the need for a robust cross-validation scheme.

For model validation, we employed a repeated *k*-fold nested cross-validation to ensure both accurate hyperparameter tuning and reliable performance estimation. Specifically, we performed a 10-times repeated 10-fold nested cross-validation. In each repetition, the outer loop ran 10 times, creating 10 unique test sets. For each iteration of the outer loop (i.e., with a fixed test set), we conducted a 9-fold cross-validation within the inner loop, using Bayesian optimization with 200 iterations to fine-tune the hyperparameters. In the inner loop, the non-test data were divided into 9 parts: 8 for training and 1 for validation. This inner cross-validation process was repeated 9 times, and the best hyperparameters identified were used to retrain the model on the combined training and validation sets before evaluating it on the test set. Predictions on the test set were stored for subsequent analysis. Figure 4 illustrates an example of the inner loop in one iteration of the outer loop. This process resulted in a total of 100 different training, validation, and testing partitions.

VAL	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TEST
TRAIN	VAL	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TEST
TRAIN	TRAIN	VAL	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TEST
TRAIN	TRAIN	TRAIN	VAL	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TEST
TRAIN	TRAIN	TRAIN	TRAIN	VAL	TRAIN	TRAIN	TRAIN	TRAIN	TEST
TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	VAL	TRAIN	TRAIN	TRAIN	TEST
TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	VAL	TRAIN	TRAIN	TEST
TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	VAL	TRAIN	TEST
TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	TRAIN	VAL	TEST
TRAIN OPTIMIZED MODEL									TEST

Figure 4. Example of the inner loop within one iteration of the nested cross-validation process. The model is optimized using Bayesian optimization over 9 folds, while the test set remains untouched during this stage. The final model is evaluated on the test fold.

Overall, this nested cross-validation process was repeated 10 times, leading to 900 executions of the inner loop for hyperparameter optimization and 100 executions of the outer loop. The final predictive performance was reported as the mean Root Mean Square Error (RMSE) obtained from the 100 outer loop iterations.

2.4.3. Robustness Analysis

Robustness analysis is crucial in machine learning (ML) applications within the industrial sector, as it ensures models are reliable, accurate, and compliant, thereby mitigating risks associated with incorrect predictions. Sensitivity analysis serves as a key tool in assessing the robustness of ML models by examining how variations in input data influence model outputs. By systematically altering input variables and observing the resulting changes in predictions, one can identify scenarios where the model may underperform and evaluate its reliability under diverse conditions.

Several methods are commonly employed for conducting sensitivity analysis in machine learning:

1. **One-at-a-time (OAT) method**: This approach involves altering one input variable at a time while keeping all other inputs constant to observe the effect on the model's output.
2. **Morris method** [27]: This technique systematically and randomly varies each input variable to generate a set of samples. Statistical methods are then applied to determine the importance of each input variable.

3. **Sobol method** [28]: A more sophisticated approach that utilizes sequences of orthogonal arrays to systematically vary input variables, enabling the identification of significant input variables and their interactions.

In our approach, we leverage predictions obtained from the testing set using the best-performing algorithms for targets S (strength) and D (deformation). For a given pressure (P) and thickness (t), along with specific values of A and B, we present figures displaying the mean predicted values and their 95% confidence intervals for each pair of current (I) and time (t) values. These intervals are compared with actual observed values; if the actual value falls within the prediction interval, it indicates the model's robustness for those input parameters. See Figures 5–9.

2.4.4. Creation of Heat Maps

Finally, a strategy was developed to utilize the ML models in a way that facilitates multicriteria decision-making, incorporating welding quality, production time, and energy savings. As previously explained, weld quality is defined by its strength S, provided it does not exceed an acceptable deformation value D. Additionally, another important objective is to reduce the time required for each weld (t) to maximize production. Lastly, minimizing energy costs, which are proportional to the product $C = It$, serves as another key criterion. Depending on the specific production requirements, these priorities may vary; for example, sometimes the primary goal is to ensure high-quality welds, while in other cases, reducing energy usage to cut costs may take precedence. The importance of this last factor has increased substantially due to recent surges in electricity prices.

At first glance, optimizing the resistance spot welding process to balance energy consumption, efficiency, and quality appears to be a multiobjective optimization problem. However, in practice, different circumstances may require prioritizing one objective over the others. This is why we considered heat maps as a valuable tool to aid in selecting the most suitable settings at any given time. Although ML models can be employed for fully automated production, production engineers often need to ensure that the process operates within a stable region that meets predefined standards for time, quality, and energy efficiency. Thus, the proposed decision support system (DSS) utilizes heat maps generated from ML model predictions to identify, based on joint thickness, the optimal operating points according to weld quality criteria, limited by a maximum deformation D_{max} and a minimum strength S_{min}.

As discussed in Section 2, relying exclusively on heat maps may sometimes be restrictive, as additional information or multiple visualization techniques are often required. To address this, we propose an enhanced method that overlays additional data onto the heat maps, aiding decision-making without overwhelming the viewer. Specifically, our approach involves generating heat maps for a given thickness, with intensity (I) and time (t) represented on the axes. The colors on the heat map depict the strength, predicted by the AI models. To further enhance decision robustness, the color values correspond to the lower bound of the 95% confidence interval ($CI95_{min}$) for S, and the upper bound ($CI95_{max}$) for D is shown as region borders within the heat map. This design allows decision-makers to quickly interpret the data and make informed choices. By combining heat maps with AI model predictions and confidence intervals, this approach provides a comprehensive solution for visualizing data and selecting optimal values for time and intensity, taking into account required strength, confidence, risk of disturbances, and energy costs. An example heat map is shown in Section 3.

The heat map construction follows these steps:

1. Using the previously selected algorithm, the entire dataset is utilized to search for optimal hyperparameters via Bayesian optimization with 10-fold cross-validation.

2. Once the hyperparameters that minimize the cross-validation error are obtained, 3000 iterations are conducted to create a series of training datasets through random sampling with replacement. In each iteration, a test dataset is also generated with multiple random pairs of I and t, constrained within plausible physical ranges for a specific thickness and pressure. At the end of the 3000 iterations, a sufficiently large prediction dataset is generated, containing multiple predictions for each combination of input features by aggregating all test predictions.
3. With this dataset, and once thickness and p are set, heat maps can be created with matrices of $CI95_{min}$ for S and $CI95_{max}$ for D across the full range of possible values for I and t.

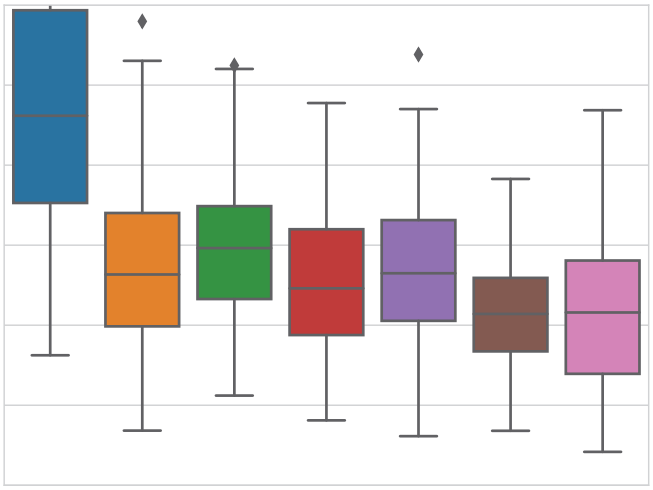


Figure 5. Boxplots of the RMSE obtained with each algorithm for predicting weld strength.

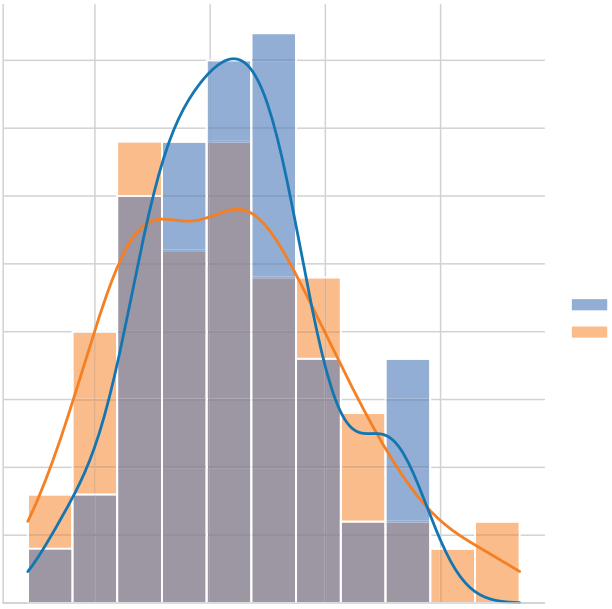


Figure 6. Comparison of the RMSE distribution obtained with the MLP and Autogluon algorithms for weld strength.

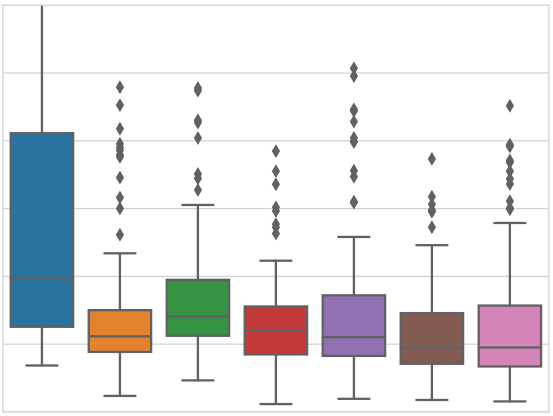


Figure 7. Boxplots of the RMSE obtained with each algorithm for predicting weld deformation.

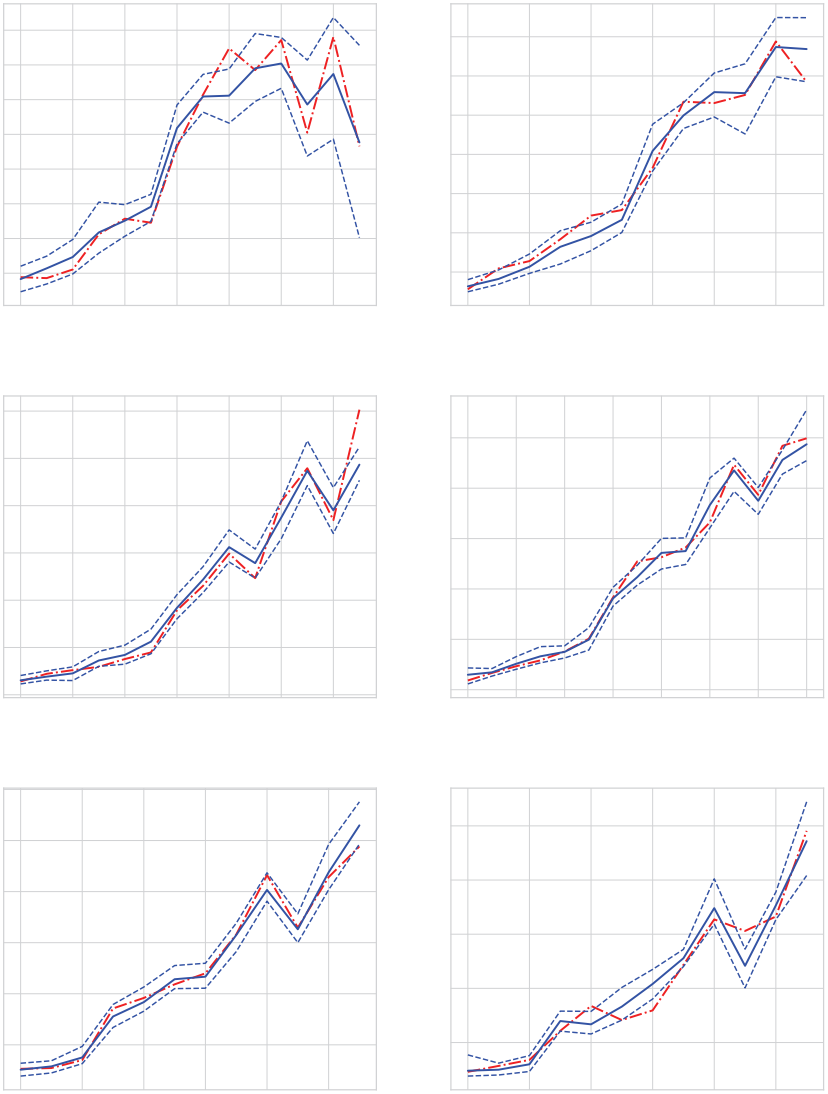


Figure 8. Prediction of S with $p = 6$ for each pair (I, t) and six thickness values. The dashed lines indicate a 95% confidence interval and the continuous line is the average value. The red dashed line corresponds to the real value for each pair of values (I, t). The values of I and t correspond to the % of the maximum intensity and % of the maximum time in 2 s.

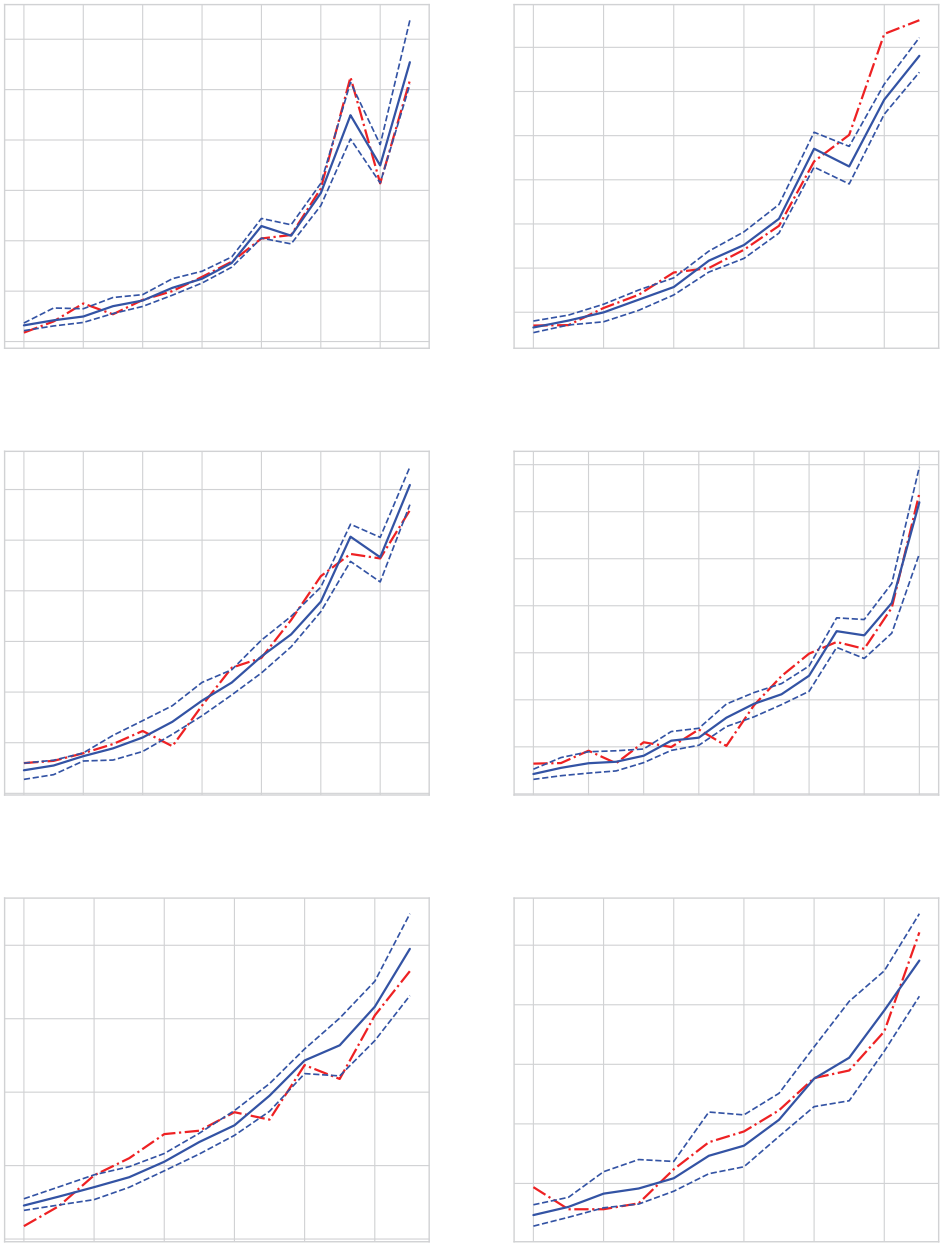


Figure 9. Prediction of D with $p = 6$ for each pair (I, t) and six thickness values. The dashed lines indicate a 95% confidence interval and the continuous line is the average value. The red dashed line corresponds to the real value for each pair of values (I, t). The values of I and t correspond to the % of the maximum intensity and % of the maximum time in 2 s.

3. Results and Discussion

This section presents the evaluation of the proposed methodology, focusing on its ability to optimize resistance spot welding (RSW) processes for steel reinforcement bars. The main objective is to balance three critical factors—weld quality, production efficiency, and energy consumption—by leveraging machine learning models and heat maps as decision support tools. The approach includes selecting robust predictive models, validating their generalization across different welding conditions, and utilizing interactive visualizations to aid decision-making. The results are discussed in terms of the accuracy, robustness, and practical applicability of the models, as well as the insights provided by the heat maps for multiobjective optimization.

It is worth emphasizing that the dataset used in this study comprised the typical thicknesses commonly employed in reinforcement fabrication. During the dataset creation process, certain thickness combinations failed to meet the minimum quality criteria due to weak or failed joints. Consequently, fewer viable records were available for extreme thickness combinations, which were limited to specific welding conditions. To overcome this limitation, a robust model selection strategy based on nested cross-validation was implemented, significantly improving the generalization capabilities of the machine learning models. The effectiveness of this approach across the entire range of thicknesses is demonstrated in Figures 8 and 9.

3.1. Search of the Best Algorithm for S and D

The first step, as outlined in the methodology, involved selecting the best ML algorithms and conducting the training and validation of the models (Steps 1 and 2 of the methodology).

Tables 4 and 5 present the error metrics for the best models obtained with each algorithm, ordered from lowest to highest Root Mean Square Error (RMSE). These tables also include the Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) for each model. The last column details the hyperparameters optimized through Bayesian optimization. Reported values correspond to the mean and standard deviation (in parentheses) of 100 measurements, collected from each iteration of the outer loop in a 10-repeated 10-fold nested cross-validation. Additionally, Figures 5 and 7 show comparative box plots for both target variables.

The selection of the best algorithm was based on a combined analysis of the mean and variance of RMSE. For *S*, the multilayer perceptron (MLP) was chosen, as it demonstrated a favorable balance between accuracy and robustness. Although AutoGluon achieved a slightly lower mean RMSE than MLP (2085.5 vs. 2104.3), the MLP model exhibited a significantly lower standard deviation (363.7 vs. 462.6), and none of the MLP predictions exceeded an RMSE of 3000, as shown in Figure 6. This indicates that MLP models not only achieved high accuracy but were also more robust, with considerably lower variability. Similar patterns were observed for MAE and MAPE, reinforcing the robustness of the MLP model.

Table 4. Errors and best hyperparameters for *S*. Values are the average and the standard deviation (between parentheses) obtained with each algorithm and a 10-repeated 10-fold nested cross-validation (100 measures).

Algorithm	RMSE	MAE	MAPE	Hyperparameters
Autogluon	**2085.5** (462.6)	**1486.9** (291.2)	**16.2** (3.2)	*presets* = *best_quality* *time_limit* = 5 min
MLP	2104.3 (**363.7**)	1546.0 (**263.6**)	17.5 (**2.7**)	*hidden_size* = 7.20 (3.99) $alpha = 10^{-1.673}$ ($10^{0.513}$) *activation* = *logistic*
RF	2275.1 (486.1)	1623.6 (320.1)	17.3 (3.4)	*max_depth* = 17.97 (6.48) *n_estimators* = 30.82 (11.74) $min_imp = 2.8 \times 10^{-5}$ (2×10^{-4})
KRidge	2350.2 (497.0)	1671.5 (311.2)	19.1 (3.5)	$alpha = 10^{-1.296}$ ($10^{0.198}$) $gamma = 10^{-0.623}$ ($10^{0.138}$) *kernel* = *rbf*
SVR	2351.4 (472.6)	1644.9 (299.0)	18.8 (3.4)	$C = 10^{0.769}$ ($10^{0.362}$) $gamma = 10^{-0.665}$ ($10^{0.175}$) *kernel* = *rbf*
DTR	2502.6 (510.6)	1840.3 (356.3)	19.7 (3.9)	*max_depth* = 14.72 (6.90) $min_imp = 6.8 \times 10^{-4}$ (7.1×10^{-4})
Ridge	3381.8 (770.4)	2366.8 (440.1)	25.6 (4.9)	$alpha = 10^{-6.062}$ ($10^{2.246}$) $tol = 1.0 \times 10^{-4}$

Table 5. Errors and best hyperparameters for D. Values are the average and the standard deviation (between parentheses) obtained with each algorithm and a 10-repeated 10-fold nested cross-validation (100 measures).

Algorithm	RMSE	MAE	MAPE	Hyperparameters
MLP	**0.0109** (0.0036)	**0.0077** (**0.0016**)	**30.91** (10.09)	*hidden_size* = 4.42 (2.50) $alpha = 10^{-2.269}$ ($10^{1.266}$) *activation* = *logistic*
Autogluon	0.0114 (0.0048)	**0.0077** (0.0018)	32.09 (11.81)	*presets* = *best_quality* *time_limit* = 5 min
RF	0.0117 (**0.0035**)	0.0083 (**0.0016**)	33.15 (12.60)	*max_depth* = 18.78 (6.35) *n_estimators* = 32.43 (10.94) $min_imp = 5.9 \times 10^{-6}$ (4.2×10^{-5})
KRidge	0.0121 (0.0049)	0.0082 (0.0018)	32.25 (11.49)	$alpha = 10^{-3.856}$ ($10^{1.172}$) $gamma = 10^{-2.037}$ ($10^{0.456}$) *kernel* = *rbf*
SVR	0.0125 (0.0055)	0.0083 (0.0020)	32.35 (11.28)	C=$10^{2.729}$ ($10^{0.378}$) $gamma = 10^{-1.789}$ ($10^{0.207}$) *kernel* = *rbf*
DTR	0.0133 (0.0044)	0.0097 (0.0018)	40.93 (15.02)	*max_depth* = 17.64 (7.18) $min_imp = 5.5 \times 10^{-4}$ (8.0×10^{-4})
Ridge	0.0198 (0.0099)	0.0114 (0.0033)	34.81 (**8.37**)	$alpha = 10^{0.148}$ ($10^{0.784}$) $tol = 1.0 \times 10^{-4}$

For D, the MLP obtained the lowest mean RMSE (0.0109) and a standard deviation value of 0.0036, very close to that achieved with RF. MLP also obtained the best MAE and MAPE.

3.2. Model Robustness Analysis

After selecting the best models and evaluating their performance, the next step was to analyze their behavior and variability in response to changes in input variables.

Figures 8 and 9 present examples of the combined models' predictions for S (strength) and D (deformation), respectively, across different combinations of I (intensity) and t (time) with $p = 6$ and for six of the seven thickness levels in the dataset. Note that I and t values are expressed as percentages of the welding machine's maximum intensity and a maximum welding time of 2 s, respectively. The cases are ordered first by increasing electrical intensity (I), and within each intensity level, by increasing t. Consequently, cases located further to the right correspond to higher energy levels, as energy is proportional to the product of I and t. The solid blue line represents the mean prediction for each group of cases defined by I and t, while the dashed lines show the 95% confidence interval ($CI95$) of the estimate. The red line indicates the average actual value for each case.

These graphs demonstrate that, for each thickness, an appropriate selection of I and t enables a balance between achieving adequate weld strength and maintaining quality. For S, the lower bound of the $CI95$ ($CI95_{min}$) is the critical value to examine, as it represents the most conservative prediction for strength. Conversely, for D, the upper bound of the $CI95$ ($CI95_{max}$) is of interest, as it reflects the worst-case scenario for deformation.

The results indicate that the models exhibit robust behavior, as evidenced by narrow confidence intervals, with the red line (actual values) generally falling within these intervals. However, for the 18 mm thickness, particularly at high energy levels (i.e., high I and t values), the confidence intervals widen, indicating increased uncertainty in predictions for these conditions. Thus, caution is advised when operating with 18 mm thicknesses at elevated energy levels.

Similar patterns were observed for pressure settings of $p = 5$ and $p = 7$.

3.3. Heat Maps for Decision-Making

Figures 10 and 11 present two examples of heat maps incorporating I (intensity) and t (time) to estimate the values of S (strength) and D (deformation).

To enhance the understanding of these figures, Table 6 provides a detailed description and interpretation of the variables depicted.

To support multicriteria decision-making, the $CI95_{min}$ of S and $CI95_{max}$ of D are superimposed on these figures. The colored regions represent the $CI95_{min}$ values of S, while solid lines indicate the $CI95_{max}$ boundaries for D. Specifically, the green line corresponds to $CI95_{max} = 0.05$ for D, the yellow line to $CI95_{max} = 0.10$, and the red line to $CI95_{max} = 0.15$.

Figure 10 shows the heat map for thickness $= 18$ and $p = 6$. The area between the green and yellow lines represents a $CI95_{max}$ range of D between 0.05 and 0.10. If a maximum D of 0.10 is required, position 1 ($t = 33\%$ and $I = 64\%$) would be appropriate, as it ensures a $CI95_{min}$ of S within 10,000–12,000N. However, this decision could be reconsidered if the objective is to further reduce time and energy consumption, albeit with a potential reduction in weld quality. For instance, position 2 represents an alternative operating point with shorter times, at the expense of reducing the S range to 8000–10,000N.

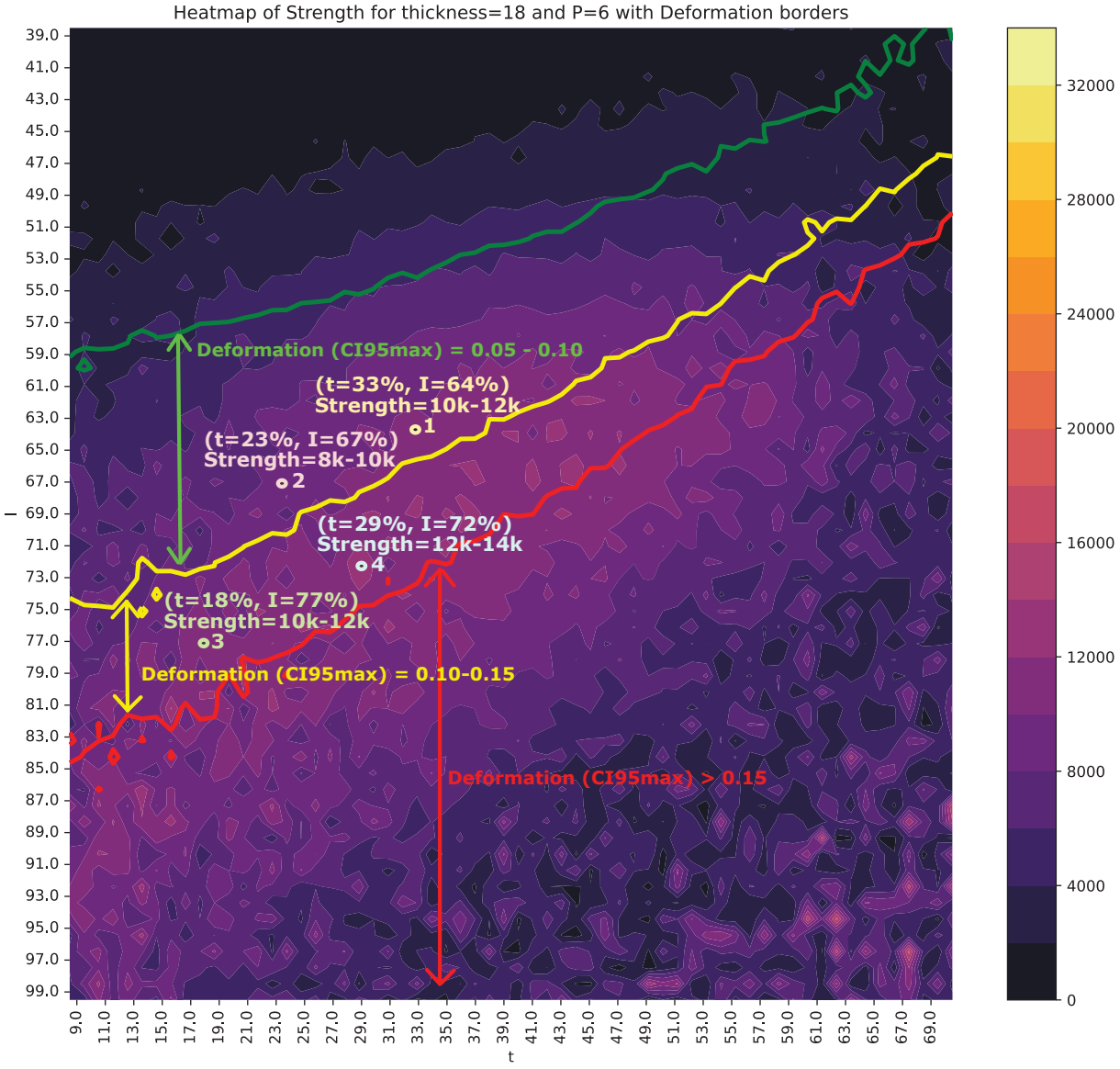


Figure 10. Heat map showing the $CI95_{min}$ value of S and $CI95_{max}$ of D for each combination of I and t, with *thickness* $= 18$ and $p = 6$. The multicolored vertical strip represents the $CI95_{min}$ value of S, while the green, yellow, and red contour lines indicate $CI95_{max}$ values of D at levels of 0.05, 0.10, and 0.15, respectively. The values of I and t correspond to the percentage of the maximum intensity and the percentage of the maximum time (2 s), respectively.

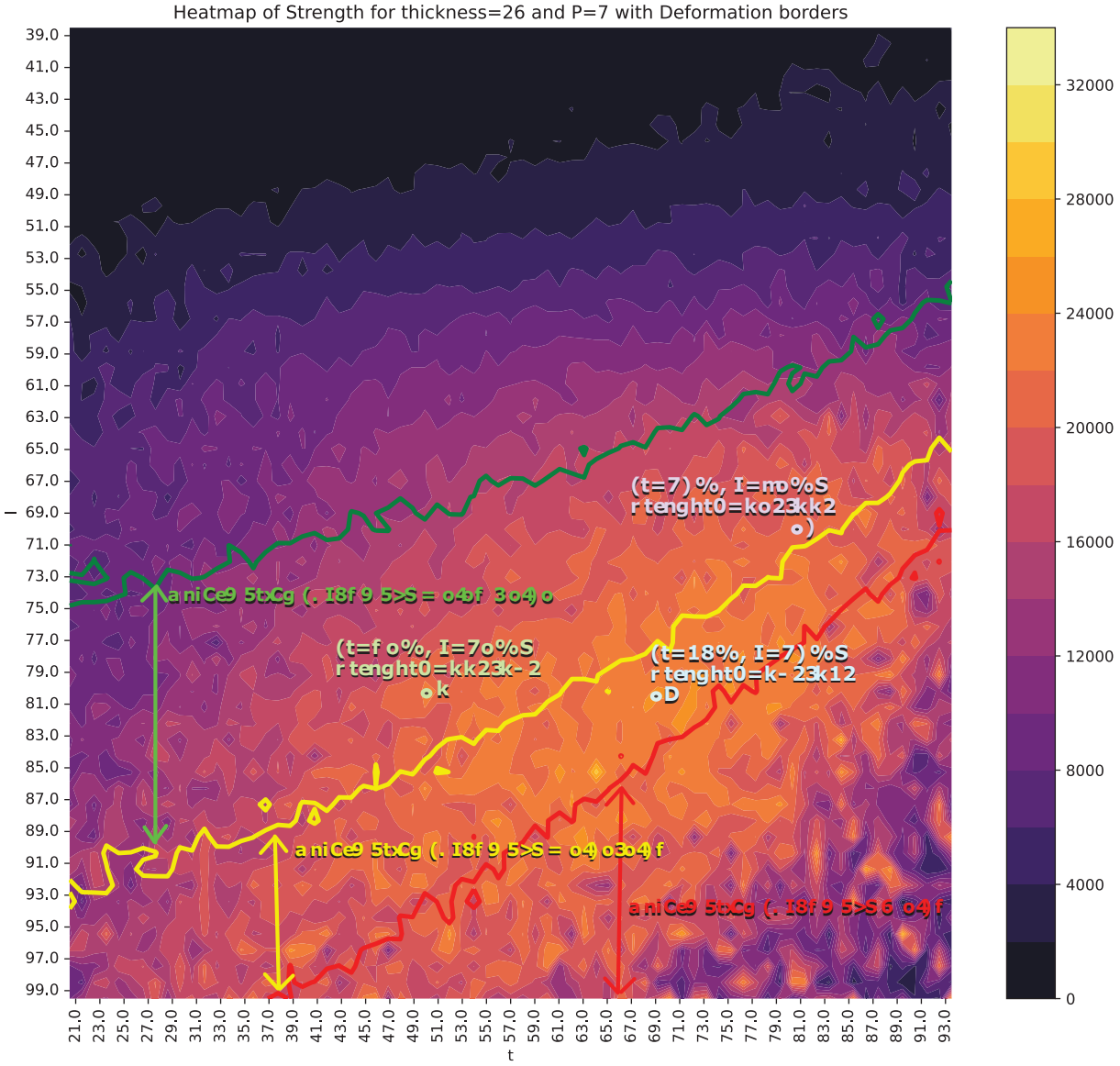


Figure 11. Heat map displaying the $CI95_{min}$ value of S and $CI95_{max}$ of D for each combination of I and t, with $thickness = 26$ and $p = 7$. The multicolored vertical strip represents the $CI95_{min}$ value of S, while the green, yellow, and red contour lines denote $CI95_{max}$ values of D at levels of 0.05, 0.10, and 0.15, respectively. The values of I and t are expressed as percentages of the maximum intensity and maximum time (2 s) of the welding machine.

Point 3 maintains the same S range as Point 1 but halves the value of t, with a trade-off: higher energy consumption and reduced weld quality, as it falls within the $CI95_{max}$ range of D between 0.10 and 0.15. Lastly, Point 4 increases the S range to 12,000–14,000N, although it lies within a smaller, isolated region of stability.

Another example is shown in Figure 11 for thickness $= 26$ and $p = 7$. In this case, a viable option is position 1, with $t = 81\%$ and $I = 70\%$, resulting in an S range between 20,000 and 22,000 N. Alternatively, other operating points may be chosen to increase S: for instance, position 2 achieves a range of 22,000–24,000 N with a reduction in t to 50% but an increase in I by 10%. Lastly, position 3 extends the time parameter, raising the S range to 24,000–26,000 N, though with deformation values between 0.10 and 0.15.

Table 6. Description and interpretation of variables included in the heat maps.

Variable	Description	Interpretation in Heat Maps
t (Time)	Represents the welding time, expressed as a percentage of the maximum welding time of 2 s.	Displayed along one axis of the heat maps, t influences the energy applied during the welding process.
I (Intensity)	The electrical current intensity used during welding, expressed as a percentage of the welding machine's maximum intensity.	Shown along one axis of the heat maps, I contributes to the energy applied during the welding process.
S (Strength)	Represents the weld strength, calculated as a combination of shear and tensile strength.	The heat maps show the lower bound of the 95% confidence interval ($CI95_{min}$) for S, representing the most conservative prediction for weld strength.
D (Deformation)	Describes the deformation of the weld, expressed as a percentage of the total thickness of the joint.	The heat maps display the upper bound of the 95% confidence interval ($CI95_{max}$) for D, reflecting the worst-case scenario for deformation.
$CI95_{min}$	The lower bound of the 95% confidence interval for S.	Indicates the most conservative prediction for weld strength, used to assess the reliability of strength predictions. The multicolored vertical strip represents this variable.
$CI95_{max}$	The upper bound of the 95% confidence interval for D.	Indicates the worst-case scenario for weld deformation, guiding decisions to minimize deformation. The green, yellow, and red contour lines denote values of this variable at levels of 0.05, 0.10, and 0.15, respectively.

3.4. Trade-Offs Between Quality, Efficiency, and Energy for thickness $= 18$ *and* $p = 6$

Table 7 presents the key parameters for quality, production time, and energy consumption in the welding process with *thickness* $= 18$ and $p = 6$ (Figure 10). This table highlights critical insights into process performance at different operational points, emphasizing the trade-offs between production efficiency, energy savings, and weld quality. Values highlighted in red indicate the most unfavorable conditions, while those in bold black represent the optimal values.

Table 7. Quality, production times, and consumed energy corresponding to the points in Figure 10 of a process with *thickness* $= 18$ and $p = 6$.. The bold values correspond to the best results obtained for each variable

	Quality		Time			Energy			
Point	***S* (kN)**	***D* (%)**	***t* (%)**	***t* (s)**	**Inc. Prod (%)**	***I* (%)**	***P* (kVA)**	***E* (kJ)**	**Reduc. *E* (%)**
1	10–12	**5–10**	33	0.66	0.0	64	32.00	675.84	10.1
2	8–10	**5–10**	23	0.46	8.2	67	33.50	516.235	**31.3**
3	10–12	10–15	18	0.36	**12.8**	77	38.50	533.61	29.0
4	**12–14**	10–15	29	0.58	3.1	72	36.00	751.68	0.0

In the time-related parameters, the column *Inc. Prod* (%) quantifies the production increase relative to the most unfavorable case, which corresponds to the longest welding time (Point 1). For these calculations, it is assumed that 25% of the total production cycle corresponds to welding operations. The percentage increment in production is calculated using the following formula:

$$\text{Inc. Prod}(\%) = \frac{t_{\max} - t}{3t_{\max} + t} \times 100\%, \tag{3}$$

where $t_{\max}$ represents the longest welding time and t is the welding time for the specific operational point. This metric provides a quantitative comparison of production efficiency gains achieved under different process conditions.

In the energy-related parameters, the column P (kVA) is derived based on the welding machine's maximum power output of 50 kVA, while the column *Reduc. E* (%) represents the percentage reduction in energy consumption relative to the most unfavorable case

(Point 4), where energy consumption is at its maximum. This reduction is determined using the following formula:

$$\text{Dec. E}(\%) = \frac{E_{\max} - E}{E_{\max}} \times 100\%, \tag{4}$$

where $E_{\max}$ is the energy consumption at Point 4, and E corresponds to the energy consumption at the respective point. This metric highlights the potential energy savings achievable through optimized process parameters.

Overall, Table 7 provides a comprehensive view of the trade-offs between weld quality, production efficiency, and energy consumption. Depending on production objectives and energy costs at a given time, different points can be selected. For example, Point 4 corresponds to a weld with excellent quality (S maximum), although it incurs a high energy cost and achieves only a 3.1% increase in production. In contrast, if quality requirements are not stringent but energy cost reduction is critical, Point 2 can be selected, offering an energy reduction of 31.3%. However, this comes at the expense of having the lowest weld strength (S) in the dataset. Point 3, on the other hand, increases production by 12.8% while achieving an energy cost reduction similar to Point 2. If the quality objectives related to S and D are sufficient, it could represent a balanced choice.

Based on the results obtained, it can be concluded that the selected machine learning models exhibit remarkable robustness under varying input conditions, maintaining stability even when operational parameters fluctuate. This reliability is further enhanced by the confidence intervals visualized in the heat maps, which provide critical guidance for production decisions. These visualizations ensure that decisions are not only optimal but also resilient to disturbances, thereby guaranteeing consistent weld quality and sustained operational efficiency.

Overall, the integration of machine learning models with interactive heat maps effectively addresses the key challenges in optimizing resistance spot welding processes for steel reinforcement bars. This approach enables the identification of stable operational zones tailored to different production priorities, such as energy efficiency, weld quality, or production speed, while also enhancing adaptability to dynamic scenarios. In conclusion, the combination of robust predictive tools with dynamic visualizations highlights the practical value of this methodology, offering a flexible and reliable framework for decision-making in industrial contexts.

4. Conclusions

This study introduces an innovative machine learning-based methodology to optimize resistance spot welding (RSW) processes for steel reinforcement bars, addressing critical challenges in balancing weld quality, energy efficiency, and productivity. By leveraging advanced predictive models and interactive visualization tools, the proposed system empowers production engineers to make adaptive and data-driven decisions, even under dynamic industrial conditions.

The novelty of this approach lies in its integration of predictive modeling with interactive heat maps as user-friendly visualization techniques, significantly enhancing its practical value. By enabling users to pinpoint optimal parameters that align with priorities such as quality, efficiency, or processing time, this methodology facilitates adaptive decision-making in diverse industrial contexts.

Future research could explore real-time sensor integration and adaptive control systems to further enhance its applicability in complex environments. Additionally, developing advanced visualization techniques tailored to high-dimensional data could open up new possibilities for intuitive decision-making.

In conclusion, this study highlights the transformative potential of AI-driven decision support systems in industrial automation, presenting a robust and scalable framework for developing reliable machine learning models, even in data-constrained scenarios. By incorporating intuitive graphical tools for parameter optimization, this approach not only addresses key challenges in sustainability and operational efficiency but also paves the way for smarter and more sustainable manufacturing practices.

Author Contributions: Conceptualization, F.J.M.-d.-P. and E.F.-G.; Methodology, F.J.M.-d.-P.; Software, A.P.-E. and J.D.; Validation, A.P.-E.; Formal analysis, J.D.; Investigation, E.F.-G.; Resources, J.F.-C.; Data curation, J.F.-C., E.F.-G. and A.P.-E.; Writing—original draft, J.D.; Writing—review & editing, F.J.M.-d.-P.; Visualization, F.J.M.-d.-P. and J.D.; Supervision, F.J.M.-d.-P.; Funding acquisition, E.F.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received external funding from projects REGI2020/67, REGI2022/41 and REGI2020/41. It was also supported by grant PID2021-123219OB-I00 and PID2020-116641GB-I00 funded by the Ministry of Science, Innovation and Universities of Spain and also by project INICIA 2023/02 funded by La Rioja Government (Spain).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: The datasets presented in this article are not readily available because the data are part of an ongoing study

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Zhan, X.; Ou, W.; Wei, Y.; Jiang, J. The feasibility of intelligent welding procedure qualification system for Q345R SMAW. *Int. J. Adv. Manuf. Technol.* **2015**, *83*, 765–777. [CrossRef]
2. Gyasi, E.A.; Handroos, H.; Kah, P. Survey on artificial intelligence (AI) applied in welding: A future scenario of the influence of AI on technological, economic, educational and social changes. *Procedia Manuf.* **2019**, *38*, 702–714. [CrossRef]
3. Wang, X.; Zhang, Y.; Liu, J.; Luo, Z.; Zielinska, T.; Ge, W. Online detection of weld surface defects based on improved incremental learning approach. *Expert Syst. Appl.* **2022**, *195*, 116407. [CrossRef]
4. Satyro, W.C.; de Almeida, C.M.V.B.; Pinto, M.J.A., Jr.; Contador, J.C.; Giannetti, B.F.; de Lima, A.F.; Fragomeni, M.A. Industry 4.0 implementation: The relevance of sustainability and the potential social impact in a developing country. *J. Clean. Prod.* **2022**, *337*, 130456. [CrossRef]
5. Matsushita, M.; Taniguchi, K.; Oi, K. Development of next generation resistance spot welding technologies contributing to auto body weight reduction. *JFE Tech. Rep.* **2013**, *18*, 111–117.
6. Goffin, N.; Jones, L.C.; Tyrer, J.; Ouyang, J.; Mativenga, P.; Woolley, E. Mathematical modelling for energy efficiency improvement in laser welding. *J. Clean. Prod.* **2021**, *322*, 129012. [CrossRef]
7. Ferreiro-Cabello, J.; Fraile-Garcia, E.; Lara-Santillán, P.M.; Mendoza-Villena, M. Assessment and Optimization of a Clean and Healthier Fusion Welding Procedure for Rebar in Building Structures. *Appl. Sci.* **2020**, *10*, 7045. [CrossRef]
8. Panda, B.N.; Babhubalendruni, M.V.A.R.; Biswal, B.B.; Rajput, D.S. Application of Artificial Intelligence Methods to Spot Welding of Commercial Aluminum Sheets (B.S. 1050). In *Advances in Intelligent Systems and Computing*; Springer: New Delhi, India, 2014; pp. 21–32. [CrossRef]
9. Zaharuddin, M.F.A.; Kim, D.; Rhee, S. An ANFIS based approach for predicting the weld strength of resistance spot welding in artificial intelligence development. *J. Mech. Sci. Technol.* **2017**, *31*, 5467–5476. [CrossRef]
10. Cho, Y.; Rhee, S. New technology for measuring dynamic resistance and estimating strength in resistance spot welding. *Meas. Sci. Technol.* **2000**, *11*, 1173–1178. [CrossRef]
11. Wu, J.; Zhang, C.; Lian, K.; Cao, H.; Li, C. Carbon emission modeling and mechanical properties of laser, arc and laser–arc hybrid welded aluminum alloy joints. *J. Clean. Prod.* **2022**, *378*, 134437. [CrossRef]
12. Gujre, V.S.; Anand, R. Machine learning algorithms for failure prediction and yield improvement during electric resistance welded tube manufacturing. *J. Exp. Theor. Artif. Intell.* **2019**, *32*, 601–622. [CrossRef]

13. Jaypuria, S.; Bondada, V.; Gupta, S.K.; Pratihar, D.K.; Chakrabarti, D.; Jha, M. Prediction of electron beam weld quality from weld bead surface using clustering and support vector regression. *Expert Syst. Appl.* **2023**, *211*, 118677. [CrossRef]
14. Zhou, B.; Pychynski, T.; Reischl, M.; Kharlamov, E.; Mikut, R. Machine learning with domain knowledge for predictive quality monitoring in resistance spot welding. *J. Intell. Manuf.* **2022**, *33*, 1139–1163. [CrossRef]
15. Panza, L.; Bruno, G.; Antal, G.; Maddis, M.D.; Spena, P.R. Machine learning tool for the prediction of electrode wear effect on the quality of resistance spot welds. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2024**, *18*, 4629–4646. [CrossRef]
16. Zhang, Y.; Li, X.; Wang, Y.; Liu, Y.; Zhang, Y. A CNN-LSTM and Attention-Mechanism-Based Resistance Spot Welding Quality Online Detection Method for Automotive Bodies. *Mathematics* **2023**, *11*, 4570. [CrossRef]
17. Chertov, A.; Maev, R.; Severin, F.; Gumenyuk, A.; Rethmeier, M. Real-Time AI driven Interpretation of Ultrasonic Data from Resistance Spot weld in Vehicle Structures Using Dynamic Resistance Adaptive Control. *Mater. Eval.* **2023**, *81*, 61–70. [CrossRef]
18. Kraljevski, I.; Ju, Y.C.; Ivanov, D.; Tschöpe, C.; Wolff, M. How to Do Machine Learning with Small Data?—A Review from an Industrial Perspective. *arXiv* **2023**, arXiv:2311.07126.
19. Pastor-López, I.; Sanz, B.; Tellaeche, A.; Psaila, G.; de la Puerta, J.G.; Bringas, P.G. Quality assessment methodology based on machine learning with small datasets: Industrial castings defects. *Neurocomputing* **2021**, *456*, 622–628. [CrossRef]
20. Knauer, R.; Rodner, E. Squeezing Lemons with Hammers: An Evaluation of AutoML and Tabular Deep Learning for Data-Scarce Classification Applications. *arXiv* **2024**, arXiv:2405.07662.
21. Merghadi, A.; Yunus, A.P.; Dou, J.; Whiteley, J.; ThaiPham, B. Machine learning methods for landslide susceptibility studies: A comparative overview of algorithm performance. *Earth-Sci. Rev.* **2020**, *207*, 103225. [CrossRef]
22. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 2951–2959.
23. Vikhar, P.A. Evolutionary algorithms: A critical review and its future prospects. In Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, India, 22–24 December 2016; pp. 261–265. [CrossRef]
24. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 10031–10061. [CrossRef]
25. Zhang, X.; Chan, F.T.; Yan, C.; Bose, I. Towards risk-aware artificial intelligence and machine learning systems: An overview. *Decis. Support Syst.* **2022**, *159*, 113800. [CrossRef]
26. Vabalas, A.; Gowen, E.; Poliakoff, E.; Casson, A.J. Machine learning algorithm validation with a limited sample size. *PLoS ONE* **2019**, *14*, e0224365. [CrossRef]
27. Morris, M.D. Factorial Sampling Plans for Preliminary Computational Experiments. *Technometrics* **1991**, *33*, 161–174. [CrossRef]
28. Sobol', I.M. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simul. (MATCOM)* **2001**, *55*, 271–280. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Beyond Information Distortion: Imaging Variable-Length Time Series Data for Classification

Hyeonsu Lee and Dongmin Shin *

Department of Industrial and Management Engineering, Hanyang University, Ansan 15588, Republic of Korea; hyeonsulee@hanyang.ac.kr

* Correspondence: dmshin@hanyang.ac.kr

Abstract: Time series data are prevalent in diverse fields such as manufacturing and sensor-based human activity recognition. In real-world applications, these data are often collected with variable sample lengths, which can pose challenges for classification models that typically require fixed-length inputs. Existing approaches either employ models designed to handle variable input sizes or standardize sample lengths before applying models; however, we contend that these approaches may compromise data integrity and ultimately reduce model performance. To address this issue, we propose Time series Into Pixels (TIP), an intuitive yet strong method that maps each time series data point into a pixel in 2D representation, where the vertical axis represents time steps and the horizontal axis captures the value at each timestamp. To evaluate our representation without relying on a powerful vision model as a backbone, we employ a straightforward LeNet-like 2D CNN model. Through extensive evaluations against 10 baseline models across 11 real-world benchmarks, TIP achieves 2–5% higher accuracy and 10–25% higher macro average precision. We also demonstrate that TIP performs comparably on complex multivariate data, with ablation studies underscoring the potential hazard of length normalization techniques in variable-length scenarios. We believe this method provides a significant advancement for handling variable-length time series data in real-world applications. The code is publicly available.

Keywords: time series classification (TSC); variable-length time series

1. Introduction

Time series data are ubiquitous and appear across various domains, from trajectories [1] and sensor-based human action recognition [2] to monitoring data from facilities and machines in manufacturing systems [3]. Time series data collected from the real-world often have heterogeneous traits, such as *variable length*—the lengths of the series sample are diverse [4]. Given the prevalence of variable-length time series data in real-world scenarios [5], finding effective ways to handle such data is crucial. Previous studies have preprocessed variable-length time series data to make their lengths uniform by sliding a fixed-size window [6], resampling, or introducing artificial values [7]. However, the best method for managing variable-length time series data and their effectiveness are still unclear, with many recent studies having highlighted it as a future research question in TSC [4,8–10].

This issue is even more pronounced for time series data collected by sensors, one of the most widely used tools to gather real-world time series data [11]. Such data often contain consistent noise [12], thus making it critical to capture *key local patterns*—the patterns or

subsequences that best explain inter-class variability [13] (e.g., time series data from walking versus sitting classes, or normal versus anomalous data in machine monitoring, may each have distinct key local patterns). To capture key local patterns for TSC, 1D-CNN methods are traditionally employed due to their success in detecting local patterns [14]. One-dimensional CNNs apply filters along the time dimension, enabling the network to identify local patterns that distinguish between classes. Additionally, CNNs can detect features regardless of their position within the time series, which is particularly advantageous in TSC, where the location of key local patterns can vary across instances [15–17].

However, capturing key local patterns in variable-length time series presents a significant challenge for 1D CNNs. These models require a fixed input size for batch processing, necessitating length uniformization techniques such as truncation, interpolation, or padding. While these methods enable the use of a 1D CNN architecture, they can introduce noise and distort critical information, including key local patterns, in the original data. This may negatively affect model performance by interfering with the capture of key local patterns, eventually leading to suboptimal results. Figure 1 illustrates this phenomenon.

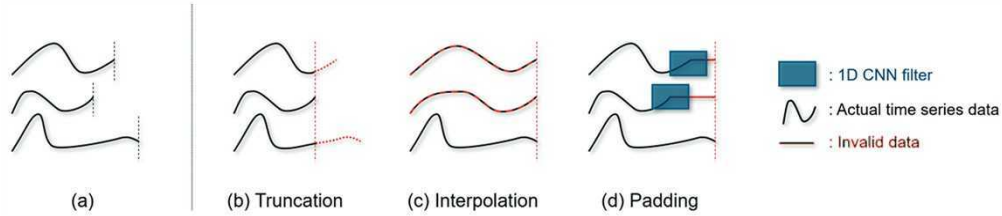


Figure 1. (**a**) Toy example illustrations of variable-length time series data. (**b**–**d**) Common techniques to uniformize the variable-length time series data. (**b**) Truncation cuts off parts of the series to fit a predetermined length, potentially removing crucial information. (**c**) Interpolation inserts data points to adjust the series to a standard length, distorting true temporal dynamics. (**d**) Padding adds repeating values at the end, creating misleading patterns and forcing 1D convolutional filters to process both actual and invalid data indiscriminately.

Although alternatives to 1D CNNs are viable for handling data heterogeneity and are frequently used in various domains [4], their ability to capture features for TSC is questionable. Specifically, for non-deep learning-based methods, the separation of feature extraction and downstream tasks can constrain their performance. While RNN-based and transformer-based approaches can preserve data integrity by using mask layers to avoid padding effects, their processing architecture may struggle to capture the often position-variant key local patterns in TSC. This creates an unresolved trade-off between capturing key local patterns and preserving the data integrity of variable-length time series.

Motivated by these limitations, we focus on one primary question: "***How can we effectively capture the key local pattern for TSC while preserving the original representation of variable-length time series?***"

To address this question, we introduce **TIP** (Time series Into Pixels), which directly maps each value at each time step into a pixel in a 2D image, where the vertical axis represents time steps and the horizontal axis represents the value at each time step. By converting 1D time series data into a 2D format, TIP enables the use of CNNs to capture key local patterns while preserving data integrity for variable-length TSC.

To evaluate our method, we conduct extensive experiments, comparing TIP to 10 baseline methods across 11 real-world benchmarks. Without relying on a powerful vision model as a backbone, we use a straightforward LeNet-like 2D CNN model. Through these experiments, TIP consistently outperforms existing methods, showing 2–5% higher accuracy and 10–25% higher macro average precision, demonstrating the effectiveness of our method for variable-length time series classification. Furthermore, we show that TIP

performs comparably in complex multivariate scenarios. Finally, through an ablation study, we confirm that length uniformization techniques for handling variable-length data can lead to performance degradation.

Our contributions are twofold:

1. We introduce TIP, a simple yet robust method that converts variable-length time series into a 2D pixel-based format, mapping time steps to the vertical axis and values to the horizontal. This representation enables vision models, including CNNs, to capture key local patterns effectively while preserving data integrity.
2. We validate TIP by employing a straightforward LeNet-like CNN, demonstrating that its performance gains arise from the data representation itself rather than the complexity or power of the vision model.

2. Existing Solutions for Variable-Length TSC

In this section, we review existing methods and potential solutions not yet explored in the literature for variable-length TSC. We categorize these solutions into three groups: non-deep learning-based solutions, masking-based solutions, and 1D-CNN-based solutions with preprocessing. We exclude single-batch-sized solutions from deep learning-based methods; although technically feasible, they are impractical due to high variance in batch loss, resulting in unstable and prolonged training. Additionally, this approach underutilizes GPU resources. We organized the main differences between previous methods and our method in Table 1.

Table 1. Comparison of existing solutions and our proposed method for variable-length TSC.

Aspect	Non-Deep Learning-Based (Section 2.1)	Masking-Based (Section 2.2)	1D CNN-Based (Section 2.3)	Ours
End-to-end learning	×	✓	✓	✓
Effective feature extraction for TSC	✓	×	✓	✓
Preserving original data integrity	✓	✓	×	✓

We include early classification methods within the scope of non-deep learning-based solutions. Although they could fit other categories depending on the classifier, recent studies employ SVMs, so we include them in this category.

2.1. Non-Deep Learning-Based Solutions

Before the emergence of deep learning in TSC, non-deep learning or machine learning models were primarily used. These models configured the feature space by measuring distances between samples or identifying key local patterns or shapelets essential for classification. With flexible feature extraction approaches, many of these models do not require fixed-size inputs, making them suitable for variable-length TSC problems. For example, DTW-1NN [18], a go-to method in TSC [19], handles variable-length time series data using dynamic time warping (DTW) distance, capturing time series characteristics like variations in speed and duration without modifying the series. Similarly, shapelet-based methods extract key local patterns by sampling subsequences or shapelets and selecting the ones that best characterize each class, supporting variable-length TSC without uniform length conversion. Early classification methods further address variable-length TSC by making confident predictions on partial data through adaptive thresholding or window-based segmentation, thus enabling efficient classification without full sequence input.

While these methods are effective, with some performing comparably to state-of-the-art models [8], they deviate from deep learning's end-to-end approach. The separation

between feature extraction and classification can limit their ability to optimize the feature space for identifying key local patterns.

2.2. Masking-Based Solutions

RNN-based models were initially the primary choice in deep learning for time series analysis due to their ability to capture temporal dependencies, making them widely used in TSC across various domains. More recently, transformers have gained popularity in time series analysis, particularly for handling variable-length sequences. Having proven their success in NLP, where sentence lengths vary, they manage sequence variability with padding, masking, and selective computation. RNNs handle padded sequences by excluding artificial values from loss calculations through element-wise summation, while transformers use attention mechanisms, assigning large negative values to padding elements, which reduces their softmax attention scores to zero.

While RNNs and transformers perform well in time series tasks like long-term forecasting and imputation, they have limited application in TSC. Apart from other issues detailed in [14], one key limitation is their tendency to capture global dependencies rather than local ones. This poses challenges in identifying key local patterns essential for classification, especially given common variations in these patterns (e.g., phase, warping, offset) in time series data [20,21]. Some approaches utilize RNNs to address variable-length TSC by embedding series into fixed-size latent representations and applying a classifier, similar to a seq2seq architecture [22]. However, this approach underperforms due to ineffective feature extraction, especially when compared with non-deep learning methods and 1D CNN-based deep learning methods, which will be discussed next.

2.3. 1D CNN-Based with Preprocessing Solutions

One-dimensional CNN-based methods have become a gold standard in TSC thanks to their robust feature extraction capabilities [19]. By employing 1D convolutional filters, these models effectively capture various local patterns—including phase shifts, warping, and offsets—regardless of where they appear along the temporal axis. Despite these strengths, 1D CNNs typically require fixed-length inputs, posing challenges for TSC tasks with variable-length sequences.

A common workaround is to uniformly extend each time series in a dataset to the maximum sequence length via padding. Unfortunately, this practice can degrade data integrity by mixing valid observations with synthetic zeros, thereby obscuring the true signal. In 1D CNNs, convolutional filters then process both real and padded values together, generating noisy feature maps that struggle to differentiate between authentic and invalid data.

This issue is compounded by z-normalization, a widely recognized preprocessing step in TSC that scales each time series to zero mean and unit variance [19,23,24]. Used extensively across the TSC domain—including many state-of-the-art methods such as InceptionTime [25]—z-normalization promotes stable training and consistent input scaling. However, once normalized, zero-padding can become statistically indistinguishable from genuinely low-value data points, making it even harder for CNN filters to separate signals from padding. This often leads to uniform, non-informative activations in the padded regions, ultimately diluting or biasing the model's learned features.

While z-normalization helps stabilize training by standardizing each series' distribution, it can inadvertently cause padded zeros to appear as legitimate (low-valued) data. Consequently, in variable-length TSC settings—where padding is unavoidable—the model may treat spurious regions as meaningful, obscuring true temporal patterns and impairing overall classification performance.

In certain controlled scenarios—such as when zero is a valid and meaningful measurement that naturally appears in the data—padding with zeros may still convey some structural information about sequence length. However, if zero values are frequent or hold no meaningful semantics in a given domain, padding risks merging authentic signals with artificially introduced noise, hindering the network's ability to differentiate between real observations and placeholders.

These findings highlight the need for careful consideration when applying padding and z-normalization together, especially for sequences of differing lengths. Despite the recognized benefits of z-normalization in accelerating convergence and promoting stable training, practitioners should remain mindful that zero-padding can confound learned feature representations—ultimately hindering the performance of 1D CNN-based TSC models such as InceptionTime [25]. See Figure 2.

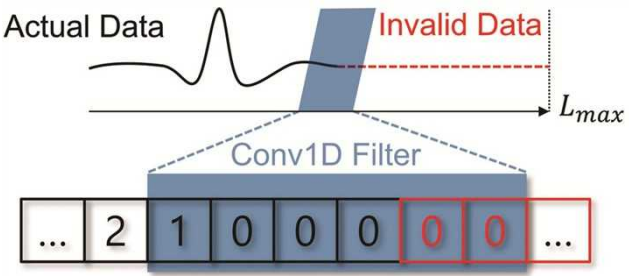


Figure 2. Illustration of the convolution operation with padding in a 1D CNN. Padding extends the series to the maximum length $L_{\max}$, introducing artificial values that can affect the feature maps and the performance of the CNN.

In addition to padding, other length uniformization techniques, such as interpolation and truncation, distort the data's inherent properties (Figure 1a,b). Interpolation can alter event timing within the series, and truncation may remove essential information. Although these techniques enable deep learning models to handle variable-length TSC, they compromise the integrity of the original time series representation, limiting the model's ability to capture critical characteristics.

While some modified 1D CNN architectures attempt to handle variable-length inputs through masking layers [26,27], these approaches remain limited in scope and are typically evaluated only against older baselines like DTW. Such masking-based strategies assume a certain level of temporal correlation between short and long sequences, which can lead to misrepresentations. In contrast, our method transforms time series data into a 2D representation and applies a two-dimensional CNN (2D-CNN), allowing convolutional models to capture key local patterns with inherent variability while preserving data integrity. Additionally, we compare our method to 10 baselines across 11 real-world benchmark datasets, positioning our approach as an accurate and effective solution for variable-length TSC.

3. Proposed Method

3.1. Preliminaries: Variable-Length TSC Problem Formulation

Let $\mathcal{D} = \{(\mathcal{X}, \mathcal{Y})\}$ be our dataset, where $\mathcal{X} = \{\mathbf{x}_n^{(1:L_n)}\}_{n=1}^{N}$ consists of variable-length time series data, with each series $\mathbf{x}_n^{(1:L_n)} := (\mathbf{x}_n^t)_{t=1}^{L_n} = (\mathbf{x}_n^1, ..., \mathbf{x}_n^{L_n})$. Each element $\mathbf{x}_n^t$ within a series lies in $\mathbb{R}^d$, where L_n denotes the length of the n-th series and d represents the number of variables or variates measured at each time point. We characterize $\mathcal{X}$ as a set of variable-length time series if there exist distinct indices $i, j \in \{1, \ldots, N\} \subset \mathbb{N}^+$ such that $L_i \neq L_j$. For simplicity in notation, we will initially focus on the univariate case ($d = 1$), where $\mathbf{x}_n^t$ is simplified to $x_n^t \in \mathbb{R}$. The primary objective of variable-length TSC is to

develop a model that assigns each variable-length times series $\mathbf{x}_n^{(1:L_n)}$ to one of the possible classes $y_n \in \mathcal{Y}$.

3.2. TIP: Time Series Into Pixels

To preserve original data from information distortion in variable-length time series classification (TSC), we introduce a transformation method called *Time series Into Pixels* (*TIP*). In this method, each time series value is mapped to a corresponding pixel using a binary mask: pixels representing the original data values are set to 1, while all other pixels are set to 0. Each time step in the series is mapped to its respective pixel by finding the appropriate horizontal axis index, with the vertical axis representing the time axis.

By transforming the time series into pixels, it eliminates the need to handle invalid values for variable-length time series, thereby preserving the original data representation while possessing the effective feature extraction capabilities of CNNs. The procedure is detailed in Algorithm 1.

Algorithm 1 TIP: Time series Into Pixels

Input: Variable-length time series $x_n^{(1:L_n)}$
Input: Maximum length L_{max}
Output: Transformed variable-length time series I_n
Parameter: Boundary constants β^-, β^+

1: Set the boundary constants to fit the time series data $x_n^{(1:L_n)}$ in $[\beta^-, \beta^+]$
2: $\mathrm{I}_n \leftarrow \mathbf{0}_{L_{max} \times L_{max}}$ ▷ Initialize image matrix
3: **for** each time step column index $i = 0, ..., L_{\max} - 1$ **do**
4: Find the corresponding row j using Equation (2)
5: Set $\mathrm{I}_n[i, j] = 1$ ▷ Mark the corresponding pixel
6: **end for**
7: **(optional):** For very long sequences, resize I_n with certain ratio
8: **(optional):** For multivariate case, repeat from step 1 to 6 by each variate and stack
9: **return** I_n

First, we scale the time series values to fit in the image; all time series data x_n^t for each time step $t \in [1, L_n]$ fit to the range $[\beta^-, \beta^+]$, where β^-, β^+ are boundary constants. This approach ensures that every time series value is effectively mapped to a corresponding position within the 2D representation, preserving the full range of the original data [28–30]. Then, we construct an image matrix I_n with dimensions $L_{max} \times L_{max}$. Each column of I_n corresponds to a time step in the time series, and each row corresponds to a value of x_n^t. To map the time series value to a corresponding row index $j \in \{0, \ldots, L_{max} - 1\}$, the range $[\beta^-, \beta^+]$ is divided into $L_{\max}$ bins, each covering an interval $[\beta^- + j\Delta, \beta^- + (j+1)\Delta]$. The incremental amount is calculated as $\Delta = \frac{\beta^+ - \beta^-}{L_{\max}}$. This discretization can be represented as:

$$[\beta^-, \beta^+] \rightarrow \{[\beta^-, \beta^- + \Delta], \ldots, [\beta^- + j\Delta, \beta^- + (j+1)\Delta], \ldots, [\beta^+ - \Delta, \beta^+]\} \quad (1)$$

Thus, the value at each time step x_n^t is mapped to a specific pixel in I_n according to:

$$x_n^t \mapsto I_n[i, j] \text{ where } x_n^t \in [\beta^- + j\Delta, \beta^- + (j+1)\Delta],\ i = t - 1, \text{ and } j \in [0, L_{max} - 1] \subset \mathbb{N}_0 \quad (2)$$

One of the notable strengths of our method is its dynamic adjustment of the representation of the original time series data through the hyperparameters β^-, β^+. Since the incremental value $\Delta = \frac{\beta^+ - \beta^-}{L_{\max}}$ between each discretized interval $[\beta^- + j\Delta, \beta^- + (j+1)\Delta]$ $= \left[\beta^- + j\left(\frac{\beta^+ - \beta^-}{L_{\max}}\right), \beta^- + (j+1)\left(\frac{\beta^+ - \beta^-}{L_{max}}\right)\right]$ is derived from β^-, β^+, a higher value of the range $[\beta^-, \beta^+]$ results in a wider range for each discretized interval, whereas a lower value

means a narrower range. This enables dynamic adjustments to the representation of time series data, allowing models to consider the appropriate scale or shape of it.

Unlike traditional techniques that rely on padding, interpolation, or truncation, our Time series Into Pixels (TIP) method transforms a variable-length time series into a two-dimensional binary pixel representation without distorting the underlying signal. Unlike existing variable-length TSC solutions that typically rely on one-dimensional architectures with masking layers or manual temporal alignment, TIP directly maps each original time series value to a unique pixel, effectively preserving the original timing, scale, and structure of the data. This representation simultaneously enables the use of vision models such as 2D-CNNs, which are known to be effective in capturing local patterns across spatial dimensions. Thus, TIP departs from prior approaches that often either compromise data integrity or require specialized modules to handle variable-length inputs.

3.3. 2D-CNN Backbone

To verify the effectiveness of our 2D representation without relying on the performance of a strong backbone model, we employed a very straightforward 2D CNN model similar to LeNet [31], the primitive CNN model. Like LeNet, our model consists of standard convolutional layers, pooling layers, and a fully connected layer. The primary differences between our model and LeNet are the inclusion of dropout layers and a global pooling layer, which are now commonly used to improve model robustness with lower computational cost. See Figure 3.

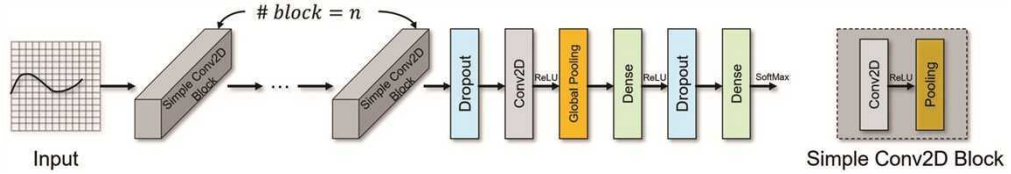


Figure 3. Illustration of our simple 2D CNN backbone. Detailed settings (e.g., kernel size, dense units) are provided in the Appendix A.

We used two baselines for TIP: ***Not-Refined TIP*** and ***Refined TIP***. Not-Refined TIP uses the same architecture and hyperparameters (i.e., batch size, learning rate, $\beta = 4$) across all datasets, hence the name. Refined TIP, on the other hand, involves tuning β and all hyperparameters of the simple backbone model, such as the number of convolutional layers and the units within specific ranges. Despite these refinements, it retains only the basic features of LeNet, including convolutional layers, pooling layers, and dense layers. Detailed explanations of our backbone are provided in Appendix A.

4. Experiment

4.1. Datasets and Metrics

To evaluate the effectiveness of our method on real-world datasets, we conducted experiments on 11 variable-length time series datasets from the UCR Time Series Classification Archive (https://www.cs.ucr.edu/~eamonn/time_series_data_2018/, accessed on 10 July 2024), one of the most widely used benchmarks in the TSC domain. For evaluation metrics, we used accuracy and macro average precision (MAP) since all datasets are class-balanced. Detailed descriptions of the datasets are available in Appendix A.

4.2. Baselines for Comparison

We compared our method with the leading existing solutions for variable-length TSC, as introduced in Section 2. We selected the top-performing methods from each category.

Below is a brief description of the methods used in our comparison and how each method handles variable-length inputs:

4.2.1. Non-DL Methods

- ED-1NN: A simple distance-based baseline using Euclidean distance for classification. Often used as a reference due to its simplicity. To handle variable-length sequences, ED-1NN typically requires padding or truncation of time series to a fixed length. We chose zero-padding here.
- DTW-1NN [18]: Long considered the gold standard for TSC, DTW-1NN excels at handling misaligned time series and has remained difficult to surpass [32]. DTW inherently handles variable-length sequences by allowing elastic shifts in the time axis, effectively aligning sequences of different lengths without the need for padding.
- TEASER [33]: An early classifier that efficiently handles variable-length inputs without extensive preprocessing. TEASER efficiently processes variable-length time series data by identifying critical patterns early in the sequence, without being influenced by the overall sequence length.
- RDST [34]: A shapelet-based method known for extracting key local patterns, providing one of the state-of-the-art performances on TSC [8]. RDST manages variable-length sequences by identifying and utilizing shapelets of different lengths, allowing it to flexibly match and classify time series without requiring uniform sequence lengths.

4.2.2. Masking Methods

- Vanilla RNN [35]: A basic recurrent model using masking to handle variable-length sequences, though surpassed by more advanced architectures. Masking involves padding shorter sequences and using a mask to indicate the actual length, ensuring that the RNN processes only the valid time steps during training and inference.
- BiLSTM [36]: Bidirectional LSTM, leveraging masking and bidirectional processing, and offering more robust results than RNN. Similar to vanilla RNN, BiLSTM uses masking to manage variable-length inputs, allowing the model to capture dependencies in both forward and backward directions without being affected by the padded values.
- TST [37]: A transformer-based model using masking, with competitive performance across diverse time series datasets. In TST, masking is employed to handle variable-length sequences by padding shorter sequences and applying attention masks to prevent the model from attending to padded positions, thereby maintaining the integrity of the actual data.

4.2.3. 1D-CNN Methods

- FCN [38]: A fully convolutional model that efficiently extracts spatial patterns, widely used for TSC tasks. To handle variable-length sequences, FCN typically applies preprocessing steps such as padding, truncation, or interpolation to convert all input time series to a fixed length.
- ResNet [38]: A deep architecture adapted from image classification, known for its strong results in TSC [39]. Like FCN, ResNet also manages variable-length time series by incorporating preprocessing steps like padding, truncation, or interpolation to standardize input lengths.
- InceptionTime [25]: A top-performing deep learning model for TSC, consistently demonstrating superior performance with its multi-scale architecture, as highlighted in recent reviews [8]. We selected this baseline for its robustness, though variants like LITETime [40] offer reduced computation time while maintaining the same performance as the original. InceptionTime handles variable-length inputs by utilizing

preprocessing techniques such as padding, truncation, or interpolation to achieve a fixed input size.

As evaluated in a recent review [8], other methods such as [41–43] have also demonstrated strong performance in TSC. However, we excluded these methods because, when compared to the baselines we selected, the performance differences were negligible, and they were either computationally too expensive or not applicable to variable-length TSC due to fixed size input constraint.

4.3. Implementation Details

Following prior publications [18,25] and the recommendations from the UCR Archive [19,24], all experiments used the default training and test splits provided by the UCR Archive, and each series was z-normalized. We implemented the best architectures reported in the literature, such as using five classifiers in the ensemble for InceptionTime [25] and 10,000 shapelets for RDST [34].

Since few studies adopt RNN-based models for TSC, we developed the vanilla RNN and BiLSTM in-house. We provide the detailed implementations of the two models in Appendix A. In addition to these models, we used implementations available in aeon (https://www.aeon-toolkit.org, accessed on 10 July 2024) and tsai (https://timeseriesai.github.io/tsai, accessed on 10 July 2024), the popular third-party libraries in time series analysis.

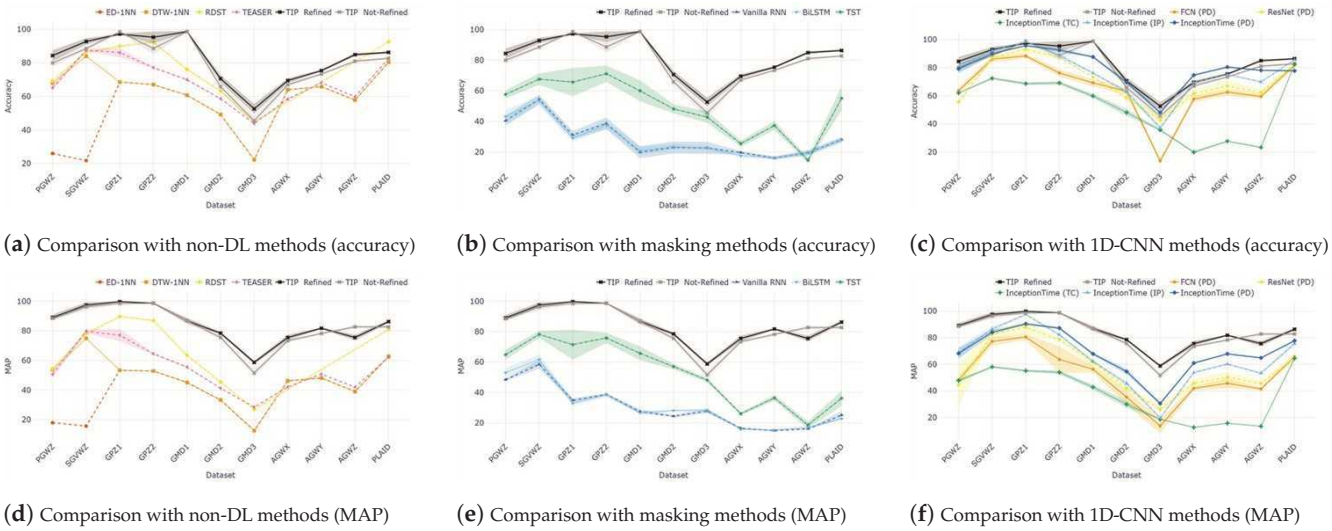
For 1D-CNN methods, we applied various length uniformization techniques, including truncation (TC), interpolation (IP), and padding (PD). For the most powerful 1D-CNN-based method, InceptionTime, we applied all three length uniformization techniques, while for other 1D-CNN baselines, we used only padding, as it yielded the best performance.

For all deep learning-based baselines, we added an early-stopping strategy and a plateau learning rate scheduler to avoid overfitting and getting stuck in local optima. These strategies are often omitted in the related literature despite their importance. By implementing these strategies and fine-tuning the batch size and learning rate over sufficient trials, we achieved improved performance for the baselines compared to their reported results. Additional implementation details, including the search space for fine-tuning and training strategy, are provided in Appendix A. All models were trained using the TensorFlow framework on a single NVIDIA RTX 2080 Ti GPU (NVIDIA Corporation, Santa Clara, CA, USA). We have made the complete source code available.

4.4. Results

As demonstrated in Figure 4, our approach consistently achieves the best or second-best results in both accuracy and MAP. Specifically, non-DL methods exhibited inferior performance due to the separation between configuring the feature space and the downstream classification task. In particular, ED-1NN recorded significantly lower performance than DTW-1NN because it failed to capture key aspects of variable-length sequences, such as speed variation. Although TEASER and RDST showed relatively decent performance, the aforementioned two-step learning process hindered the optimal configuration of the feature space for classification.

Similarly, recurrent models and transformers also demonstrated subpar performance. While these models maintain data integrity through the use of masking, their classification performance remains low. Beyond the traditional reasons for their low performance in TSC [14], their forward structures are inadequate for classification tasks where key local patterns are often position-variant, making it difficult to effectively capture the underlying dynamics. Although TST outperformed both recurrent models and transformers, it still lagged behind 1D-CNN-based methods and our convolutional model-based method, TIP, which excel at learning local patterns.



(**a**) Comparison with non-DL methods (accuracy) (**b**) Comparison with masking methods (accuracy) (**c**) Comparison with 1D-CNN methods (accuracy)

(**d**) Comparison with non-DL methods (MAP) (**e**) Comparison with masking methods (MAP) (**f**) Comparison with 1D-CNN methods (MAP)

Figure 4. Comparison to existing variable-length TSC solutions on 11 UCR real-world benchmarks, evaluated by (**a**–**c**) Accuracy (%) and (**d**–**f**) Macro Average Precision (MAP), with standard deviations computed across five seeds.

Lastly, 1D-CNN-based models generally performed well; however, our method, TIP, outperformed them across most datasets. Notably, InceptionTime, which employs truncation as a preprocessing step, achieved the lowest performance despite being the latest and ostensibly the best-performing algorithm in its category. This result indicates that truncation as a preprocessing method significantly distorts the original data information. When padding, instead of truncation, was applied, InceptionTime, ResNet, and FCN showed progressively higher performance. This improvement can be attributed to their complex structures: InceptionTime utilizes ensemble techniques and inception blocks, which are modern feature extraction methods; ResNet incorporates residual connections; and FCN consists solely of convolutional layers. Additionally, padding generally outperformed interpolation, likely because interpolation excessively alters the temporal dynamics of the original data. However, overall, these imputation methods using artificial values resulted in lower performance compared to our method, which preserves data integrity by employing a straightforward vision model with a 2D-CNN backbone.

5. Additional Analysis

5.1. What About in Multivariate Setting?

To evaluate the effectiveness of our method in multivariate setting, we conducted experiments on variable-length and multivariate real-world benchmark datasets from the UCI Archive [44]: JapaneseVowels and CharacterTrajectories. For this analysis, each variate time series data was converted into an image and then stacked (i.e., three variates = three-channel images). Note that we continued to utilize the simple 2D CNN as a backbone for this analysis. Unlike other baselines, this model does not capture the relationships between variables, which is crucial in multivariate TSC.

As shown in Table 2, our method achieves performance comparable to other methods, despite lacking features specifically designed to capture spatial patterns between variables. While InceptionTime, the current state-of-the-art TSC integrated with length uniformization techniques, achieved the highest accuracy, our method delivered a comparable accuracy score and outperformed it in MAP. Nevertheless, our approach may face computational challenges when scaling to extremely large datasets or handling high-dimensional multivariate settings. This is primarily because each variate is treated as a separate channel and stacked, which could increase memory and processing demands. However, our results

suggest that even under these constraints, the method remains effective in maintaining high performance and robustness for complex multivariate TSC problem.

Table 2. A performance comparison with existing solutions for variable-length TSC in multivariate settings. **Best results** are highlighted in **bold**. Note that our backbone lacks the ability to capture spatial patterns and the relationships between variables.

Dataset	**JapaneseVowels**		**CharacterTrajectories**	
# Variates	12		3	
Lengths	[7,29]		[60,182]	
Train/Test	270/370		1422/1436	
Metrics	Accuracy	MAP	Accuracy	MAP
DTW-1NN	96.93	94.01	66.22	48.46
Vanilla RNN	25.52 ± 0.50	22.44 ± 0.24	22.76 ± 3.48	20.99 ± 0.58
FCN (PD)	95.10 ± 1.36	90.43 ± 2.47	76.43 ± 1.87	61.53 ± 2.11
ResNet (PD)	98.75 ± 0.16	97.44 ± 0.33	78.38 ± 2.03	65.20 ± 1.81
InceptionTime (IP)	99.21 ± 0.03	98.40 ± 0.06	**82.81** ± 0.40	69.28 ± 0.45
InceptionTime (PD)	**99.39** ± 0.05	98.73 ± 0.10	82.43 ± 0.34	68.67 ± 0.55
Not Refined TIP	98.43 ± 0.07	99.66 ± 0.03	55.30 ± 2.99	58.74 ± 2.60
Refined TIP	98.64 ± 0.23	**99.69** ± 0.05	78.11 ± 1.18	**83.35** ± 0.83

5.2. Is Preserving the Original Representation Really a Matter in Variable-Length TSC?

To investigate the impact of information distortion caused by length uniformization techniques in variable-length TSC, we conducted an ablation study of our method. We assessed performance on the PGWZ dataset using Not-Refined TIP with four different strategies for length uniformization: zero-padding, interpolation, truncation, and no adjustment (Default), which is our approach.

As shown in Table 3, the performance in terms of accuracy and MAP significantly dropped when using truncation, interpolation, and padding compared to no length uniformization techniques. This demonstrates that distorting the representation of time series data has a critical impact on model performance.

Table 3. An ablation study comparing three length uniformization techniques—truncation (TC), interpolation (IP), and padding (PD)—and their impact on information distortion in the original representation, as opposed to our technique, which introduces no information distortion.

Strategy	**Accuracy**	**MAP**
TIP (TC)	46.80 ± 1.60	58.13 ± 0.91
TIP (IP)	72.80 ± 2.04	82.81 ± 3.40
TIP (PD)	78.00 ± 2.83	85.78 ± 1.96
TIP (Default)	**80.00 ± 2.19**	**88.57 ± 0.36**

6. Conclusions and Discussion

This work highlights the challenges inherent in applying current deep learning methods to the variable-length TSC problem. To address these challenges, we propose TIP, a novel approach that directly maps time series data into 2D image representations. We believe our findings contribute significantly to advancing the handling of variable-length time series, which represent the most natural form of time series data encountered in real-world applications.

While our results in variable-length TSC are promising, several issues remain unresolved. One notable concern is the potential for information loss when resizing series, particularly for long sequences—a challenge that is pervasive and yet unresolved in current

TSC methods. Furthermore, our method treats each variate as a separate channel, which could pose challenges when dealing with high-dimensional data or sparse datasets. Such scenarios may result in computational inefficiencies or difficulty capturing the complex interdependencies between variates. To address these challenges, future research could explore methods to better preserve interdependencies in multivariate time series. For example, instead of treating each variate as a distinct channel, randomly positioning multivariate images into a shared channel stack could allow the backbone vision model to capture intricate relationships between variables more effectively. Alternatively, incorporating feature extraction or attention mechanisms tailored to multivariate data may enhance both computational efficiency and performance. Resolving these limitations remains an open area for further investigation, which we aim to address in future work.

Author Contributions: Conceptualization, H.L.; methodology, H.L.; software, H.L.; validation, H.L. and D.S.; formal analysis, H.L.; investigation, H.L.; resources, D.S.; data curation, H.L.; writing—original draft preparation, H.L.; writing—review and editing, H.L. and D.S.; visualization, H.L.; supervision, D.S.; project administration, H.L.; funding acquisition, D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the reported results in this study are publicly available. All datasets used in this research are accessible from the UCR Time Series Classification Archive at https://www.cs.ucr.edu/~eamonn/time_series_data_2018/ (accessed on 10 July 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Experimental Details

Appendix A.1. Datasets Description

The datasets used in this study are from the UCR Time Series Archive [15], a widely utilized resource in time series research. Originally introduced in 2002 with 16 datasets, the archive has grown to encompass 128 datasets, making it a critical benchmark in the field. Among these, 11 datasets are specifically designed for variable-length time series, presenting unique challenges for classification methods. A detailed description of these datasets can be found in Table A1.

One example of such a dataset is GesturePebble, which showcases the variability in time series lengths. The data is derived from the z-axis readings of a 3-axis accelerometer in Pebble smartwatches, worn on participants' wrists as they perform six distinct gestures. The dataset consists of 304 gestures, with lengths varying naturally due to differences in individual movements. Two versions of the dataset were created:

- GesturePebbleZ1: The training set contains data from all participants in the first session, while the test set contains data from the second session.
- GesturePebbleZ2: Participants are split between the training and testing sets, making it more challenging due to the distinctiveness of individual gaits and movements.

For evaluation, we adhered to preprocessing methods, including z-normalization, as described in [25]. To ensure consistency with previous research, we used the default train/test splits provided by the UCR archive, following standard practices in the field. Additionally, each split was resampled using five random seeds to evaluate robustness. Table A1 provides an overview of the datasets used in this study.

Table A1. Descriptions of 11 UCR Benchmark Univariate Datasets with Variable Lengths.

Dataset	Train/Test	Range of Series Length	Class
PGWZ (PickupGestureWiimoteZ)	50/50	[29, 361]	10
SGWZ (ShakeGestureWiimoteZ)	50/50	[40, 385]	10
GPZ1 (GesturePebbleZ1)	132/172	[100, 455]	6
GPZ2 (GesturePebbleZ2)	146/158	[100, 455]	6
GMAD1 (GestureMidAirD1)	208/130	[80, 360]	26
GMAD2 (GestureMidAirD2)	208/130	[80, 360]	26
GMAD3 (GestureMidAirD3)	208/130	[80, 360]	26
ADWX (AllGestureWiimoteX)	300/700	[2, 385]	10
ADWY (AllGestureWiimoteY)	300/700	[2, 385]	10
ADWZ (AllGestureWiimoteZ)	300/700	[2, 385]	10
PLAID	537/537	[100, 1344]	11

Appendix A.2. Implementing Baselines

Although most baselines were primarily adapted from aeon and tsai, well-known third-party libraries for time series analysis, we developed the vanilla RNN and BiLSTM baselines from scratch. These RNN-based solutions are rarely explored in TSC due to their suboptimal feature extraction capabilities.

To address the padding issue, we utilized a mask layer to differentiate the padded parts. We padded with a high random integer and set it as the mask value, allowing us to exclude padded elements from computation. For the model architecture, we stacked RNN layers for the vanilla RNN and bidirectional LSTM layers for the BiLSTM, followed by two dense layers. We used the Adam optimizer [45] and treated the number of hidden layers and units in dense layers as hyperparameters, allowing the model to adjust its complexity based on the dataset.

Appendix A.3. Hyperparameter Settings

To ensure a fair comparison between baselines, we fine-tuned each deep learning baseline (except Not-Refined TIP) with 30 trials. Given our limited computational resources, we used the best architectures reported in the related literature (e.g., using five classifiers for the ensemble in InceptionTime [25], and 10,000 shapelets for RDST [34]). For all deep learning baselines, we fine-tuned their batch size and learning rate for each dataset, with the batch size increasing in powers of two starting from 4, and setting the maximum batch size to 32. For the learning rate, we used float numbers between 1×10^{-5} and 1×10^{-3} with logarithmic scaling.

For RNN-based methods, we additionally fine-tuned the number of units in dense layers, choosing from 64, 128, and 256, and the number of hidden layers, ranging from 2 to 5.

For the Not-Refined TIP model, no hyperparameter fine-tuning was performed. The specific hyperparameters used are as follows: $\beta = 4$, batch size = 32, learning rate = 0.0001, dense units = 256, dropout rate = 0.5, number of Conv2D layers = 5, number of Conv2D filters = 64, kernel size = 9×9, pooling type = MaxPooling2D, and global pooling type = GlobalMaxPooling2D.

In contrast, for the Refined TIP model, we fine-tuned all hyperparameters. The range for β was set between 3 and 6. The dense units were fine-tuned with values of 64, 128, and 256. The number of Conv2D layers was adjusted between 3 and 7, while the number of filters was chosen from 16, 32, and 64. The filter size was varied from 3×3 to 11×11 (odd numbers). The pooling type was selected between MaxPooling and AveragePooling. The batch size and learning rate were fine-tuned within the same ranges as the other compared baselines. We include all hyperparameters for every dataset in the code.

Using a simple 2D CNN model as the backbone, these refinements were found to be acceptable. As mentioned in the main text, apart from the inclusion of global pooling layers and dropout to enhance model robustness with reduced computational cost, no advanced techniques were incorporated into our 2D CNN backbone model beyond those derived from LeNet.

Appendix A.4. Training Strategy

Since the UCR archive provides default splits for the training and test sets, and prior publications follow this convention, we ensured consistency by using the same default splits. We used the test sets, instead of splitting the training set into a validation set, to refine each deep learning baseline, applying this method identically to all baselines. This approach is acceptable for comparative purposes.

For the deep learning baselines, we implemented an early-stopping strategy and a plateau learning rate scheduler across all methods, a detail often omitted in the related literature. We set the patience for early stopping at 50 epochs and for the plateau learning rate scheduler at 15 epochs, with a reduction factor of 0.8 and a minimum learning rate of 1×10^{-8}. We set the maximum number of epochs for both fine-tuning and evaluation at 1000, and confirmed that all baselines converged in fewer than 1000 epochs. Notably, our implementation of these methods and extensive trials resulted in higher performance scores than those reported in the related literature. The entire training procedures for each baseline are provided in our code.

Table A2. Comparison to non-DL solutions on 11 UCR real-world benchmarks, evaluated by accuracy, with standard deviations computed across five seeds. The **highest-performing results** are emphasized in **bold**, while the second-highest are underlined. "NA" indicates that the corresponding method is not applicable on this dataset.

Dataset	ED-1NN	DTW-1NN	RDST	TEASER	Not Refined TIP	Refined TIP
PGWZ	26.00	68.00	69.20 ± 0.98	65.20 ± 1.79	80.00 ± 2.22	**84.40 ± 3.20**
SGWZ	21.75	84.00	86.40 ± 0.80	87.60 ± 0.89	88.57 ± 1.13	**92.80 ± 1.94**
GPZ1	68.60	68.60	90.00 ± 0.77	86.16 ± 2.71	**98.60 ± 0.28**	97.21 ± 0.44
GPZ2	67.09	67.09	92.41 ± 0.57	77.22 ± 0.45	88.61 ± 3.61	**95.32 ± 3.42**
GMAD1	60.77	60.77	76.15 ± 0.96	70.00 ± 0.54	**98.66 ± 0.45**	**98.66 ± 0.41**
GMAD2	49.23	49.23	63.23 ± 0.58	58.62 ± 0.34	65.85 ± 2.34	**70.62 ± 2.25**
GMAD3	22.31	22.31	44.77 ± 0.58	43.85	45.54 ± 2.94	**52.62 ± 2.94**
ADWX	64.00	64.00	NA	58.43 ± 0.74	67.03 ± 1.42	**69.49 ± 1.42**
ADWY	65.86	65.86	NA	68.23 ± 0.13	73.36 ± 0.61	**75.40 ± 0.72**
ADWZ	57.86	57.86	NA	59.83	81.04 ± 0.98	**84.94 ± 0.94**
PLAID	80.45	80.45	**92.59 ± 0.22**	82.79 ± 0.47	82.75 ± 0.67	86.32 ± 0.69

Appendix A.5. Comparison to Existing Variable-Length TSC Solutions on 11 UCR Real-World Benchmarks

Here, we included the quantitative details for the contents in Figure 4. for better comparison.

Table A3. Comparison to non-DL solution on 11 UCR real-world benchmarks, evaluated by mean average precision (MAP), with standard deviations computed across five seeds. The **highest-performing results** are emphasized in **bold**, while the second-highest are underlined. "NA" indicates that the corresponding method is not applicable on this dataset.

Dataset	ED-1NN	DTW-1NN	RDST	TEASER	Not Refined TIP	Refined TIP
PGWZ	18.00	53.88	54.39 ± 1.16	50.71 ± 2.42	88.57 ± 0.36	**89.10 ± 1.94**
SGWZ	15.75	74.96	78.09 ± 1.16	79.83 ± 1.47	96.28 ± 2.44	**97.48 ± 1.81**
GPZ1	53.41	53.41	89.76 ± 0.80	77.07 ± 4.14	98.55 ± 0.41	**99.77 ± 0.14**

Table A3. *Cont.*

Dataset	ED-1NN	DTW-1NN	RDST	TEASER	Not Refined TIP	Refined TIP
GPZ2	52.96	52.96	87.01 ± 0.85	64.59 ± 0.67	98.66 ± 0.45	**98.68 ± 0.41**
GMAD1	45.19	45.19	63.50 ± 1.08	55.55 ± 0.68	**86.87 ± 1.82**	**86.87 ± 1.82**
GMAD2	33.36	33.36	45.52 ± 0.51	41.30 ± 0.13	75.57 ± 1.30	**78.51 ± 0.93**
GMAD3	12.68	12.68	27.00 ± 0.53	28.50 ± 0.02	51.64 ± 2.40	**58.76 ± 1.20**
ADWX	46.26	46.26	NA	42.18 ± 0.51	73.36 ± 1.07	**75.55 ± 1.95**
ADWY	48.20	48.20	NA	50.82 ± 0.23	78.24 ± 0.90	**81.79 ± 0.43**
ADWZ	39.08	39.08	NA	42.05	**82.75 ± 1.30**	75.55 ± 1.95
PLAID	62.84	62.84	80.94 ± 0.80	62.28 ± 0.91	82.75 ± 1.00	**86.32 ± 0.99**

Table A4. Comparison to existing masking-based solutions on 11 UCR real-world benchmarks, evaluated by accuracy, with standard deviations computed across five seeds. The **highest-performing results** are emphasized in **bold**, while the second-highest are underlined. "NA" indicates that the corresponding method is not applicable on this dataset.

Dataset	VanillaRNN	BiLSTM	TST	Not Refined TIP	Refined TIP
PGWZ	40.40 ± 3.44	43.20 ± 4.31	57.60 ± 2.76	80.00 ± 2.22	**84.40 ± 3.20**
SGWZ	54.40 ± 2.81	53.60 ± 3.45	67.60 ± 3.78	88.57 ± 1.13	**92.80 ± 1.94**
GPZ1	31.16 ± 2.69	29.07 ± 2.57	65.70 ± 9.20	**98.60 ± 0.28**	97.21 ± 0.44
GPZ2	38.61 ± 3.89	37.59 ± 2.21	71.14 ± 5.40	88.61 ± 3.61	**95.32 ± 3.42**
GMAD1	19.85 ± 3.98	20.77 ± 2.56	60.00 ± 6.83	**98.66 ± 0.45**	**98.66 ± 0.41**
GMAD2	22.92 ± 3.89	23.38 ± 1.98	48.15 ± 3.00	65.85 ± 2.34	**70.62 ± 2.25**
GMAD3	22.62 ± 3.98	22.62 ± 1.43	42.77 ± 3.75	45.54 ± 2.94	**52.62 ± 2.94**
ADWX	19.63 ± 1.04	17.60 ± 1.37	25.43 ± 1.95	67.03 ± 1.42	**69.49 ± 1.42**
ADWY	16.03 ± 1.14	16.11 ± 1.14	37.29 ± 2.93	73.36 ± 0.61	**75.40 ± 0.72**
ADWZ	19.46 ± 1.65	19.46 ± 1.65	14.54 ± 1.67	81.04 ± 0.98	**84.94 ± 0.94**
PLAID	28.08 ± 1.65	27.78 ± 1.83	55.12 ± 7.84	82.75 ± 0.67	**86.32 ± 0.69**

Table A5. Comparison to existing masking-based solutions on 11 UCR real-world benchmarks, evaluated by mean average precision (MAP), with standard deviations computed across five seeds. The **highest-performing results** are emphasized in **bold**, while the second-highest are underlined. "NA" indicates that the corresponding method is not applicable on this dataset.

Dataset	VanillaRNN	BiLSTM	TST	Not Refined TIP	Refined TIP
PGWZ	48.43 ± 0.90	53.04 ± 2.91	64.88 ± 3.26	88.57 ± 0.36	**89.10 ± 1.94**
SGWZ	58.48 ± 2.73	61.68 ± 3.45	78.21 ± 2.31	96.28 ± 2.44	**97.48 ± 1.81**
GPZ1	34.82 ± 1.40	32.88 ± 1.67	71.51 ± 9.80	98.55 ± 0.41	**99.77 ± 0.14**
GPZ2	38.68 ± 0.80	38.41 ± 1.44	75.90 ± 3.44	98.66 ± 0.45	**98.68 ± 0.41**
GMAD1	27.47 ± 1.37	26.64 ± 0.98	65.78 ± 4.56	**86.87 ± 1.82**	**86.87 ± 1.82**
GMAD2	24.41 ± 0.82	28.11 ± 0.79	57.05 ± 1.83	75.57 ± 1.30	**78.51 ± 0.93**
GMAD3	27.73 ± 1.41	28.38 ± 1.28	48.14 ± 1.35	51.64 ± 2.40	**58.76 ± 1.20**
ADWX	16.45 ± 0.48	15.72 ± 0.52	26.00 ± 1.06	73.36 ± 1.07	**75.55 ± 1.95**
ADWY	14.90 ± 0.29	15.35 ± 0.34	36.39 ± 1.84	78.24 ± 0.90	**81.79 ± 0.43**
ADWZ	16.03 ± 0.37	16.93 ± 0.48	18.60 ± 2.34	**82.75 ± 1.30**	75.55 ± 1.95
PLAID	25.07 ± 2.78	22.77 ± 0.32	36.18 ± 5.01	82.75 ± 1.00	**86.32 ± 0.99**

Table A6. Comparison to 1D-CNN-based with preprocssing solutions on 11 UCR real-world benchmarks, evaluated by accuracy, with standard deviations computed across five seeds. The **highest-performing results** are emphasized in **bold**, while the second-highest are underlined. "NA" indicates that the corresponding method is not applicable on this dataset.

Dataset	FCN (PD)	ResNet (PD)	InceptionTime (TC)	InceptionTime (PD)	InceptionTime (IP)	Not Refined TIP	Refined TIP
PGWZ	63.60 ± 2.65	55.60 ± 2.05	62.00 ± 0.23	79.20 ± 3.25	79.60 ± 3.25	80.00 ± 2.22	**84.40 ± 3.20**
SGWZ	86.00 ± 2.19	87.00 ± 1.79	72.40 ± 0.80	90.00 ± 1.13	92.00 ± 1.79	88.57 ± 1.13	**92.80 ± 1.94**
GPZ1	88.26 ± 1.70	93.02 ± 2.96	68.60 ± 1.10	95.60 ± 0.44	**98.60 ± 0.28**	**98.60 ± 0.28**	97.21 ± 0.44
GPZ2	76.20 ± 2.56	87.22 ± 0.74	69.11 ± 1.23	92.41 ± 1.69	88.61 ± 1.80	88.61 ± 3.61	**95.32 ± 3.42**
GMAD1	69.23 ± 2.68	73.38 ± 0.45	59.85 ± 1.49	87.61 ± 0.65	76.15 ± 0.58	**98.66 ± 0.45**	**98.66 ± 0.41**
GMAD2	63.23 ± 2.48	58.57 ± 2.09	48.15 ± 2.51	69.54 ± 1.84	62.92 ± 1.79	65.85 ± 2.34	**70.62 ± 2.25**
GMAD3	13.72 ± 1.72	42.77 ± 3.43	35.69 ± 0.62	48.15 ± 1.34	37.08 ± 1.75	45.54 ± 2.94	**52.62 ± 2.94**
ADWX	57.63 ± 2.34	61.57 ± 3.05	19.94 ± 0.87	**74.69 ± 0.67**	69.46 ± 0.67	67.03 ± 1.42	69.49 ± 1.42
ADWY	62.57 ± 2.21	66.83 ± 3.79	27.74 ± 0.44	**80.37 ± 0.72**	75.20 ± 0.50	73.36 ± 0.61	75.40 ± 0.72
ADWZ	59.48 ± 1.94	62.37 ± 2.35	23.34 ± 0.47	78.11 ± 0.56	69.86 ± 0.49	81.04 ± 0.98	**84.94 ± 0.94**
PLAID	81.75 ± 1.64	82.16 ± 1.85	82.35 ± 0.25	77.71 ± 0.67	85.52 ± 0.55	82.75 ± 0.67	**86.32 ± 0.69**

Table A7. Comparison to 1D-CNN-based with preprocssing solutions on 11 UCR real-world benchmarks, evaluated by mean average precision (MAP), with standard deviations computed across five seeds. The **highest-performing results** are emphasized in **bold**, while the second-highest are underlined. "NA" indicates that the corresponding method is not applicable on this dataset.

Dataset	FCN (PD)	ResNet (PD)	InceptionTime (TC)	InceptionTime (PD)	InceptionTime (IP)	Not Refined TIP	Refined TIP
PGWZ	48.32 ± 2.67	44.43 ± 15.77	47.75 ± 1.36	68.15 ± 4.13	68.80 ± 4.30	88.57 ± 0.36	**89.10 ± 1.94**
SGWZ	77.21 ± 3.27	82.59 ± 3.04	58.04 ± 0.85	84.09 ± 1.93	86.57 ± 2.03	96.28 ± 2.44	**97.48 ± 1.81**
GPZ1	80.50 ± 2.37	87.81 ± 4.79	55.16 ± 0.85	90.34 ± 0.73	97.42 ± 0.51	98.55 ± 0.41	**99.77 ± 0.14**
GPZ2	63.61 ± 9.93	78.49 ± 1.45	53.97 ± 1.54	87.16 ± 1.09	82.16 ± 1.09	98.66 ± 0.45	**98.68 ± 0.41**
GMAD1	56.29 ± 2.68	61.62 ± 3.50	42.80 ± 1.85	67.82 ± 1.24	62.14 ± 1.00	**86.87 ± 1.82**	**86.87 ± 1.82**
GMAD2	35.29 ± 5.54	41.89 ± 2.31	29.92 ± 2.69	54.60 ± 2.26	45.62 ± 1.79	75.57 ± 1.30	**78.51 ± 0.93**
GMAD3	13.72 ± 5.61	26.18 ± 2.49	18.68 ± 0.32	30.56 ± 1.12	19.69 ± 0.76	51.64 ± 2.40	**58.76 ± 1.20**
ADWX	41.89 ± 1.82	45.78 ± 3.22	12.62 ± 0.33	60.92 ± 0.94	53.80 ± 0.98	73.36 ± 1.07	**75.55 ± 1.95**
ADWY	45.77 ± 3.67	50.33 ± 3.79	15.70 ± 0.17	67.81 ± 0.58	60.15 ± 0.68	78.24 ± 0.90	**81.79 ± 0.43**
ADWZ	41.51 ± 1.16	45.64 ± 1.98	13.42 ± 0.19	64.81 ± 0.68	53.35 ± 0.66	**82.75 ± 1.30**	75.55 ± 1.95
PLAID	65.64 ± 1.93	64.99 ± 2.97	64.47 ± 0.64	77.71 ± 0.67	75.55 ± 1.95	82.75 ± 1.00	**86.32 ± 0.99**

References

1. Bian, J.; Tian, D.; Tang, Y.; Tao, D. Trajectory data classification: A review. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 1–34. [CrossRef]
2. Norgaard, S.; Saeedi, R.; Gebremedhin, A.H. Multi-sensor time-series classification for activity tracking under variable length. *IEEE Sens. J.* **2019**, *20*, 2701–2709. [CrossRef]
3. Wang, Y.; Perry, M.; Whitlock, D.; Sutherland, J.W. Detecting anomalies in time series data from a manufacturing system using recurrent neural networks. *J. Manuf. Syst.* **2022**, *62*, 823–834. [CrossRef]
4. Farahani, M.A.; McCormick, M.; Harik, R.; Wuest, T. Time-series classification in smart manufacturing systems: An experimental evaluation of state-of-the-art machine learning algorithms. *Robot. Comput. Integr. Manuf.* **2025**, *91*, 102839. [CrossRef]
5. Kahveci, T.; Singh, A. Variable length queries for time series data. In Proceedings of the 17th International Conference on Data Engineering, Heidelberg, Germany, 2–6 April 2001; IEEE: Piscataway, NJ, USA, 2001; pp. 273–282.
6. Schäfer, P. Towards time series classification without human preprocessing. In Proceedings of the International Workshop on Machine Learning and Data Mining in Pattern Recognition, St. Petersburg, Russia, 21–24 July 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 228–242.
7. Tan, C.W.; Petitjean, F.; Keogh, E.; Webb, G.I. Time series classification for varying length series. *arXiv* **2019**, arXiv:1910.04341.
8. Middlehurst, M.; Schäfer, P.; Bagnall, A. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Min. Knowl. Discov.* **2024**, *38*, 1958–2031. [CrossRef]
9. Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv* **2022**, arXiv:2210.02186.
10. Wang, Y.; Wu, H.; Dong, J.; Liu, Y.; Long, M.; Wang, J. Deep time series models: A comprehensive survey and benchmark. *arXiv* **2024**, arXiv:2407.13278.
11. Shi, Y.; Ying, X.; Yang, J. Deep unsupervised domain adaptation with time series sensor data: A survey. *Sensors* **2022**, *22*, 5507. [CrossRef]

12. Spiegel, S.; Gaebler, J.; Lommatzsch, A.; De Luca, E.; Albayrak, S. Pattern recognition and classification for multivariate time series. In Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data, San Diego, CA, USA, 21 August 2011; pp. 34–42.
13. Tang, W.; Long, G.; Liu, L.; Zhou, T.; Blumenstein, M.; Jiang, J. Omni-scale cnns: A simple and effective kernel size configuration for time series classification. *arXiv* **2020**, arXiv:2002.10061.
14. Mohammadi Foumani, N.; Miller, L.; Tan, C.W.; Webb, G.I.; Forestier, G.; Salehi, M. Deep learning for time series classification and extrinsic regression: A current survey. *ACM Comput. Surv.* **2024**, *56*, 1–45. [CrossRef]
15. Dau, H.A.; Bagnall, A.; Kamgar, K.; Yeh, C.C.M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C.A.; Keogh, E. The UCR time series archive. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 1293–1305. [CrossRef]
16. Lines, J.; Davis, L.M.; Hills, J.; Bagnall, A. A shapelet transform for time series classification. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing China, 12–16 August 2012; pp. 289–297.
17. Schäfer, P. The BOSS is concerned with time series classification in the presence of noise. *Data Min. Knowl. Discov.* **2015**, *29*, 1505–1530. [CrossRef]
18. Keogh, E.; Ratanamahatana, C.A. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.* **2005**, *7*, 358–386. [CrossRef]
19. Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; Keogh, E. The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.* **2017**, *31*, 606–660. [CrossRef] [PubMed]
20. Cui, Z.; Chen, W.; Chen, Y. Multi-scale convolutional neural networks for time series classification. *arXiv* **2016**, arXiv:1603.06995.
21. Xiao, Q.; Wu, B.; Zhang, Y.; Liu, S.; Pechenizkiy, M.; Mocanu, E.; Mocanu, D.C. Dynamic sparse network for time series classification: Learning what to "see". *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 16849–16862.
22. Tang, Y.; Xu, J.; Matsumoto, K.; Ono, C. Sequence-to-sequence model with attention for time series classification. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, Spain, 12–15 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 503–510.
23. Ratanamahatana, C.A.; Keogh, E. Three myths about dynamic time warping data mining. In Proceedings of the 2005 SIAM International Conference on Data Mining, Newport Beach, CA, USA, 21–23 April 2005; SIAM: Philadelphia, PA, USA, 2005; pp. 506–510.
24. Rakthanmanon, T.; Campana, B.; Mueen, A.; Batista, G.; Westover, B.; Zhu, Q.; Zakaria, J.; Keogh, E. Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Trans. Knowl. Discov. Data* **2013**, *7*, 1–31. [CrossRef]
25. Ismail Fawaz, H.; Lucas, B.; Forestier, G.; Pelletier, C.; Schmidt, D.F.; Weber, J.; Webb, G.I.; Idoumghar, L.; Muller, P.A.; Petitjean, F. Inceptiontime: Finding alexnet for time series classification. *Data Min. Knowl. Discov.* **2020**, *34*, 1936–1962. [CrossRef]
26. Schneider, M.; Greifzu, N.; Wang, L.; Walther, C.; Wenzel, A.; Li, P. An end-to-end machine learning approach with explanation for time series with varying lengths. *Neural Comput. Appl.* **2024**, *36*, 7491–7508. [CrossRef]
27. Sawada, A.; Miyagawa, T.; Ebihara, A.; Yachida, S.; Hosoi, T. Convolutional neural networks for time-dependent classification of variable-length time series. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–8.
28. Hatami, N.; Gavet, Y.; Debayle, J. Classification of time-series images using deep convolutional neural networks. In Proceedings of the Tenth International Conference on Machine Vision (ICMV 2017), Vienna, Austria, 13–15 November 2018; SPIE: Bellingham, WA, USA, 2017; Volume 10696, pp. 242–249.
29. Xu, G.; Liu, M.; Jiang, Z.; Shen, W.; Huang, C. Online fault diagnosis method based on transfer convolutional neural networks. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 509–520. [CrossRef]
30. Wang, Z.; Oates, T. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In Proceedings of the Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
31. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
32. Wang, X.; Mueen, A.; Ding, H.; Trajcevski, G.; Scheuermann, P.; Keogh, E. Experimental comparison of representation methods and distance measures for time series data. *Data Min. Knowl. Discov.* **2013**, *26*, 275–309. [CrossRef]
33. Schäfer, P.; Leser, U. TEASER: early and accurate time series classification. *Data Min. Knowl. Discov.* **2020**, *34*, 1336–1362. [CrossRef]
34. Guillaume, A.; Vrain, C.; Elloumi, W. Random dilated shapelet transform: A new approach for time series shapelets. In Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence, Xiamen, China, 23–25 September 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 653–664.
35. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
36. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef]

37. Zerveas, G.; Jayaraman, S.; Patel, D.; Bhamidipaty, A.; Eickhoff, C. A transformer-based framework for multivariate time series representation learning. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 2114–2124.
38. Wang, Z.; Yan, W.; Oates, T. Time series classification from scratch with deep neural networks: A strong baseline. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1578–1585.
39. Ismail Fawaz, H.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Deep learning for time series classification: a review. *Data Min. Knowl. Discov.* **2019**, *33*, 917–963. [CrossRef]
40. Ismail-Fawaz, A.; Devanne, M.; Berretti, S.; Weber, J.; Forestier, G. Lite: Light inception with boosting techniques for time series classification. In Proceedings of the 2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA), Thessaloniki, Greece, 9–13 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–10.
41. Dempster, A.; Schmidt, D.F.; Webb, G.I. Hydra: Competing convolutional kernels for fast and accurate time series classification. *Data Min. Knowl. Discov.* **2023**, *37*, 1779–1805. [CrossRef]
42. Ismail-Fawaz, A.; Devanne, M.; Weber, J.; Forestier, G. Deep learning for time series classification using new hand-crafted convolution filters. In Proceedings of the 2022 IEEE International Conference on Big Data, Osaka, Japan, 17–20 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 972–981.
43. Middlehurst, M.; Large, J.; Flynn, M.; Lines, J.; Bostrom, A.; Bagnall, A. HIVE-COTE 2.0: a new meta ensemble for time series classification. *Mach. Learn.* **2021**, *110*, 3211–3243. [CrossRef]
44. Frank, A. UCI Machine Learning Repository. 2010. Available online: http://archive.ics.uci.edu/ml (accessed on 10 July 2024).
45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Machine Learning-Based Prediction of Well Logs Guided by Rock Physics and Its Interpretation

Ji Zhang, Guiping Liu, Zhen Wei, Shengge Li, Yeheya Zayier and Yuanfeng Cheng *

School of Geology and Mining Engineering, Xinjiang University, Urumqi 830046, China; 18799632186@163.com (J.Z.); ping0991@163.com (G.L.); weizhen@xju.edu.cn (Z.W.); shenggeli0428@163.com (S.L.); yeheya0926@163.com (Y.Z.)

* Correspondence: chengyuanfeng@aliyun.com

Abstract: The refinement of acquired well logs has traditionally relied on predefined rock physics models, albeit with their inherent limitations and assumptions. As an alternative, effective yet less explicit machine learning (ML) techniques have emerged. The integration of these two methodologies presents a promising new avenue. In our study, we used four ML algorithms: Random Forests (RF), Gradient Boosting Decision Trees (GBDT), Multilayer Perceptrons (MLP), and Linear Regression (LR), to predict porosity and clay volume fraction from well logs. Throughout the entire workflow, from feature engineering to outcome interpretation, our predictions are guided by rock physics principles, particularly the Gardner relations and the Larionov relations. Remarkably, while the predictions themselves are satisfactory, SHapley Additive exPlanations (SHAP) analysis uncovers consistent patterns across the four algorithms, irrespective of their distinct underlying structures. By juxtaposing the SHAP explanations with rock physics concepts, we discover that all four algorithms align closely with rock physics principles, adhering to its cause–effect relationships. Nonetheless, even after intentionally excluding crucial controlling input features that would inherently compromise prediction accuracy, all four ML algorithms and the SHAP analysis continue to operate, albeit in a manner that seems irrational and starkly contradicts the fundamental principles of rock physics. This integration strategy facilitates a transition from solely mathematical explanations to a more philosophical interpretation of ML-based predictions, effectively dismantling the traditional black box nature of these ML models.

Keywords: machine learning; well logs; rock physics; SHapley Additive exPlanations (SHAP); explainable artificial intelligence (XAI)

1. Introduction

Well logs are capable of capturing a diverse array of geophysical parameters within the penetrated formations, encompassing caliper measurements, acoustic velocity, gamma-ray radioactivity, neutron density, spontaneous potential, resistivity, nuclear magnetic resonance, and more. These vast well logging datasets are extensively employed in comprehensively analyzing the physical attributes of the target strata, offering indispensable insights into the geological composition and potential of hydrocarbon reservoirs [1,2].

Rock physics models are typically employed in the refinement of initially acquired well logs, aimed at correcting inaccuracies or predicting missing data segments [3,4]. These models bridge the gap between macroscopic measured seismic properties and microscopic rock features, such as porosity, resistivity, mineralogy, fluid saturation, and pore pressure [5–7].These models can be broadly categorized into two types: empirical and

physically based [8]. Physically grounded models, such as Gassmann's equation, often hinge on certain assumptions, thus restricting their scope of application [9–12]. In contrast, empirical models, exemplified here by the Gardner relations and the Larionov relations, are founded on experimental data and stem from straightforward regression analyses [13–17]. Despite their simplicity, these empirical models have demonstrated remarkable practicality and widespread applicability in the petroleum industry [18,19].

Alternatively, machine learning (ML) algorithms offer a solution for correcting errors or filling in gaps in well logs [20,21]. In recent years, the application of ML technology has experienced a dramatic surge in popularity and adoption across the logging industry, marking a significant leap in technological progress [22–25]. While rock physics relies on explicit knowledge gained through extensive experience and experimentation, ML algorithms depend on data training [26,27]. This often renders ML a black box with limited interpretability, significantly reducing its perceived enhanced value and credibility within the industry [28–31]. Explainable artificial intelligence (XAI) serves as an umbrella term, encompassing the incorporation of human expertise into ML workflows [32–35]. This integration spans a wide spectrum, from precisely defined equations (PINN) where human knowledge is explicit [36–38], to more implicit and less readily articulated experience that individuals possess [39,40]

In this study, we propose to integrate the SHapley Additive exPlanations (SHAP) framework and rock physics into the workflow for predicting well logs, leveraging four distinct ML algorithms. This approach strives to dismantle the traditional opacity of ML models, thereby fostering unprecedented transparency. Specifically, SHAP allows us to quantify, on a localized level, the precise impact of each input feature on individual predictions, while also presenting a holistic, global overview across the entire dataset [41–43]. Furthermore, rock physics serves as a pivotal guide, quality guardian, and reference point throughout the ML-based workflow, spanning from feature engineering to results interpretation. By leveraging this integration, we aim to not only bolster the accuracy but also elevate the interpretability of ML-based well log predictions, thus propelling the field forward towards more reliable and intelligible outcomes.

2. Methodology

2.1. Rock Physics

To describe the dependency between bulk density and compressional velocity, a set of relations were formulated as [16]

$$\rho_b = a V_p{}^b \tag{1}$$

where ρ_b is bulk density in g cm^{-3}, V_p is the P-wave velocity in ft s^{-1}, and a and b are lithology-specific constant coefficients that vary across different rock types.

In this study, the Gardner relations were initially used to calibrate the correlation between acoustic logs and density logs. Subsequently, these relations were used to generate density logs in cases of suspicion or inaccuracies. It is noteworthy that, besides acoustic data, density can also correlate with shear wave velocity, neutron porosity, or gamma ray [44]. Therefore, in addition to the Gardner relations, a machine learning-driven approach can be implemented to generate density logs from a diverse range of inputs.

To describe the clay volume fraction in a rock, the gamma index, I_c, was introduced [14], which is simply a normalized representation of the recorded nature gamma logs. Specifically,

$$I_c = \frac{GR - GR_{min}}{GR_{max} - GR_{min}} \tag{2}$$

where GR is the measured value of gamma logs expressed in American Petroleum Institute (API) units, and GR_{max} and GR_{min} are the respective maximum and minimum

values encountered in shales and sands, with these thresholds herein specified as 140 and 20 API units.

Since the gamma index tends to exceed the actual clay volume fraction, V_c, alternative modified equations are often employed [14]:

$$V_c = 0.083\left(2^{3.7I_c} - 1\right) \tag{3}$$

The formula holds applicability for both the Tertiary period and more recent geological epochs. The gamma logs capture the natural gamma radiation emanating from radioactive elements such potassium (K), uranium (U), and thorium (Th), which are primarily present in clays. However, it is worth noting that non-shale radioactive mineral like Sylvite, feldspars, and micas can also contribute to gamma logs, potentially introducing inaccuracies [45–48]. Therefore, for specific geological problems, the above equations may be further modified or machine learning algorithms can be employed, using either gamma logs alone or a combination of multivariate inputs including gamma logs, density, P-wave velocity, and acoustic impedance, among others.

2.2. Data

Nestled in the southern North Sea basin, the F3 block (54.8669° N, 4.8131° E) in the Dutch sector lies atop the Central Graben, bordered by the Ringkøbing-Fyn High to the east and the Mid-North Sea High to the west. Geologically, the Mid-Miocene Unconformity (MMU) serves as a demarcation line, separating the Cenozoic into the underlying Paleogene units and the overlying Neogene units [49]. The Paleogene units beneath have been structurally influenced by the syn-depositional halokinetic movements of the underlying Zechstein evaporates, manifesting in the formation of faults and drapes. In contrast, the Neogene units overlaying them consist primarily of coarse-grained prograding sediments with an overall high porosity of 22–30%. These sediments have formed immense polycyclic fluvio-deltaic systems, a direct response to high-frequency relative sea-level cycles [50]. Evidence of these cycles can be seen in large-scale sigmoidal beddings of truncations, with onlaps onto them to the east, and downlaps to the west [51,52]. Within this geological context, our study focuses specifically on a select interval of log data (FS2–FS6) obtained from four wells within the F3 block, providing insights into a subset of these remarkable fluvio-deltaic systems.

The 3D seismic data and log data for this study originate from the F3 block, with the seismic data acquired in 1987 by NAM for petroleum exploration purposes [53]. dGB Earth Sciences has kindly provided the upper segment of the seismic data, encompassing a time range of 0 to 1.848 s. The inline range ranges from 100 to 750, while the crossline range extends from 300 to 1250, both with a 25 m line spacing, covering a total area of 386.93 km^2. An inline is shown in Figure 1b. Four vertical wells inside the survey are provided: F02-1, F06-1, F03-2, and F03-4, all measured by NLOG. These wells are equipped with sonic and gamma ray logs. However, it is noteworthy that according to dGB's records, only F02-1 and F03-2 have density logs available. For F03-4 and F06-1, density logs were predicted using a neural network model trained on data from the first two wells, using sonic and gamma logs as input. Additionally, the original porosity logs for all wells were calculated from density data based on a linear formula [3,13]:

$$\phi = \frac{\rho_{ma} - \rho_b}{\rho_{ma} - \rho_{fl}} \tag{4}$$

where ϕ is porosity, and ρ_{ma} and ρ_{fl} are the densities of the matrix and mud filtrate respectively, with ρ_{ma} set to 2.65 g cm^{-3} and ρ_{fl} set to 1.05 g cm^{-3}. Furthermore,

for this study, the clay volume fraction was calculated from gamma logs using the Larionov relations.

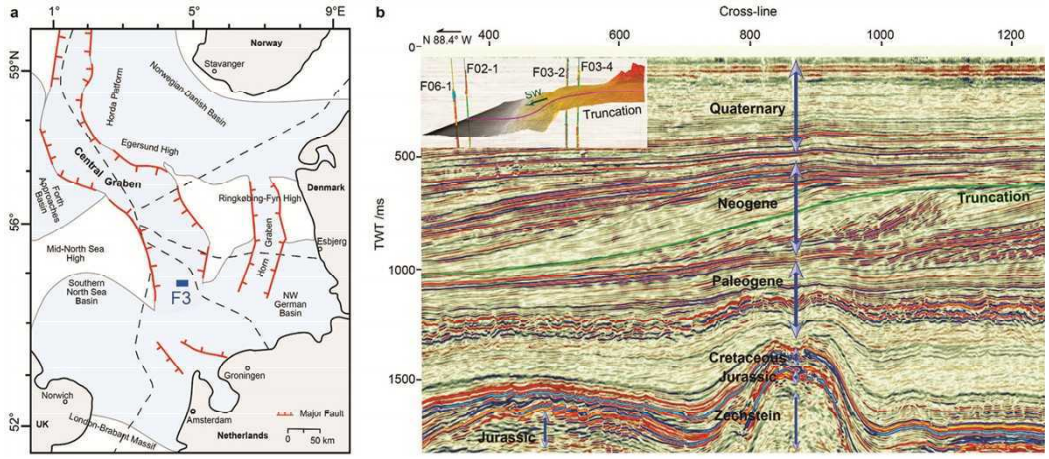


Figure 1. Geology setting of the F3 block. (**a**) Structural setting of the North Sea basin. The location of the F3 block is highlighted by a blue rectangle. (**b**) Seismic section of the F3 block (Inline 362). The geological periods of strata are annotated. The green line is the Truncation horizon. The inset is the Truncation horizon in 3D, with the four wells and the prograding direction annotated. The red line serves as the intersection between the horizon and the seismic section.

Figure 2 gives a comprehensive overview of the seven logs corresponding to the four distinct wells. For each well, a specific interval ranging from FS2 to FS6 is selected and presented. In comparison to F03-2 and F03-4, the depths recorded for F02-1 and F06-1 are notably deeper and shorter. This difference is due to their geographic position, which lies in the southwestern periphery of the survey area, precisely at the foreland of the prograding delta, as depicted in Figure 1.

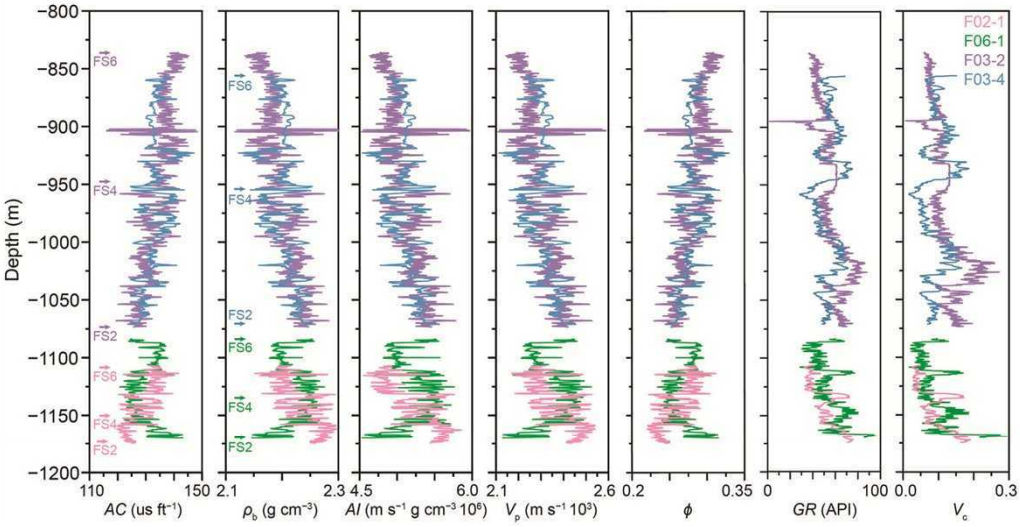


Figure 2. Log data obtained from the four wells. From **left** to **right**: acoustic, bulk density, acoustic impedance, P-wave velocity, porosity, gamma, clay volume. The top of each interval in each well is denoted using the same color as the line of the logs.

2.3. Machine Learning

Four different popular ML algorithms were used, comprising Linear Regression (LR), Multi-Layer Perceptron (MLP), Random Forests (RF), and Gradient Boosting Decision Tree (GBDT). LR, without any regularization, as the first type of regression analysis to undergo rigorous study and extensive practical application, is widely used owing to its simplicity, interpretability, and ease of implementation [54]. MLP, a commonly applied artificial

neural network model, boasts robust expressive and generalization capabilities, enabling it to tackle non-linear problems and high-dimensional data [55–57]. Both GBDT and RF are ensemble learning techniques rooted in decision trees. GBDT sequentially constructs multiple decision trees, with each subsequent tree aiming to correct the prediction errors of its predecessor, optimizing predictive performance by gradually minimizing residuals [58]. Conversely, RF concurrently builds numerous decision trees, utilizing averaging for regression and voting for classification, thereby mitigating the risk of overfitting and often yielding more stable and accurate predictions [59].

Due to the limited availability of only four well logs, we opted to utilize three of them for training the ML algorithms, reserving the remaining well for validating the predictions of the trained models. However, in scenarios where there were suspicions of inaccuracies in these logs, we relied on rock physics principles to calibrate and reconstruct the logs as replacements. As detailed in the Data section, we were able to generate density logs employing the Gardner relations, while V_c logs were derived utilizing the Larionov relations.

Standard scaling was applied to the input features for MLP and LR, whereas it was omitted for the two tree-based algorithms. The coefficient of determination (R^2) and root mean squared error (RMSE) were used as metrics to evaluate the accuracy of predictions made by ML algorithms. Furthermore, to interpret the mathematical principles of how these ML algorithms operate, both locally and globally, we used the SHapley Additive exPlanations (SHAP) framework [41]. Based on the solid theoretical foundation of cooperative game theory and the utilization of Shapley values, SHAP offers quantitative insights into the specific contributions of each feature to individual predictions, as well as their collective influence across the entire dataset [60,61]. This approach not only facilitates the assessment of ML algorithm performance but also serves as a valuable guide for feature selection and model design optimization.

The Python scripts used in this study for ML and SHAP analysis were based on the Scikit-learn and SHAP libraries [62,63].

Our primary objective is to elucidate the mathematical frameworks that underpin ML algorithms and interpret the causality behind their predictions within the context of rock physics. As such, we prefer to use standard and straightforward ML methods to facilitate their explanations and to derive conclusions that are broadly applicable. In interpreting SHAP, it is crucial to concentrate solely on the influence of input features by maintaining the consistency of the ML models. Our benchmarks, detailed in Appendix A, reveal that optimizing the models' hyperparameters did not significantly improve their performance compared to using fixed settings (Tables A1 and A2), but it did lead to notable changes in SHAP explanations (Figure A1). Therefore, we consistently avoid making any modifications based on optimization outcomes.

To address variability from random initialization, we tested 40 random seeds and found consistent results within each stochastic model, except for the MLP which showed significant fluctuations. This is shown in Figure A2.

3. Results and Discussions

3.1. Porosity

First, we show the results of ML-based prediction of porosity logs.

Figure 3 shows the density logs of the four wells, before and after correction using the Gardner relations. The original density of F02-1 is significantly lower than the other three wells. Since density logs were calculated from measured acoustic logs, we used the Gardner relations to fit the density of the three wells, where in the exponential function we

optimized the coefficient to be 0.1018 and the index to be 0.3445. Relying on the Gardner relations, density logs of F02-1 were corrected using acoustic logs as inputs.

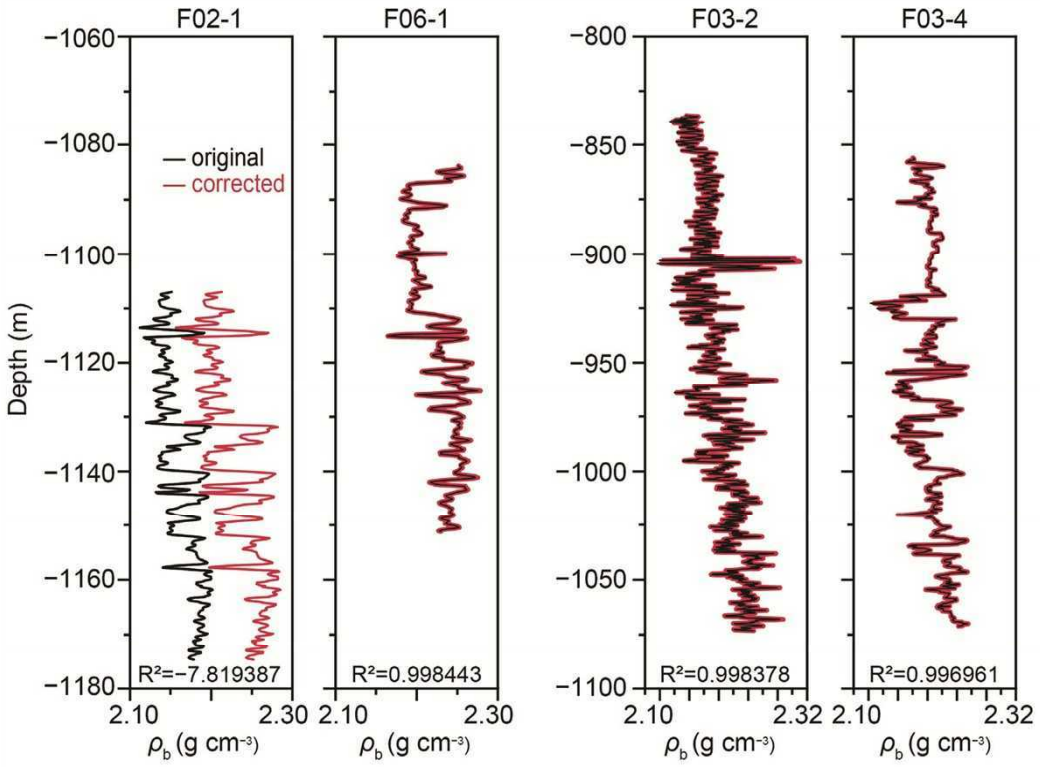


Figure 3. Density logs of the four wells before and after correction using the Gardner relations.

To gain a deeper understanding, we compiled a crossplot that illustrates the interrelationships among the seven log data sets, as shown in Figure 4. Notably, we have incorporated an additional well, F02-1Cor, which is a corrected version of F02-1. In this updated version, the density was fine-tuned using the Gardner relations, leading to a subsequent recalculation of the porosity. As observed in the crossplot comparing ρ_b and AC, the data points of F02-1 stand out conspicuously, indicating the need for further scrutiny. The corrected F02-1Cor, however, exhibits a seamless alignment with the other wells, demonstrating the effectiveness of the corrections. Moreover, we recalculated V_p as the reciprocal of AC, and consequently, AI was recalculated as the product of ρ_b and V_p. This corrected well, and F02-1Cor, exhibits an exceptional correlation across all seven logs, testament to the rigor and reliability of our findings. However, as Figure 4 indicates, the correlations are poor between gamma logs (and consequently V_c) and elastic logs, including AC, V_p, ρ_b, AI, and ϕ. This incongruence in characteristics could potentially influence ML-based predictions, and its implications will be discussed in the next section.

Figure 5 presents a comparative analysis of the ML-based prediction of porosity logs, highlighting the difference between the pre- and post-correction states of density logs. In this process, four parallel ML algorithms were used, specifically RF, GBDT, MLP, and LR. The training of these ML algorithms encompassed data from three wells, whereas the remaining well served as the testing ground. Notably, the problematic density logs from F02-1 posed a challenge.

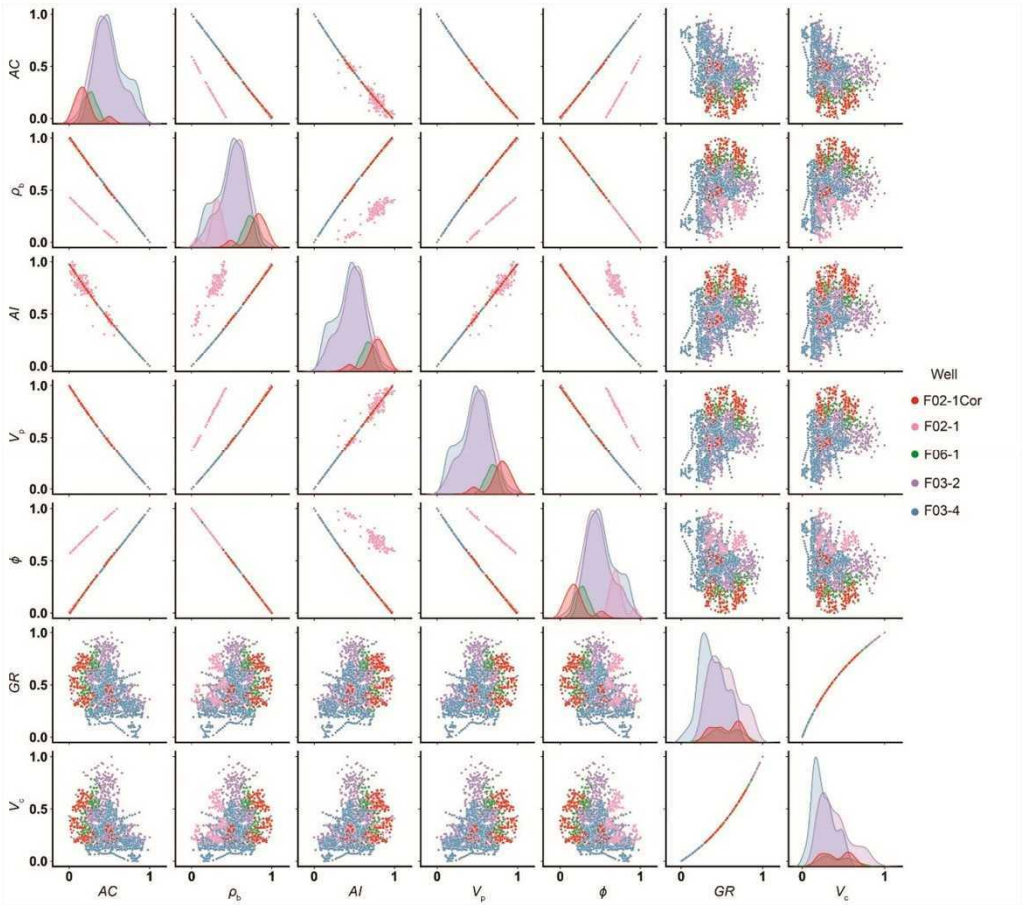


Figure 4. Crossplot of the seven logs of the four wells and F02-1Cor. From **left** to **right**, *AC*, ρ_b, *AI*, V_p, ϕ, *GR* and V_c.

In the pre-correction scenarios, two distinct approaches were adopted for ML training and testing: either including F02-1 in the training set or reserving it for testing. Given that porosity measurements were not readily available for all wells and were instead estimated from density logs using a linear formula, density itself was excluded as an input variable for ML training. Similarly, V_p and *AI* were also excluded, leaving only the measured logs, comprising *AC* and *GR*, as the input features.

Remarkably, for all four ML algorithms, the prediction accuracy of pre-correction states was deemed unsatisfactory. However, an interesting trend emerged when the problematic F02-1 was included in the training dataset, resulting in improved prediction performance compared to when it served as the testing well. Furthermore, the prediction accuracy of all four algorithms underwent a significant enhancement when both acoustic and gamma logs were used as inputs, surpassing the accuracy achieved by relying solely on acoustic logs. Despite the fact that gamma logs display a notably feeble correlation with porosity, as depicted in Figure 5, their integration with acoustic logs contributes to a more precise prediction.

After the correction of density logs, however, the prediction accuracy for all four ML algorithms experienced a substantial surge, achieving exceptionally high levels of accuracy, with R^2 values approaching 1 for each algorithm. This trend was observed when F03-4 served as the testing well, and the corrected F02-1 was incorporated into the training dataset, as shown in Figure 5c. It is crucial to emphasize that the achieved optimal performance hinges critically on the establishment of the ideal dataset setting, where the porosity data is systematically derived from the measured acoustic data through the application of a distinct and carefully calibrated formula. This underscores the paramount importance of a

rigorous and systematic approach to data processing and model training, which is essential for attaining predictions of both high fidelity and precision.

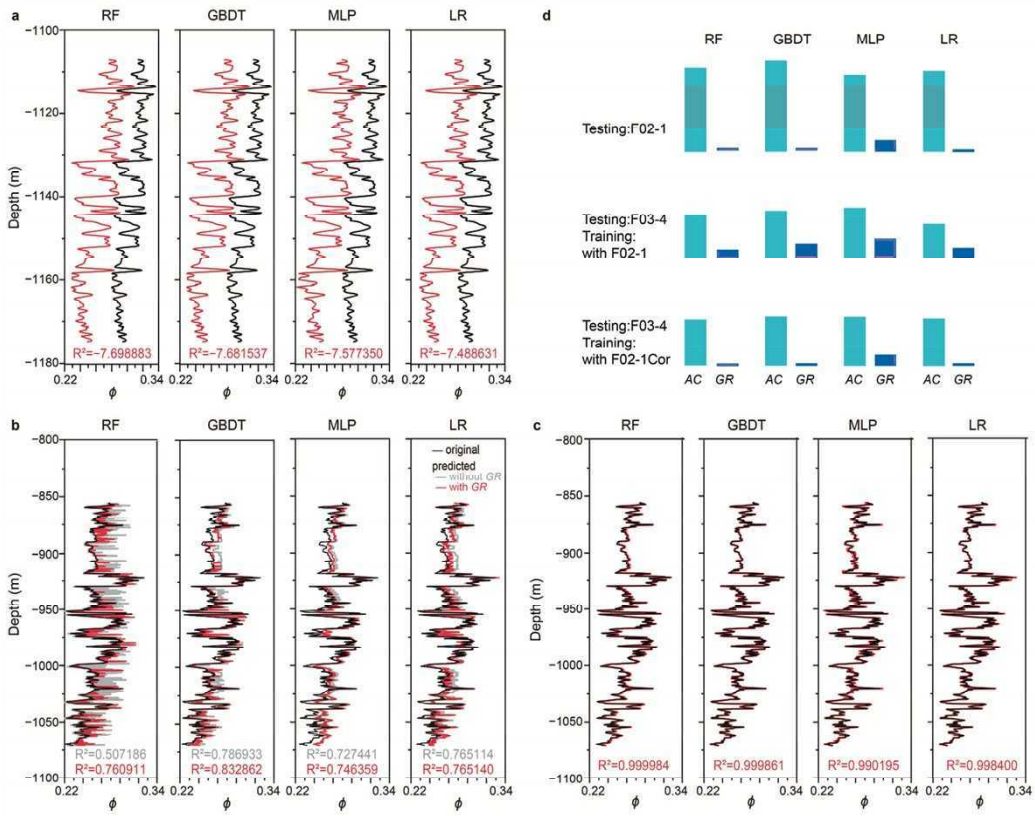


Figure 5. Prediction of porosity logs with original and corrected density logs using four ML algorithms. (**a**) F02-1 as the testing well. (**b**) F03-4 as the testing well. F02-1 was included in the training dataset. (**c**) F03-4 as the testing well. F02-1Cor was included in the training dataset. (**d**) SHAP explanation for these predictions. Top, for cases in both (**a**) and the cases where F02-1Cor served as the testing well. Middle, for cases in (**b**). Bottom, for cases in (**c**).

To gain a deeper insight into the mechanisms underlying ML-based predictions, global SHAP values were calculated to interpret the results. These analyses consistently pointed to acoustic logs as having an overarching influence on predicting porosity, regardless of the ML algorithm, training and testing scenarios, or prediction accuracy, as summarized in Figure 5d. Conversely, the other input logs, namely gamma logs, were largely disregarded, with the exception of one instance when the problematic F02-1 log was incorporated as an input. In this particular scenario, the inclusion of gamma logs resulted in a marginal improvement in prediction accuracy. This interpretation aligns with the weak correlation exhibited between gamma logs and porosity logs, as illustrated in Figure 4. Generally, MLP exhibits a greater reliance on gamma logs compared to the other three algorithms.

3.2. Clay Volume Fraction

Here we present the outcomes of ML-based prediction of clay volume fraction (V_c) logs and their interpretations.

Typically, V_c is derived from gamma logs, which records the natural gamma radiation emitted from radioactive elements present in clays. In this study, the Larionov relations were used to calculate V_c from the measured gamma logs. As depicted in Figure 4, V_c theoretically exhibits a strong correlation with gamma logs, but it correlates weakly with acoustic logs and other logs derived from acoustic logs, such as density, acoustic impedance, P-wave velocity, and porosity. Figure 6c shows the prediction of V_c for F03-4

when the inputs for training the ML algorithms included logs that correlate poorly with V_c. Specifically, the ML algorithm used was RF. The training data originated from the other three wells, spanning the interval between FS2 and FS4. As anticipated, the prediction performance was unsatisfactory. However, when gamma logs were incorporated into the training process, the prediction accuracy was improved significantly, as shown in Figure 6c. Similar results were observed when F03-2 served as the testing well and the other three wells as the training well, as shown in Figure 6e. Nonetheless, there were sections of the V_c logs for both F03-4 and F03-2 that remained inaccurately predicted.

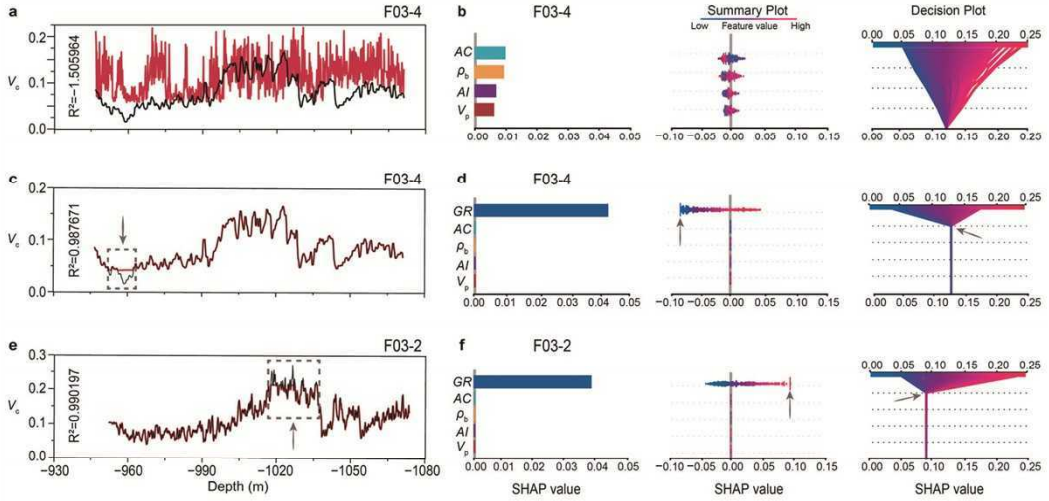


Figure 6. Prediction of clay volume fraction using RF and the SHAP explanations. (**a**) Prediction for F03-4 where gamma logs were excluded as inputs. (**b**) Feature importance for (**a**) as explained by SHAP summary plot and decision plot. (**c**) Prediction for F03-4 where gamma logs were incorporated as inputs. (**d**) SHAP explanations for (**c**). (**e**) Prediction for F03-2 where gamma logs were incorporated as inputs. (**f**) SHAP explanations for (**e**). Arrows indicate unpredictable samples.

To gain a more comprehensive understanding of the prediction's performance using RF and the other three ML algorithms, we calculated both global and local SHAP values for each individual prediction. In the case of predicting V_c for F03-4 using RF, gamma logs emerged as the predominant input feature, as explained by both the SHAP summary plot and decision plot (comparing Figure 6b with Figure 6d). This finding aligns with the anticipation based on human knowledge, as gamma logs are the physical trigger for the effect, namely V_c. Meanwhile, the impact of the other four input features is minimal. However, in scenarios where gamma logs are absent from the training dataset, the accuracy of V_c predictions declines significantly, rendering the importance of the four remaining input features indispensable. In such a situation, irrespective of the prediction inaccuracy, and regardless of the weak correlation between the input features and the prediction target, the ML algorithm relies heavily on the provided inputs, as it lacks alternative data sources. When assessing the impact on the inaccurate prediction results, the four input features are approximately of equal significance. This trend is observed across the other three algorithms, as shown in Figure 7b. However, if a ranking is necessary, acoustic logs emerge as the most crucial input except for LR, likely attributed to their status as the only measured feature among the inputs.

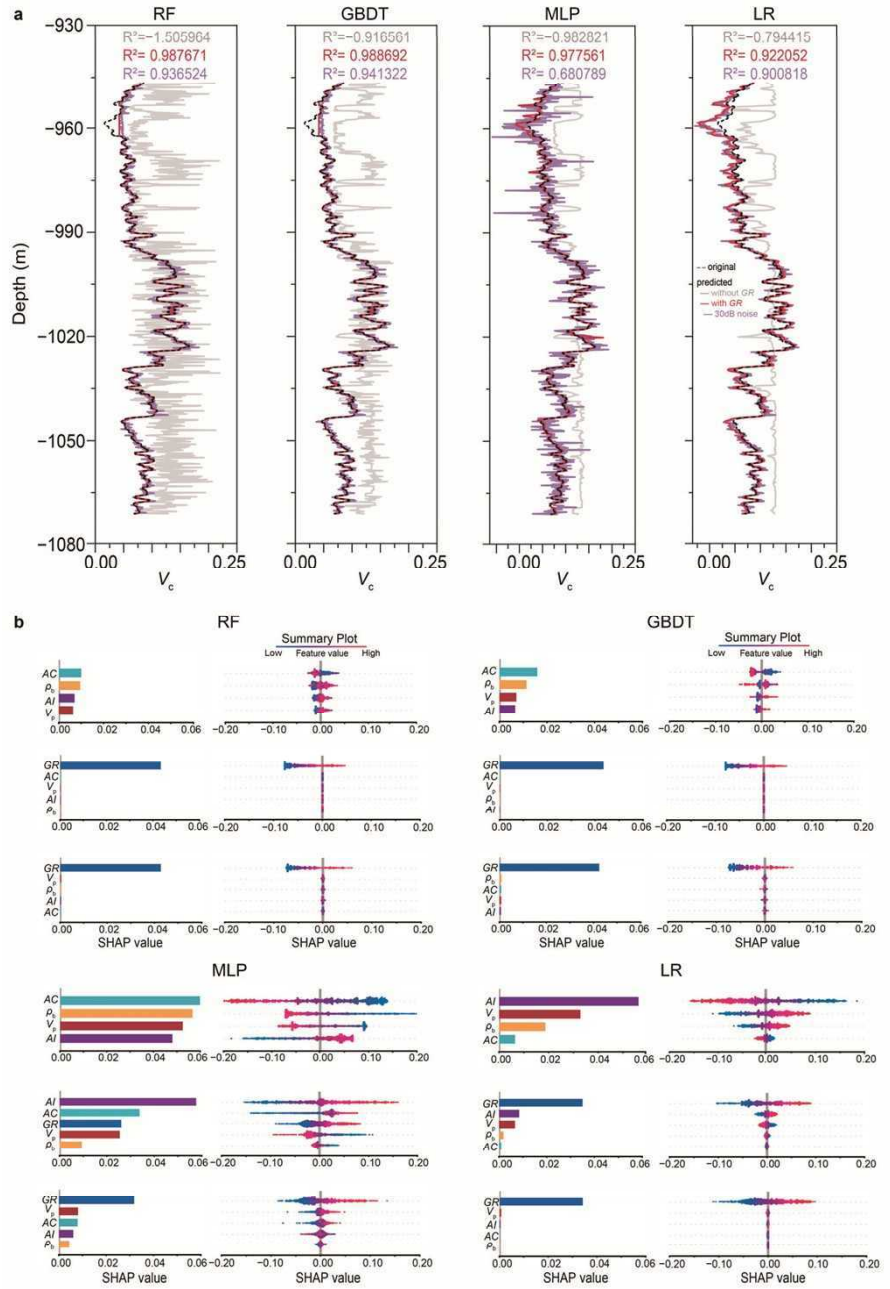


Figure 7. Accuracy and feature importance in predicting clay volume fraction for F03-4 as explained by SHAP summary plot for four ML algorithms. (**a**) Predicted V_c logs under three different scenarios. In contrast to a standard scenario, two variations were tested: one excluded GR in training, and the other employed a noise-robust training and testing approach with an SNR of 30. (**b**) Both local and global SHAP explanations for these three different prediction scenarios. For each algorithm: Top, training without GR. Middle, standard training and testing with GR. Bottom, noise-robust training and testing.

The interpretation of the significance of various features across all four ML algorithms is summarized in Figure 7. Notably, although MLP initially failed to prioritize gamma logs as the primary input feature, it regained its rationality after undergoing a noise-robust training procedure. This can be observed in Figure 7b for MLP, which demonstrates the algorithm's adaptability. Within this noise-robust training paradigm, the predictions attained satisfactory levels, albeit with a notable decline in MLP's performance due to a relatively low signal-to-noise ratio (SNR) of 30, as evident in Figure 7a for MLP. The SHAP analysis revealed consistent patterns among all four algorithms, regardless of their

varied underlying architectures—whether tree-based, neutron network-driven, or linear regression-oriented. By comparing the SHAP explanations with rock physics principles, we observed a close alignment of the four algorithms with the fundamentals of rock physics, indicating adherence to its cause relationships.

To interpret the sections of the V_c logs that exhibited unpredictable patterns at a depth of approximately −960 m, a review of the training and the testing dataset was undertaken. Specifically, considering the scenario where F03-4 serves as the testing well and the other three wells as the training well, we note that the V_c range for the other three wells ranges from 0.050 to 0.300, whereas the V_c range for F03-4 to be predicted is 0.025 to 0.200. Therefore, the predicted lowest values of V_c for F03-4 exceed the bounds of the training dataset, which is indicative of an Independent and Identically Distributed (IID) issue. Analogously, reversing the roles of the testing and training well also leads to an inaccurate prediction of the higher V_c values in F03-2. The lowest and highest unpredictable V_c values are positioned at the periphery of both the SHAP summary plot and the decision plot, as indicated by the grey arrows in Figure 6c,d, and the yellow arrows in Figure 6e,f.

To further elaborate on the intricacies of the IID problem and explain the inability to make accurate predictions, we employed a straightforward binary logarithm function of $y = \log_2 x$, with the range of x set from 1 to 11. We subjected the prediction of y to a rigorous test using four different ML algorithms, contrasting the results against those obtained from cubic spline interpolation and the ground truth. As shown in Figure 8a at $x = 2.4$, where the true value of y is denoted by black crosses along the black line, RF predicted $y = 1.291$, marked by a black dot, exceeding the actual value. From $x = 2.0$ to 2.8, the predictions either exceeded or fell below the true values, as evident in the lower inset. Notably, as we incrementally removed one or two neighboring labels from the training dataset, the predicted values rose significantly, depicted by blue dots. Conversely, the predicted values decreased progressively when one or two labels to the right were eliminated. A comparison with the ground truth is illustrated in the upper inset. Similar trends were observed for the x range from 9.0 to 9.8, as shown in Figure 8b. The performance of the remaining three algorithms is presented in Figure 8c–e. Despite the remarkable success ML has achieved owing to its numerous advantages, the predictions generated by the four ML algorithms can still trail behind the accuracy of cubic spline interpolation, even with a substantially vast training dataset.

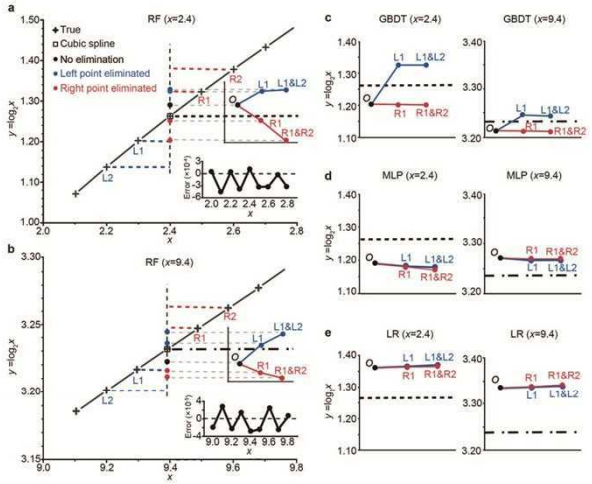


Figure 8. ML-based prediction of the binary logarithm function. (**a**) For x range from 2.0 to 2.8. Black line, theoretical line. Black crosses, theoretical values. Dots, ML-based predictions. Upper inset, predictions versus truth as neighboring samples eliminated from training. Lower inset, prediction errors at each testing point. (**b**) For x range from 9.0 to 9.8. (**c**) For GBDT. **Left**, $x = 2.4$. **Right**, $x = 9.4$. (**d**) For MLP. (**e**) For LR.

These findings underscore the sensitivity of ML-based predictions to the training dataset. Furthermore, as neighboring samples are gradually removed, the IID assumption becomes invalid, leading to an out-of-distribution (OOD) problem that ultimately compromises the predictive accuracy.

4. Conclusions

We have integrated SHAP and rock physics into the ML-based approach for predicting well logs, specifically targeting porosity and clay volume fraction. While ML excels in handling complex, nonlinear challenges that involve diverse input features and extensive datasets, its operational mechanisms are often less mathematically transparent compared to traditional algorithms such as least squares and Bayesian methods. Conversely, rock physics, with its clear and intuitive foundations, faces constraints in its applicability due to the numerous inherent assumptions, hampering its capacity to address ambiguous or uncertain scenarios.

The union of these two methodologies has yielded significant advantages in our study. Rock physics has played a pivotal role in feature engineering, selection, algorithm comparison, and the comprehensive evaluation of four distinct algorithms: RF, GBDT, MLP, and LR. Despite their varied architectures—tree-based, neutron network-driven, or linear regression-oriented—the SHAP analysis has uncovered consistent operational patterns across these algorithms in noise-robust training and testing scenarios. Moreover, the SHAP explanations closely align with the causal relationships outlined in rock physics principles. However, when crucial controlling input features were intentionally omitted, affecting prediction accuracy, all four ML algorithms and the SHAP analysis continued to operate in a seemingly irrational manner, starkly contrasting with the fundamental principles of rock physics. By comparing SHAP explanations with rock physics concepts, we have established parallels and contrasts with human knowledge, inherently rooted in causal relationships. The invaluable guidance provided by rock physics has empowered us to intervene in ML-based predictions, not only enhancing our comprehension of the strengths and limitations of ML models but also potentially paving the way for optimizing the performance of these workflows. This approach has propelled a transition from purely mathematical explanations to a more philosophical interpretation of ML-based predictions, effectively dismantling the traditional black-box nature of ML models.

Author Contributions: Methodology, J.Z. and Y.C.; Data analysis, J.Z. and Z.W.; Results interpretation, J.Z., G.L., S.L., Y.Z. and Y.C.; Writing—original draft, J.Z., G.L. and Y.C.; Funding acquisition, G.L. and Y.C.; Supervision, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Xinjiang Uygur Autonomous Region (2022D01C422); the Tianchi Scholarship of Xinjiang Uygur Autonomous Region (Chengyuanfeng2022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The custom Python scripts and data associated with this research are available and can be accessed via the following URL: https://gitee.com/zhangjixju/ml-code/tree/master (accessed on 13 January 2025).

Acknowledgments: The authors are very grateful to Xingmou Wang of SINOPEC for valuable scientific discussions on field applications of MLs. We thank dGB Earth Sciences for making the data available as an OpendTect project via their TerraNubis portal terranubis.com.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In this appendix, we provide a comprehensive account of the execution of the ML and SHAP methodologies.

To assess the effectiveness of hyperparameter optimization, two comprehensive benchmark tests were conducted for the three stochastic models, RF, GBDT and MLP. One test, termed case-ϕ (corresponding to Figure 5c), evaluated the porosity prediction with exceptional accuracy across all four algorithms. The other test, termed case-V_c (corresponding to Figure 6a), evaluated the V_c prediction and found significantly lower accuracy for all four algorithms.

Table A1 summarizes all hyperparameters and their respective final values for each model. Table A2 provides a detailed description of the hyperparameter tuning process, which utilized random search and k-fold cross-validation.

The benchmark tests indicate that optimizing the models' hyperparameters does not yield a substantial improvement in their performance metrics, specifically R^2 and RMSE, compared to using fixed settings. Furthermore, as illustrated in Figure A1, while optimization has a negligible effect, it significantly modifies the SHAP explanations for case-V_c. Both the magnitude and the relative importance of input features exhibit notable variations.

To assess the effect of random initialization on the performance of the three stochastic models with fixed hyperparameter settings, we conducted experiments using 40 different random seeds. Our findings indicated that the results were generally consistent for each model for case-ϕ. Random initialization exhibits minimal influence on the performance of all three models. Nevertheless, for case-V_c, there are substantial fluctuations in prediction accuracy, especially for MLP, as illustrated in Figure A2.

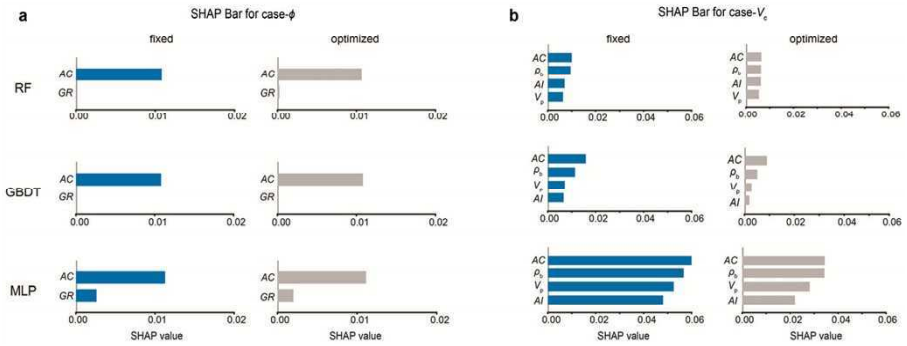


Figure A1. Global SHAP explanations for the two benchmark tests. (**a**) case-ϕ and (**b**) case-V_c. Model hyperparameters are fixed (blue) and optimized (grey).

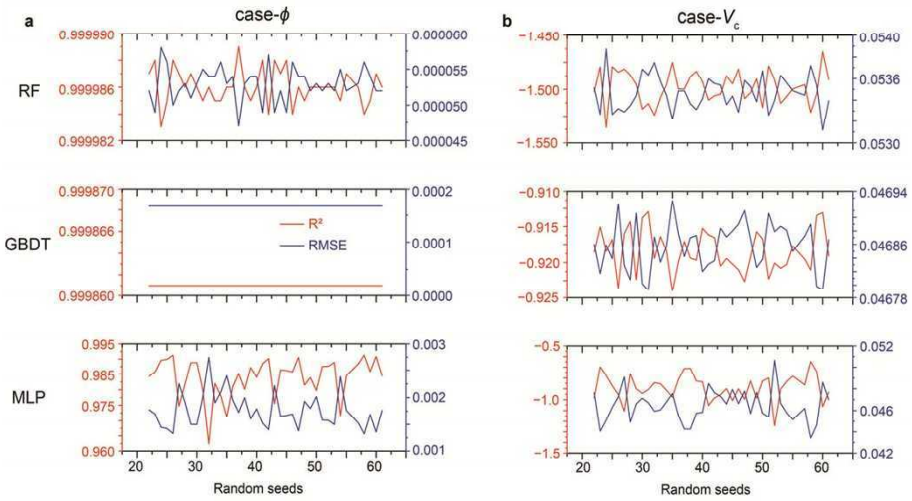


Figure A2. Performance of the three stochastic models' initialization randomly with 40 different seeds for the fixed hyperparameter settings. (**a**) case-ϕ. (**b**) case-V_c.

Table A1. Hyperparameters for the three stochastic ML models.

Model			Hyperparameters														Performance	
RF			n_estimators		max_depth		min_samples_splits		bootstrap		min_samples_leaf		random_state				R^2	RMSE
	fixed	case-ϕ	200		none		2		true		1		42				0.999984	0.000057
		case-V_c															−1.505964	0.053555
	optimization range		100–500		10–50, none		2–10		true, false		1–4		42					
	opt.	case-ϕ	200		20		2		false		1		42				0.999977	0.000068
		case-V_c	300		10		9		true		1		42				−1.110164	0.049144
GBDT			n_estimators		max_depth		learning_rate		subsample		min_samples_leaf		random_state				R^2	RMSE
	fixed	case-ϕ	200		3		0.1		1.0		1		42				0.999861	0.000167
		case-V_c															−0.916561	0.046836
	optimization range		100–500		3–7		0.01–0.3		0.6–1		1–6		42					
	opt.	case-ϕ	300		6		0.03		0.6		1		42				0.999992	0.000039
		case-V_c	200		3		0.01		0.7		3		42				−0.823654	0.045686
MLP			hidden layers	number of neurons	activation	batch size	learn. rate	solver	beta1	beta2	epsilon	tol	max iter	n_iter no change	early stop.	alpha	random state	R^2	RMSE
	fixed	case-ϕ	2	L1 100 L2 50	relu	auto	0.001	adam	0.9	0.999	10^{-8}	0.0001	200	50	true	0.0001	42	0.990195	0.001403
		case-V_c																−0.982821	0.047638
	optimization range		1–3		logistic, relu, tanh	16–128	0.001–0.1	adam, sgd	0.9	0.999	10^{-8}	0.0001–0.01	200–2000	50	true	0.0001–1	42		
	opt.	case-ϕ	2	L1 100 L2 100	logistic	40	0.003	adam	0.9	0.999	10^{-8}	0.01	200	50	true	0.0001	42	0.998986	0.000451
		case-V_c	1	50	relu	126	0.05	sgd	0.9	0.999	10^{-8}	0.01	1200	50	true	0.9474	42	−0.883199	0.046426

Table A2. Optimization of model hyperparameters via random search (200 iterations) with k-fold cross-validation, using RMSE as the performance metric.

Models	K-Fold (n_Folds = 5)			Testing: F03-4	
	Training: F02-1, F06-1, F03-2				
		Mean_R^2	Mean_RMSE	R^2	RMSE
RF	case-ϕ	0.999910	0.000186	0.999977	0.000068
	case-V_c	0.020244	0.046256	−1.110164	0.049144
GBDT	case-ϕ	0.999965	0.000114	0.999992	0.000039
	case-V_c	0.110894	0.044113	−0.823654	0.045686
MLP	case-ϕ	0.999084	0.000615	0.998986	0.000451
	case-V_c	0.122800	0.043823	−0.883199	0.046426

References

1. Ellis, D.V.; Singer, J.M. *Well Logging for Earth Scientists*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2008.
2. Selley, R.C.; Sonnenberg, S.A. *Elements of Petroleum Geology*, 3rd ed.; Academic Press: San Diego, CA, USA, 2015.
3. Tiab, D.; Donaldson, E.C. *Petrophysics: Theory and Practice of Measuring Reservoir Rock and Fluid Transport Properties*, 4th ed.; Gulf Professional Publishing: Boston, MA, USA, 2015.
4. Wang, H.; Ma, J.; Li, L. Pore pressure prediction based on rock physics theory and its application in seismic inversion. *J. Appl. Geophys.* **2024**, *229*, 105494. [CrossRef]
5. Aki, K.; Richards, P.G. *Quantitative Seismology*, 1st ed.; W.H. Freeman and Co.: San Francisco, CA, USA, 1981.
6. Avseth, P.; Mukerji, T.; Mavko, G. *Quantitative Seismic Interpretation*, 1st ed.; Cambridge University Press: Cambridge, UK, 2005.
7. Ettayebi, M.; Wang, S.; Landrø, M. Time-Lapse 3D CSEM for Reservoir Monitoring Based on Rock Physics Simulation of the Wisting Oil Field Offshore Norway. *Sensors* **2023**, *23*, 7197. [CrossRef] [PubMed]
8. Mavko, G.; Mukerji, T.; Dvorkin, J. *The Handbook of Rock Physics*, 1st ed.; Cambridge University Press: Cambridge, UK, 1998.
9. Hertz, H. Ueber die Berührung fester elastischer Körper. *J. Reine Angew. Math.* **1882**, *92*, 156–171. [CrossRef]
10. Zoeppritz, K. Erdbebenwellen VIII B. Uber reflexion und durchgang seismischer wellen durch unstetigkeisflachen. *Gottinger Nachr.* **1919**, *1*, 66–84.
11. Gassmann, F. Uber die elastizitat poroser medien. *Vierteljahrsschr. Naturforschenden Ges. Zur.* **1951**, *96*, 1–23.
12. Kuster, G.T.; Toksöz, M.N. Velocity and attenuation of seismic waves in two-phase media: Part I. Theoretical Formulations. *Geophysics* **1974**, *39*, 587–606. [CrossRef]
13. Wyllie, M.R.J.; Gregory, A.R.; Gardner, L.W. Elastic wave velocities in heterogeneous and porous media. *Geophysics* **1956**, *21*, 41–70. [CrossRef]
14. Larionov, V.V. *Borehole Radiometry*; Nedra: Moscow, Russia, 1969.
15. Poupon, A.; Gaymard, R. The evaluation of clay content from logs. In Proceedings of the SPWLA Annual Logging Symposium, Los Angeles, CA, USA, 3–6 May 1970; SPWLA–1970–G.
16. Gardner, G.H.F.; Gardner, L.W.; Gregory, A.R. Formation velocity and density-the diagnostic basics for stratigraphic traps. *Geophysics* **1974**, *39*, 770–780. [CrossRef]
17. Han, D. Effects of Porosity and Clay Content on Acoustic Properties of Sandstones and Unconsolidated Sediments. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 1986.
18. Fjaer, E.; Holt, R.M.; Horsrud, P.; Raaen, A.M.; Risnes, R. *Petroleum Related Rock Mechanics*; Elsevier: Amsterdam, The Netherlands, 2008.
19. Simm, R.; Bacon, M. *Seismic Amplitude: An Interpreter's Handbook*, 1st ed.; Cambridge University Press: Cambridge, UK, 2014.
20. Pham, N.; Wu, X.; Naeini, E.Z. Missing well log prediction using convolutional long short-term memory network. *Geophysics* **2020**, *85*, WA159–WA171. [CrossRef]
21. Feng, R.; Grana, D.; Balling, N. Variational inference in Bayesian neuralnetwork for well-log prediction. *Geophysics* **2021**, *86*, M91–M99. [CrossRef]
22. Chen, B.; Zeng, X.; Fan, L.; Li, K.; Zhang, W.; Cai, G.; Wang, Y.; Du, R.; Chen, T.; Zhang, B.; et al. Adaptive spatiotemporal neural networks based on machine learning for missing well-log prediction. *Geophysics* **2023**, *88*, V431–V443. [CrossRef]
23. Lin, L.; Wei, H.; Wu, T.; Zhang, P.; Zhong, Z.; Li, C. Missing well-log reconstruction using a sequence self-attention deep-learning framework. *Geophysics* **2023**, *88*, D391–D410. [CrossRef]
24. Pan, W.; Torres-Verdín, C.; Duncan, I.J.; Pyrcz, M.J. Improving multiwell petrophysical interpretation from well logs via machine learning and statistical models. *Geophysics* **2023**, *88*, D159–D175. [CrossRef]

25. Anjom, F.K.; Vaccarino, F.; Socco, L.V. Machine learning for seismic exploration: Where are we and how far are we from the holy grail? *Geophysics* **2024**, *89*, WA157–WA178. [CrossRef]
26. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
27. O'Callaghan, J. How OpenAI's text-to-video tool Sora could change science-and society. *Nature* **2024**, *627*, 475–476. [CrossRef] [PubMed]
28. Quiñonero-Candela, J.; Sugiyama, M.; Lawrence, N.D.; Schwaighofer, A. *Dataset Shift in Machine Learning*, 1st ed.; The MIT Press: Cambridge, MA, USA, 2009.
29. Pearl, J.; Mackenzie, D. *The Book of Why: The New Science of Cause and Effect*, 1st ed.; Basic Books: New York, NY, USA, 2018.
30. Zhang, Y.; Tino, P.; Leonardis, A.; Tang, K. A survey on neural network interpretability. *IEEE Trans. Emerg. Top. Comput. Intell.* **2021**, *5*, 726–742. [CrossRef]
31. Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; Yu, P.S. Generalizing to unseen domains: A survey on domain generalization. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 8052–8072.
32. Schölkopf, B. Causality for machine learning. *arXiv* **2019**, arXiv:1911.10500v2.
33. Schölkopf, B.; Locatello, F.; Bauer, S.; Ke, N.R.; Kalchbrenner, N.; Goyal, A.; Bengio, Y. Toward causal representation learning. *Proc. IEEE* **2021**, *109*, 612–634. [CrossRef]
34. Khan, I.U.; Aslam, N.; AlShedayed, R. A proactive attack detection for heating, ventilation, and air conditioning (HVAC) system using explainable extreme gradient boosting model (XGBoost). *Sensors* **2022**, *22*, 9235. [CrossRef] [PubMed]
35. Mustafa, A.; Koster, K.; AlRegib, G. Explainable machine learning for hydrocarbon prospect risking. *Geophysics* **2024**, *89*, WA13–WA24. [CrossRef]
36. Karniadakis, G.E.; Kevrekisdis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [CrossRef]
37. Li, P.; Liu, M.; Alfarraj, M.; Tahmasebi, P.; Grana, D. Probabilistic physics-informed neural network for seismic petrophysical inversion. *Geophysics* **2024**, *89*, M17–M32. [CrossRef]
38. Zhao, T.; Liu, C.; Song, C.; Waheed, U.B.; Zhang, X. Smoothness: The key factor in well-log information-assisted PINNtomo. *J. Appl. Geophys.* **2024**, *226*, 105417. [CrossRef]
39. Liu, T.; Ungar, L.; Kording, K. Quantifying causality in data science with quasi-experiments. *Nat. Comput. Sci.* **2021**, *1*, 24–32. [CrossRef]
40. von Rueden, L.; Mayer, S.; Beckh, K.; Georgiev, B.; Giesselbach, S.; Heese, R. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 614–633. [CrossRef]
41. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4765–4774.
42. Lee, J.; Lumley, D.E. Interpreting the effects of shale rock properties on seismic anisotropy by statistical and machine learning methods. *Geoenergy Sci. Eng.* **2023**, *224*, 211631. [CrossRef]
43. Nazat, S.; Arreche, O.; Abdallah, M. On Evaluating Black-Box Explainable AI Methods for Enhancing Anomaly Detection in Autonomous Driving Systems. *Sensors* **2024**, *24*, 3515. [CrossRef]
44. Oloruntobi, O.; Butt, S. The new formation bulk density predictions for siliciclastic rocks. *J. Pet. Sci. Eng.* **2019**, *180*, 529–537. [CrossRef]
45. Tudge, J.; Tobin, H.J. Velocity-porosity relationships in smectite-rich sediments: Shikoku Basin, Japan. *Geochem. Geophys. Geosystems* **2013**, *14*, 5194–5207. [CrossRef]
46. Wang, H.; Liu, T.; Tang, T.; Shi, Y. A unified model to evaluate shaliness in compacted and soft formations using downhole GR log. *J. Pet. Sci. Eng.* **2017**, *156*, 877–883. [CrossRef]
47. Aftab, F.; Zafar, M.; Hajana, M.I.; Ahmad, W. A novel gas sands characterization and improved depositional modeling of the Cretaceous Sembar Formation, Lower Indus Basin, Pakistan. *Front. Earth Sci.* **2022**, *10*, 1039605. [CrossRef]
48. Zhu, L.Q.; Sun, J.; Zhou, X.Q.; Li, Q.P.; Fan, Q.; Wu, S.L.; Wu, S.G. Well logging evaluation of fine-grained hydrate-bearing sediment reservoirs: Considering the effect of clay content. *Pet. Sci.* **2023**, *20*, 879–892. [CrossRef]
49. Steeghs, P.; Overeem, I.; Tigrek, S. Seismic volume attribute analysis of the Cenozoic succession in the L08 block (Southern North Sea). *Glob. Planet. Change* **2000**, *27*, 245–262. [CrossRef]
50. Sørensen, J.C.; Gregersen, U.; Breiner, M.; Michelsen, O. High-frequency sequence stratigraphy of Upper Cenozoic deposits in the central and southeastern North Sea areas. *Mar. Pet. Geol.* **1997**, *14*, 99–123. [CrossRef]
51. Mojeddifar, S.; Kamali, G.; Ranjbar, H. Porosity prediction from seismic inversion of a similarity attribute based on a pseudo-forward equation (PFE): A case study from the North Sea Basin, Netherlands. *Pet. Sci.* **2015**, *12*, 428–442. [CrossRef]
52. Ishak, M.A.; Islam, M.A.; Shalaby, M.R.; Hasan, N. The Application of Seismic Attributes and Wheeler Transformations for the Geomorphological Interpretation of Stratigraphic surfaces: A Case Study of the F3 Block, Dutch Offshore Sector, North Sea. *Geosciences* **2018**, *8*, 79. [CrossRef]
53. dGB. Available online: https://terranubis.com/datainfo/F3-Demo-2020 (accessed on 26 June 2024).
54. Gross, J. *Linear Regression*; Springer: Berlin/Heidelberg, Germany, 2003.

55. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **1958**, *65*, 386–408. [CrossRef]
56. Minsky, M.; Papert, S. *Perceptrons: AN Introduction to Computational Geometry*, 1st ed.; The MIT Press: Cambridge, MA, USA, 1969.
57. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
58. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
59. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
60. Shapley, L.S. A value for n-person games. In *Contributions to the Theory of Games*; Princeton University Press: Princeton, NJ, USA, 1953; pp. 307–317.
61. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.-I. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. [CrossRef] [PubMed]
62. Scikit-Learn. Available online: https://scikit-learn.org (accessed on 12 January 2025).
63. Lundberg, S.M. Available online: https://pypi.org/project/shap (accessed on 26 June 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

MDPI Books Editorial Office
E-mail: books@mdpi.com
www.mdpi.com/books



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Topic Editors. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editors and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.





Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-4168-4