

N. R. Shetty · L. M. Patnaik ·
H. C. Nagaraj · K. R. Venugopal ·
Sreekanth Rallapalli *Editors*

Advances in Artificial Intelligence and Machine Learning

Proceedings of ERCICAM 2024, Volume 2

Lecture Notes in Electrical Engineering

Volume 1335

Series Editors

Leopoldo Angrisani, Department of Electrical and Information Technologies Engineering, University of Napoli Federico II, Napoli, Italy

Marco Arteaga, Departament de Control y Robótica, Universidad Nacional Autónoma de México, Coyoacán, Mexico

Samarjit Chakraborty, Fakultät für Elektrotechnik und Informationstechnik, TU München, Munich, Germany

Shanben Chen, School of Materials Science and Engineering, Shanghai Jiao Tong University, Shanghai, China

Tan Kay Chen, Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore

Rüdiger Dillmann, University of Karlsruhe (TH) IAIM, Karlsruhe, Germany

Haibin Duan, Beijing University of Aeronautics and Astronautics, Beijing, China

Gianluigi Ferrari, Dipartimento di Ingegneria dell'Informazione, Sede Scientifica Università degli Studi di Parma, Parma, Italy

Manuel Ferre, Centre for Automation and Robotics CAR (UPM-CSIC), Universidad Politécnica de Madrid, Madrid, Spain

Faryar Jabbari, Department of Mechanical and Aerospace Engineering, University of California, Irvine, USA

Limin Jia, State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China

Janusz Kacprzyk, Intelligent Systems Laboratory, Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Alaa Khamis, Department of Mechatronics Engineering, German University in Egypt El Tagamoa El Khames, New Cairo City, Egypt

Torsten Kroeger, Intrinsic Innovation, Mountain View, USA

Yong Li, College of Electrical and Information Engineering, Hunan University, Changsha, China

Qilian Liang, Department of Electrical Engineering, University of Texas at Arlington, Arlington, USA

Ferran Martín, Departament d'Enginyeria Electrònica, Universitat Autònoma de Barcelona, Bellaterra, Spain

Tan Cher Ming, College of Engineering, Nanyang Technological University, Singapore, Singapore

Wolfgang Minker, Institute of Information Technology, University of Ulm, Ulm, Germany

Pradeep Misra, Department of Electrical Engineering, Wright State University, Dayton, USA

Subhas Mukhopadhyay, School of Engineering, Macquarie University, Sydney, NSW, Australia

Cun-Zheng Ning, Department of Electrical Engineering, Arizona State University, Tempe, AZ, USA

Toyooki Nishida, Department of Intelligence Science and Technology, Kyoto University, Kyoto, Japan

Luca Oneto, Department of Informatics, Bioengineering, Robotics and Systems Engineering, University of Genova, Genova, Italy

Bijaya Ketan Panigrahi, Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi, India

Federica Pascucci, Dipartimento di Ingegneria, Università degli Studi Roma Tre, Rome, Italy

Yong Qin, State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China

Gan Woon Seng, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

Joachim Speidel, Institute of Telecommunications, University of Stuttgart, Stuttgart, Germany

Germano Veiga, FEUP Campus, INESC Porto, Porto, Portugal

Haitao Wu, Academy of Opto-electronics, Chinese Academy of Sciences, Beijing, China

Walter Zamboni, Department of Computer Engineering, Electrical Engineering and Applied Mathematics, DIEM—Università degli studi di Salerno, Fisciano, Italy

Kay Chen Tan, Department of Computing, Hong Kong Polytechnic University, Hong Kong, Hong Kong

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering—quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

N. R. Shetty · L. M. Patnaik · H. C. Nagaraj ·
K. R. Venugopal · Sreekanth Rallapalli
Editors

Advances in Artificial Intelligence and Machine Learning

Proceedings of ERCICAM 2024, Volume 2

Editors

N. R. Shetty
Nitte Deemed to be University, Nitte
Meenakshi Institute of Technology
Bengaluru, Karnataka, India

H. C. Nagaraj
Nitte Deemed to be University, Nitte
Meenakshi Institute of Technology
Bengaluru, Karnataka, India

Sreekanth Rallapalli 
Nitte Deemed to be University, Nitte
Meenakshi Institute of Technology
Bengaluru, Karnataka, India

L. M. Patnaik
National Institute of Advanced Studies
Bengaluru, Karnataka, India

K. R. Venugopal
Bengaluru University
Bengaluru, Karnataka, India

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-96-3104-9 ISBN 978-981-96-3105-6 (eBook)
<https://doi.org/10.1007/978-981-96-3105-6>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

If disposing of this product, please recycle the paper.

Organizing Committee

ERCICAM 2024

The First International Conference on “**Emerging Research in Computing, Information, Communication, Artificial Intelligence and Machine Learning**” (ERCICAM 2024) was held during February 23–24, 2024, at Nitte Meenakshi Institute of Technology (NMIT), Bengaluru.

ERCICAM 2024—Committee

Chief Patrons

Dr. N. V. Hegde, Hon’ble Chancellor—Nitte University and President, Nitte Education Trust, Mangaluru, India

Dr. N. R. Shetty, Former Chancellor—Central University of Karnataka, Kalaburagi and Advisor, Nitte Education Trust, Mangaluru, India

Mr. Rohit Punja, Administrator, NET, Mangaluru, India

Dr. S. Vidyashankar, Hon’ble Vice Chancellor, Visvesvaraya Technological University, Belagavi, Karnataka, India

Patrons

Dr. Sandeep Shastri, Director (Academics), Nitte Education Trust, Mangaluru, Karnataka, India

Dr. V. Sridhar, Dean (Academics), NMIT, Bengaluru, India

Conference Chair

Dr. H. C. Nagaraj, Principal, NMIT, Bengaluru, India

Program Chairs

Dr. N. Nalini, Professor (CSE), NMIT, Bengaluru, Karnataka, India

Dr. Sreekanth Rallapalli, Professor and Head, Department of MCA, NMIT, Bengaluru, Karnataka, India

Publication

Springer-LNEE Series

Advisory Committee

Dr. L. M. Patnaik, INSA Senior Scientist, NIAS, Bengaluru, India

Dr. K. D. Nayak, Former OS and CC, R&D (MED and MIST), DRDO, India

Dr. Anil Sahasrabudhe, Chairman, EC National Assessment and Accreditation Council Educational (NAAC), India

Dr. Kalidas Shetty, Founding Director of Global Institute of Food Security and International Agriculture (GIFSIA), North Dakota State University, Fargo, USA

Dr. Varun M. Deshpande, Edison Digital Expert, Schneider Electric, Franklin, Tennessee, USA

Dr. K. R. Venugopal, Former Vice-Chancellor, Bengaluru University, Bengaluru, Karnataka, India

Mr. Neal Wagner, Complex Systems Scientist, Systems and Technology Research, Massachusetts, USA.

Dr. Vincenzo Piuri, Professor, University of Milan, Italy

Mr. Victor Méndez Munoz, Head of Software Development/Consultant Professor IUL, S.A/Universitat Oberta de Catalunya, Spain

Dr. Ho Chin Kuan, Vice Chancellor, Asia Pacific University of Technology and Innovation Technology, Malaysia

Dr. Sohail S. Chaudhry, Professor, Villanova University, USA

Mr. Arvind Kumar Bhardwaj, Senior Software Architect, Capgemini, Houston, Texas, USA

Dr. Wei-Chiang Hong, Professor, Oriental Institute of Technology, Taiwan

Mr. B. L. Narasimharaju, Associate Professor, National Institute of Technology, Warangal, India

Dr. Saurabh Kumar Pandey, Assistant Professor, Indian Institute of Technology, Patna, India

Dr. P. Deepa Shenoy, 2022 Chair, IEEE Bangalore Section, University Visvesvaraya College of Engineering, Bengaluru, Karnataka, India

Publication Committee

Dr. Piyush Kumar Pareek, Professor and Head, Department of AI and ML, NMIT, Bengaluru, Karnataka, India

Dr. B. D. Parameshachari, Professor, Department of ECE, NMIT, Bengaluru, Karnataka, India

Program Committee

Dr. Ayush Goyal, Assistant Professor, Texas A&M University—Kingsville, USA

Dr. Victor Mendez Munoz, Head of Software Development/Consultant Professor, IUL, S.A/Universitat Oberta de Catalunya, Spain

Dr. Jordan Pascual Espada, Deputy Director of the School of Computer Engineering, University of Oviedo, USA

Dr. Sohail S. Chaudhry, Professor, Villanova University, USA

Dr. Bui Thanh Khoa, Industrial University of Ho Chi Minh City, Vietnam

Dr. Michael E. Auer, Prof. Dr. (mult.), Carinthia University of Applied Sciences, Austria

Dr. Man Qi, Senior Lecturer Canterbury Christ Church University, UK

Dr. Y-h. Taguchi, Professor, Chuo University, Japan

John Acken, Professor, Department of Electrical and computer Engineering, Portland State University, Portland Oregon, USA

Dr. S. Vijaya Shetty, Professor and Head, Department of CSE, NMIT, Bengaluru, Karnataka, India

Dr. S. G. Mohan, Professor and Head, Department of ISE, NMIT, Bengaluru, Karnataka, India

Dr. A. C. Ramachandra, Professor and Head, Department of ECE, NMIT, Bengaluru, Karnataka, India

Dr. Premila Manohar, Professor and Head, Department of EEE, NMIT, Bengaluru, Karnataka, India

Dr. Meenaskhi Sundaram, Professor and Head, Department of AI and DS, NMIT, Bengaluru, Karnataka, India

Preface

The First International Conference on **“Emerging Research in Computing, Information, Communication, Artificial Intelligence and Machine Learning”**, ERCICAM 2024 is an annual event organized at Nitte Meenakshi Institute of Technology (NMIT), Yelahanka, Bengaluru, India.

ERCICAM aims to provide an interdisciplinary forum for discussion among researchers, engineers, and scientists to promote research and exchange of knowledge in computing, information, communication, artificial intelligence, and machine learning. This conference will provide a platform for networking of academicians, engineers, and scientists and will encourage the participants to undertake high-end research in the above thrust areas.

ERCICAM 2024 received more than 300 papers from all over the world, viz. from Nepal, Bangladesh, Malaysia, Bahrain, China, and India. The ERCICAM Technical Review Committee has followed all necessary steps to screen more than 300 papers by going through various rounds of quality checks on each paper before selection for presentation/publication in Springer’s LNEE (Lecture Notes in Electrical Engineering) proceedings, which is SCOPUS indexed.

The acceptance ratio is 1:3.

Bengaluru, India
February 2024

N. Nalini
Sreekanth Rallapalli

Acknowledgments

First, we would like to thank Prof. N. R. Shetty, Advisor, Nitte Education Trust, who has always been the guiding force behind this event's success. It was his dream that we have striven to make a reality. Our sincere thanks to Mr. Rohit Punja, Administrator, Nitte Education Trust, Dr. Sandeep Shastri, Director-Academic, Nitte Education Trust, and Dr. V. Sridhar, Dean-Academic, NMIT, for their encouragement in organizing this conference. Our thanks to Dr. H. C. Nagaraj, Principal, NMIT, who has monitored the whole activity of the conference from the beginning till its successful end.

Our special thanks to Springer for agreeing to publish the quality papers in their LNEE Series.

We would like to express our gratitude to all the review committee members of all the themes of computing, information, communication, artificial intelligence, and machine learning and the best paper award review committee members.

Finally, we would like to express our heartfelt gratitude and warmest thanks to the ERCICAM 2024 organizing committee members for their hard work and outstanding efforts. We know how much time and energy this assignment demanded, and we deeply appreciate all the efforts to make it a grand success.

Our special thanks to all the authors who have contributed to publish their research work in this conference and participated to make this conference a grand success. Thanks to everyone who have directly or indirectly contributed to the success of this conference ERCICAM 2024.

Program Chairs
ERCICAM 2024

About the Conference

ERCICAM 2024

The First International Conference on **“Emerging Research in Computing, Information, Communication, Artificial Intelligence and Machine Learning”**, ERCICAM 2024 is an annual event held during February 23–24, 2024, at Nitte Meenakshi Institute of Technology (NMIT), Yelahanka, Bengaluru, India.

ERCICAM 2024 is organized under the patronage of Prof. N. R. Shetty, Advisor, Nitte Education Trust. Mr. Rohit Punja, Administrator, Nitte Education Trust. Dr. Sandeep Shastri, Director-Academic, Nitte Education Trust, and Dr. V. Sridhar, Dean-Academic, NMIT. Dr. H.C. Nagaraj who is the principal of NMIT acted as a conference chair and the Program Chairs of the conference were Dr. N. Nalini, Professor, CSE and Dr. Sreekanth Rallapalli, Professor and Head, MCA, NMIT, Bengaluru, Karnataka, India.

ERCICAM aims to provide an interdisciplinary forum for discussion among researchers, engineers, and scientists to promote research and exchange of knowledge in computing, information, communication, artificial intelligence, and machine learning. This conference will provide a platform for networking of academicians, engineers, and scientists and will encourage the participants to undertake high-end research in the above thrust areas.

For ERCICAM-2025, authors are invited to submit the manuscripts of their original and unpublished research contributions to ercicam2025@nmit.ac.in (ERCICAM website: <https://www.nmit.ac.in/ercicam>). All the submitted papers will go through a peer review process, and the corresponding authors will be notified about the outcome of the review process. There will be many rounds of quality checks on each paper before selection for Presentation/Publication. Authors of the selected papers to present their papers during the conference.

Contents

Leverage Machine Learning to Infer Proof of the Nipah Influenza	1
D. C. Shubhangi, Baswaraj Gadgay, and S. Anita	
Comparative Analysis of Seizure Prediction Methods	13
R. Priyakanth, N. M. Sai Krishna, Viswanadham Ravuri, K. Buvika, M. Yamini, N. Harathi, and Y. Lakshmi	
Enhancing Crop Yield Through Rainfall Analysis Using Machine Learning Techniques	27
Spandana N. Raj, K. Amuthabala, and Shantala Devi Patil	
Performance Evaluation of Machine Learning Algorithms for Breast Cancer Detection in Women	39
B. Chaitrashree, A. Manasa, Shridhar B. Devamane, Reshma D'Souza, S. Prerana, and K. Pooja	
Sarcasm Detection in Hindi Social Media Text Using HindiBERT-LSTM and XGBoost Models	53
Y. Indira Priyadarshini, K. Mahesh Babu, Y. Bhargavi, Y. Dharani, Y. Ramya, and P. Mani Chandana	
Enhancing Breast Cancer Prediction: A Comparative Analysis of CNN and Ensemble Models with Data Balancing Techniques	67
Y. Indira Priyadarshini, K. Mahesh Babu, Katika Sai, B. Supriya, Ala Chaitanya Kumari, and Epuru Vyshnavi	
Supervised Machine Learning Approach for Diabetes Detection and the Impact of Data Balancing Methods	81
K. Seshadri Ramana, N. Asra Shaheen, S. Safa Chowdary, Shaik Afroz Jaha, Syeda Uzma Tasneem, and Talari Renuka	

Comparative Analysis of Machine Learning Classifiers for Football Fake News Detection: A TextRNN-Based Approach	97
V. Prema Manvi, U. Saipriya, Shaik Suhana, S. Firoz Begum, and S. Abhitha Angel	
ResNetEnsemble Model for Accurate and Robust Detection of Rice Leaf Diseases	111
S. Aslam Shareef, K. Mahesh Babu, Kavalakuntla Bhuvaneshwari, Konda Priyanka, Chegu Sai Varshika, and Bojji Reddy Nikhila	
From Pixels to Captions: Deep Learning Solutions for Accurate Image Description and Retrieval	127
Anju, Aditya Soni, and Er. Neha	
Framework to Analyze the Essential Parameters in Poultry Farming Through Deep Learning Methods for Production Enhancement	139
Deepika, Nagarathna, and Channegowda	
Deep Neural Network-Based Classification of Spyware Populations Using Macroscopic Statistical Attributes of Operation Codes	161
Sonal Malhotra, Deepak Upadhyay, Rahul Chauhan, Arun Balodi, and Swati Devliyal	
Exploring the Effectiveness of Sequence Embedding as a Powerful Tool for Unmasking Deceptive Tactics and Enhancing Insurance Fraud Detection	173
G. L. Vidyashree, Y. Umadevi, R. Kanagavalli, Kumaraswamy, and M. R. Sowmya	
Analyzing the Impact of Social Media Advertising Using Machine Learning Technique	187
B. Roopashri Tantri, Padma Bhaskar, and N. Sandhya	
The Ultimate Air Quality Predictor Using Virus Colony Search Optimisation-Based UATNet Classification	193
Zabiha Khan, R. J. Anandhi, and B. Guna Priya	
Consumer Sentiment Analysis Using ROA-Based Hybrid Deep Learning Model from Twitter Data	219
Sreekanth Rallapalli, K. Jayasheel Kumar, and Tarunika Sharma	
Potato Plant Leave Disease Detection Using TuberNet Model in Smart Agriculture	235
Sreekanth Rallapalli, V. Revathi, and V. Asha	
Detection of Alzheimers Using Imaging and Machine Learning	251
Somu Geetha Sravya, Kamsala Tharun, B. R. Sandhya, Tresa Biju, and Gopi Sagar	

Detection of Fire using Deep Learning Models	269
Mohith Krishna M. Kamath, R. Sinchana, K. Sreenidhi, and Jeshma Nishitha Dsouza	
Anxiety or Depression Prediction of MDD in Children Using Ensemble Learning: A Systematic Review	281
C. R. Anju and Shiny Duella	
Lie Detection Using Audio Classification	297
Tejas Ravindra Rote, Shivama, and M. Narayanamoorthi	
Comparative Study of NLP Tools for the Detection of Cyber Bullying	311
Jayati Bhadra, Navaneeth Kashyap, Kunkum Poovamma, and S. Dutta	
Simulated Intelligence-Based Risk Expectation Models for Diabetic Retinopathy	319
Balluru Adishankar Nithya, Md. Sameeruddin Khan, and R. Chinnaiyan	
Sentimental Analysis Using Machine-Learning Models—A Comparative Study	339
Veda Spandana Tungala, Aarthi Hasitha Yepuri, Sai Sindhuja Anaparthi, Aasritha Sai Vemulapalli, and S. R. Reeja	
Real-Time Object Detection and Classification Using Deep Learning Models	351
Piyush Kumar Pareek, B. R. Shobha Rani, and S. Bharathi	
Multidimensional Object Detection in Real Time Using Deep Learning Techniques	367
Piyush Kumar Pareek, S. A. Shreyanshu, S. Bharathi, and B. R. Shobha Rani	
Enhancing Campus Safety Monitoring: A Machine Learning Approach	381
Meenakshi, G. Bhavyashree, K. S. Prakruthi, and B. S. Keerthi	
AI-Driven Early Detection of Chronic Kidney Disease: Harnessing Deep Learning Techniques	397
Y. Umadevi, G. L. Vidyashree, K. Anitha, R. Kanagavalli, and Kumaraswamy	
Interaction Modeling in Logistic Regression: An Analysis of Dropout Regularization and Adaptive Optimization	413
Niveditha Kundpuram, Subha Vidisha Atilli, and K. S. Srinivas	
Crop Yield Prediction in Karnataka Using Machine Learning Techniques	429
Aman Chhabria and Jayati Bhadra	

Autism Spectrum Disorder (ASD) Prediction Using EfficientNet-B7 and Haar Cascade	443
Anson J. Maliackal, Ancyline J. Maliakkal, and Jayati Bhadra	
Lung Cancer Prediction Using Integrated Deep Learning Model	455
M. V. Sudhamani and Anil Kumar	
Leaf Disease Detection and Classification in Grapevine and Potato Leaves Using Deep Learning	465
A. Hari Krishna, K. Ranjith, and Jeshma Nishitha Dsouza	
A Meticulous Evaluation of Musculoskeletal Fracture Diagnosis and Categorization Methods	479
Shweta Dhareshwar, Manjula Gururaj Rao, and Piyush Kumar Pareek	
Innovative Model of Brain Tumor Classification by Discovering the Potential of ENCO-MSRNet with MRI Imaging	491
S. Renukadevi, K. Rashmi, K. Asha, K. Ramya, C. S. Soumya, and V. Sunil Kumar	
Intelligent Application for Dog Breed Identification Using Deep Learning and Machine Learning	509
Pradnya Narkhede, Anuradha Thakare, Ayush Dongardive, Sakshi Shitole, and Vrushali Guldagad	
A Survey on Machine-Learning Models Used for Detecting Brain Tumors	525
K. S. Supriya, Anil Kumar, S. Sree Priya, R. Supriya, and K. S. Tejaswini	
An Efficient Swarm Optimized Based Machine-Learning Technique for Improving the Prediction of Chronic Diseases	537
Sandeep Kumar Hegde, Rajalaxmi Hegde, and Thangavel Murugan	
Unsupervised Resume Evaluation and Information Extraction: Leveraging Data-Driven Strategies for Applicant Ranking and Data Retrieval	549
V. Jaivignesh and V. UmaRani	
Enhancing Hydroponic Efficiency Using IoT and Machine Learning Algorithms	561
R. Punith Raj, R. Shushank, S. Vaibhav, and M. Shahina Parveen	
Index	573

About the Editors

N. R. Shetty serves as Chancellor of the Central University of Karnataka in Kalaburagi and is Chairman of the Review Commission for State Private Universities in Karnataka. Additionally, he is Advisor at the Nitte Meenakshi Institute of Technology (NMIT) in Bengaluru. He is Founder and Vice-President of the International Federation of Engineering Education Societies (IFEES) in the USA. He previously held the position of Vice-Chancellor at Bengaluru University for two terms and served as President of the Indian Society for Technical Education (ISTE) in New Delhi for three terms. He was also Member of the Executive Committee of the All India Council for Technical Education (AICTE) and chaired its South West Region Committee.

L. M. Patnaik earned his Ph.D. in 1978 in the area of real-time systems and his D.Sc. in 1989 focusing on computer systems and architectures, both from the Indian Institute of Science (IISc) in Bengaluru. From March 2008 to August 2011, he served as Vice-Chancellor of the Defence Institute of Advanced Technology, a Deemed University in Pune. Currently, he is Honorary Professor in the Department of Electronic Systems Engineering at IISc and INSA Senior Scientist and Adjunct Professor at the National Institute of Advanced Studies in Bengaluru. Over his 50 years of service, his teaching, research, and development interests have encompassed parallel and distributed computing, computer architecture, CAD of VLSI systems, high-performance computing, mobile computing, theoretical computer science, real-time systems, soft computing, and computational neuroscience, including machine cognition. He has published 1286 papers in refereed international journals and conference proceedings.

H. C. Nagaraj obtained his B.E. degree in electronics and communication engineering with first-class distinction from the University of Mysore in 1981. He received his M.E. in communication systems from P.S.G. College of Technology, Bharathiar University, in 1984, securing first rank. He completed his doctoral degree

in biomedical signal processing and instrumentation at the Indian Institute of Technology (IIT) Madras in 2000. Since September 2003, he has been serving as Principal of Nitte Meenakshi Institute of Technology in Bengaluru. With approximately 38 years of experience in teaching, research, and administration, Dr. Nagaraj has published numerous papers in national and international journals and conferences and has delivered invited talks in fields such as biomedical signal processing, image processing, and mobile communication. He has also guided several students in their Ph.D. programs.

K. R. Venugopal Former Vice-Chancellor of Bengaluru University, has dedicated over five decades to serving both Bengaluru University and University Visvesvaraya College of Engineering (UVCE). He holds 11 degrees, including two Ph.Ds.—one in Economics from Bengaluru University and another in Computer Science Engineering from IIT Madras. He earned his M.E. degree in Computer Science Engineering from IISc, Bengaluru, and has degrees in electronics and communication, law, journalism, public relations, business finance, industrial relations, mass communication, and economics. Currently, he is pursuing a Ph.D. in law. Dr. Venugopal has previously served as Chairman of the All India Council for Technical Education (AICTE) at the South West Regional Office and as Principal of UVCE for 15 years. He has authored 102 books and published 1207 research papers and holds 40 granted patents. He has supervised 30 Ph.D. students, informally guided 200 research scholars, and supervised over 800 postgraduate dissertations in computer science engineering. He has received the IEEE Fellow and ACM Distinguished Educator awards from the USA for his significant contributions to the field of computer science engineering. He has also been Post-Doctoral Research Scholar and Visiting Professor at the University of Southern California, USA. During his tenure as Vice-Chancellor (2018–2022), he played a key role in restructuring Bengaluru University into four state universities: Bengaluru University, Bengaluru City University, Bengaluru North University, and the University of Visvesvaraya College of Engineering, following the IIT model. Under his leadership, Bengaluru University achieved an A++ grade accreditation from NAAC with a score of 3.75, ranking first among public universities in India.

Sreekanth Rallapalli received his Ph.D. from Bharathiar University, Coimbatore, in 2020. He also holds an M.Phil. and an M.C.A. degree. As Active Researcher, he has published over 35 research articles in various national and international conferences and journals. With 20 years of academic experience, he has worked at universities in India and abroad. Currently, he is Professor at Nitte Meenakshi Institute of Technology in Bengaluru. His research interests include data science, big data, artificial intelligence, and machine learning. He also serves as Reviewer for various journals and conferences.

Leverage Machine Learning to Infer Proof of the Nipah Influenza



D. C. Shubhangi, Baswaraj Gadgay, and S. Anita

Abstract The Nipah virus is a lethal animal and human disease that originates in bats. The urgency in detecting and identifying the virus, driven by a desire to comprehend the efficacy of machine learning, stems from the fact that the virus is lethal. From a medical standpoint, there is currently no vaccine or drug that can cure the Nipah virus. Machine learning algorithms play an essential role in the medical industry when using ML predictions to identify and isolate viruses in critical and questionable situations. In order to determine if a patient has an infection with the Nipah virus, this method will provide numerical data. "Prevention is better than cure" when it comes to the Nipah virus, which does not yet have a vaccine. More and more, machine learning techniques, such as decision trees and random forests, are being used to enhance the accuracy of models.

Keywords Nipah · Restricted Boltzmann machine (RBM) · Random forest (RF) · Decision tree (DT)

D. C. Shubhangi · S. Anita (✉)

Department of Computer Science, Vishveshwarayya Technological University CPGS Kalaburagi, Kalaburagi, Karnataka, India

e-mail: anitaswamy1995@gmail.com

D. C. Shubhangi

e-mail: drshubhangipatil1972@gmail.com

B. Gadgay

Department of Electronics and Communication, Vishveshwarayya Technological University CPGS Kalaburagi, Kalaburagi, Karnataka, India

e-mail: baswaraj_gadgay@vtu.ac.in

S. Anita

Department of Computer Science, Vishveshwarayya Technological University CPGS Kalaburagi, Kalaburagi, Karnataka, India

1 Introduction

A set of numerical phylogenetic indicators derived from computations using DNA sequence allows for the classification of closely related viruses. To estimate role of natural selection in development of a given sequence, we look for such markers that deviate considerably from what would be predicted for random mutations. In Asia, pig farmers have been hit hard by outbreaks of Nipah virus, a zoonotic contagious virus. Over the course of hundreds of years, infectious diseases including SARS, swine flu, MERS, avian flu, Nipah, and Henipa have caused global public health disasters. Infections may linger dormant for years before re-emerging, infecting hundreds of millions of individuals worldwide. NiV, a novel zoonotic paramyxovirus, is closely related to hydra virus. System employs ML models like random forest and decision tree to make Nipah virus forecasts. Such models allow us to forecast geographically dispersed cases of viral infection. That we could be prudent in this matter. Foreground prediction is handled through random forest and decision tree classifier methods. Six symptoms related to Nipah virus are included in dataset being utilized. Furthermore, user's sickness history may be used to inform system's recommendations for safety measures.

2 Related Work

Mathematical models [1] have been built to investigate viral zoonoses in wild animals and point out where further work needs to be done in this field. Author presented an ODE-based mathematical model for characterizing host-pathogen interactions developed a novel host-pathogen model [2] which takes into account seasonal migration and reproduction as well as seasonal changes in the host's surroundings [10]. Three main conclusions can be drawn from this model's analysis. Nipah virus [12] infections in Malaysia and following epidemics in India and Bangladesh have very different epidemiology and medical aspects. In this paper [13], these two viruses were proposed to comprise a novel genus within the Paramyxoviridae family. Nipah virus, like hydra virus, is rare among paramyxoviruses since it could affect or even fatally illness a wide variety of host animals.

3 Existing System

In this study, the fractional dynamics of NiV are examined. It is investigated mathematically and qualitatively how NiV spreads. The qualitative study of NiV fractional model is conducted using fixed-point results, a powerful numerical technique. The numerical outcomes of NiV model are derived using the predictor-corrector

(PC) algorithm. Using nonlinear analysis, the stability of the numerical technique is investigated.

The numerical outcomes are represented graphically in different fractional orders. Numerical simulations are used to show the influence of transmission and infection rates on the development of the suggested model.

Disadvantages:

1. The existing system's prediction of NiV utilizing Restricted Boltzmann Machine might suffer from inadequate precision because of complexity and variability of virus patterns, leading to potential misclassifications and false positives/negatives.
2. The reliance on a single machine learning technique like Restricted Boltzmann Machine may hinder the system's ability to adapt to evolving virus strains and may lack robustness in handling new data sources or emerging patterns.

4 Proposed System

We utilize deep learning through high-level programming. In order to make reliable predictions, data must be sent across many different levels. Here, the work is carried out in a certain order, from one layer to the next, in order to eliminate duplication in layered structure. High-yield strategies for diagnosis treatment have been the subject of a lot of research. In the same vein, we use a Restricted Boltzmann Machine (RBM) to determine if Nipah virus is present in patient's system. It's structured like a bipartite graph, with a layer for intra-node communication. Using random forest and decision tree for comparison study purpose.

Advantages:

1. The proposed system utilizing Restricted Boltzmann Machine offers a potential improvement in accuracy and reliability for Nipah virus prediction, thanks to its ability to learn complex patterns and relationships in data.
2. By leveraging the power of machine learning, the system can adapt and update its predictions based on new data, enhancing its efficacy in monitoring and early detection of Nipah virus outbreaks.

5 Methodology

Dataset: Nipah virus dataset is used for this study purpose. **Data Preprocessing:** In the first round of data preparation, we shall select just three characteristics from data file. The adjacent dataset is divided in training and testing set, with, say, 20% testing data and 80% training data.

Classification

Projected Algorithm

Step 1: Preprocess dataset.

Step 2: Divide dataset into training and testing samples.

Step 3: Apply classifiers RF and DT. For solving regression problems, MSE for data branches from each node.

Step 4: Predict NiVD infection.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (f_i - y_i)^2, \quad (1)$$

where N is total number of observations, f_i is predicted value, and y_i is observed value at observation i .

Using this technique, you can determine how far off your predictions are for every node in forest, giving you better idea of which path to take. Here, f_i is result provided by the decision tree, and y_i is data point's value at node in question. Utilizing Gini index—a measure for determining how nodes upon decision tree branch—is common practice when running random forests upon classification data.

$$\text{Gini} = 1 - \sum_{i=1}^c (p_i)^2. \quad (2)$$

Gini of every branch upon node is calculated using class and likelihood to establish most probable branch. Number of classes, c , is denoted by, while frequency, p_i , of class being seen in dataset is denoted by.

A decision tree's branching structure may also be calculated using entropy.

$$\text{Entropy} = \sum_{i=1}^c -p_i \times \log_2(p_i). \quad (3)$$

When deciding which way node must branch off in network, entropy considers likelihood of various outcomes. Logarithmic function employed in its calculation makes it more complex than Gini index.

Feature Extraction: In feature extraction equations related with shape aspects.

$$\text{Solidity} = \frac{\text{Area of Nucleus}}{\text{Area of Convex Hull}}. \quad (4)$$

Solidity is calculated as the ratio of the area of nucleus that is the membrane enclosed organelle within cell that contains a chromosome an array of hole to the area of Convexhull that is the area of compact region from the set of discrete observations. If these to ratio is equal to 1 it indicates a solid object if it is less it indicates the object an irregular boundary or containing holes.

$$\text{Convexity} = \frac{\text{Perimeter of Convex Hull}}{\text{Perimeter of Nucleus}}$$

$$\text{Circularity} = \frac{(\text{Perimeter of Nucleus})^2}{4 \times \Pi \times (\text{Area of Nucleus})} \quad (5)$$

Convexity is to check the extracted blood cell shape which is curved or round by considering the boundaries of Convexhull and nucleus of blood cell.

Circularity is to check whether the extracted blood cell is being shaped like round or not. If circularity is equals to 1 indicates the perfect round or circle if circularity goes down as far as 0 indicates highly Non- circular shape.

Color qualities make up second set of attributes. Hematologist's knowledge suggests that, in along with form characteristics, random forest: Covariate selection and explanatory power evaluation are two of its main applications. Using random selections of response variable and covariates, several decision trees are built, and predictions are averaged utilizing this machine learning technique. When compared to using a single decision tree, overfitting is greatly reduced when many trees are constructed and averaged. Python was used for linear modeling and random forests.

Random Forest Algorithm

Step 1: The data or training set may be split into random samples.

Step 2: Each piece of training information will be used by the algorithm to create a decision tree.

Step 3: Voting process will be carried out by averaging tree of decisions.

Step 4: In the end, choose forecast outcome that received most votes and use that one.

Pseudocode

```

To generate c classifiers:
for i = 1 to c do
  Randomly sample the training data D with replacement to produce Di
  Create a root node, Ni containing Di
  call BuildTree (Ni)
end for
BuildTree(N):
if N contains instances of only one class then
  return
else
  Randomly select x % of the possible splitting features in N
  select the feature F with the highest information gain to split on
  Create f child nodes of N, N1, ... Nf, where F as f possible values (F1, ... Ff)
  for i = 1 to f do
    set the contents of Ni to Di, where Di is all instances in N that match F
  Call BuildTree (Ni)
end for

```

end if

Ensemble is the term for these merged models. The ensemble employs two strategies:

Bagging: Bagging refers to the process of creating a new training subset from existing sample training data by swapping out some of the samples. The ultimate result is decided by a simple majority.

Boosting: Boosting is the process of combining many low-performing learners into a single high-performing learner via the use of sequential model creation. BOOST, XG BOOST, ADA BOOST, etc.

Decision Tree Algorithm

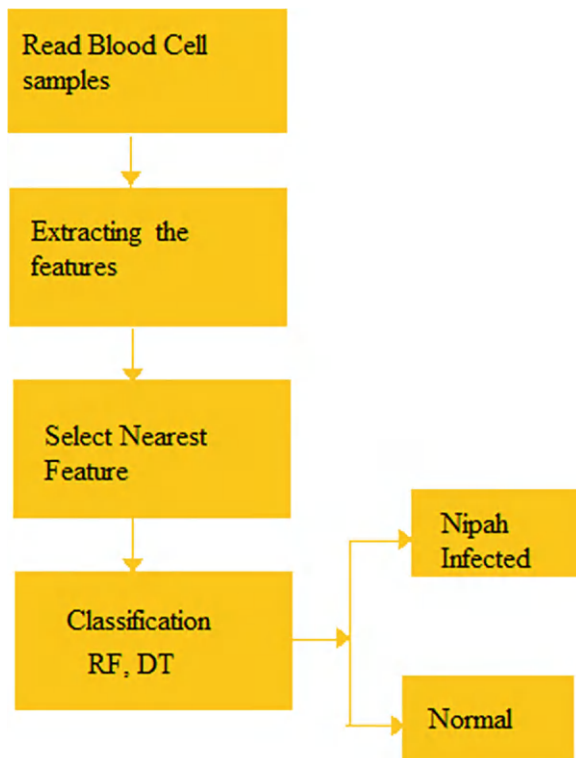
- **Step-1:** According to S, the whole Nipah virus dataset may be found at tree's root node.
- **Step-2:** Use Attribute Selection Measure (ASM) to zero down on most useful characteristic of data collection.
- **Step-3:** Separate S in subsets containing candidate values for most desirable characteristics.
- **Step-4:** Best attribute should be generated as node of decision tree.
- **Step-5:** Construct more decision trees utilizing filtered datasets from Step-3 in a recursive fashion. Keep going till you reach viral prediction stage wherein you're no longer categorize prediction-nodes, at which point you will have leaf node.

6 System Architecture

Architecture Description: As per diagram, the system accepts the blood cell samples. Extracts the features, then applies classification like RF and DT. Then, the system predicts whether the patient is Nipah virus infected or not (Fig. 1).

7 Results and Discussion

Fractal dimension and Shannon entropy were used to analyze the Nipah virus's glycoprotein and nucleoprotein sequences. The phylogenetic analysis is based on the variation in the atomic number of nucleotides. The major findings of conventional phylogeny analysis are reflected in the classification, although the classification is better at distinguishing between closely related strains. Glycoprotein sequence GC pair content as a function of fractal dimension. Differential development of cell and di-nucleotide entropy in nucleoprotein. Extrapolating from Nipah and Spanish flu-like viruses, low fractal dimension in nucleoprotein sequence may serve as a diagnostic indicator of these pathogens. There are two main flaws in Shannon's method. To begin, it can't be used to evaluate how various scales of diversity distributions

Fig. 1 System architecture

stack up against one another. Second, it isn't able to evaluate subsets of diversity distributions against the whole set. To overcome this problem purpose, we are using ML algorithms in our system.

8 Implementation

The patient dataset is collected in terms of patient ID, age, and sex (Fig. 2).

It performs preprocessing which converts the image into grayscale and using morphological technique denoise the image (Figs. 3, 4, and 5).

(a) Grayscale Image

Grayscale image is 8bit image with each pixel having 256 combination of shades, it measures the values ranging between 0 to 255 each gray scale color is having equal values of Red green blue components of the RGB color model change the color image to black and white image.

(b) Threshold Image

	PatientID	Age	Sex	...	Seizures	Coma	Brain swelling (encephalitis)
0	1	56	1	...	1	1	1
1	2	77	1	...	0	1	1
2	3	58	0	...	0	0	0
3	4	76	0	...	1	0	0
4	5	87	1	...	1	1	0

[5 rows x 13 columns]

Fig. 2 Nipah virus dataset

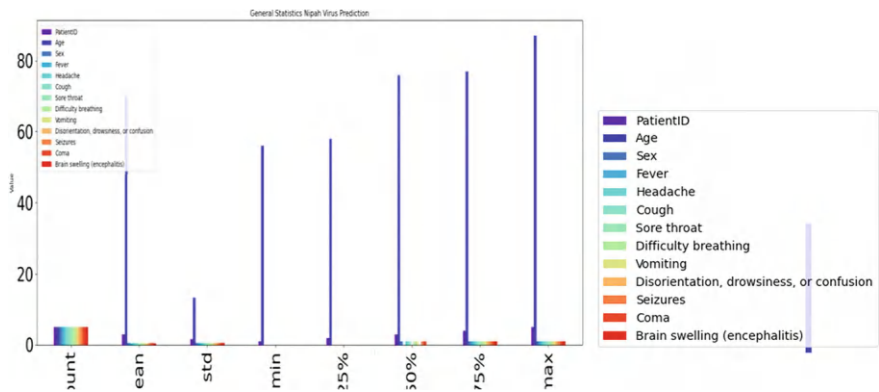


Fig. 3 Attributes versus count graph statistics analysis of Nipah virus attributes

Threshold image will partition the image sample into foreground and background and convert the grayscale into binary images: if the image intensity is less than threshold fixed value, convert the image into black color; if the intensity of image is greater than threshold fixed value, convert the image into white color. In the threshold we consider only intensity of the image not any relationship between the pixels (Fig. 6).

The sample or identical images are used both for training and testing while test accuracy represents that the trained model identifies that independent images were not used in training. Validation accuracy represents a percentage of how predictions hit their labels of data (Figs. 7 and 8).

When the percentage of training and validation accuracy could not hit the maximum labels of data, we predict that patient blood cell sample is effected by virus.



Fig. 4 Blood cell sample patient. Blood cell sample is taken for testing purpose

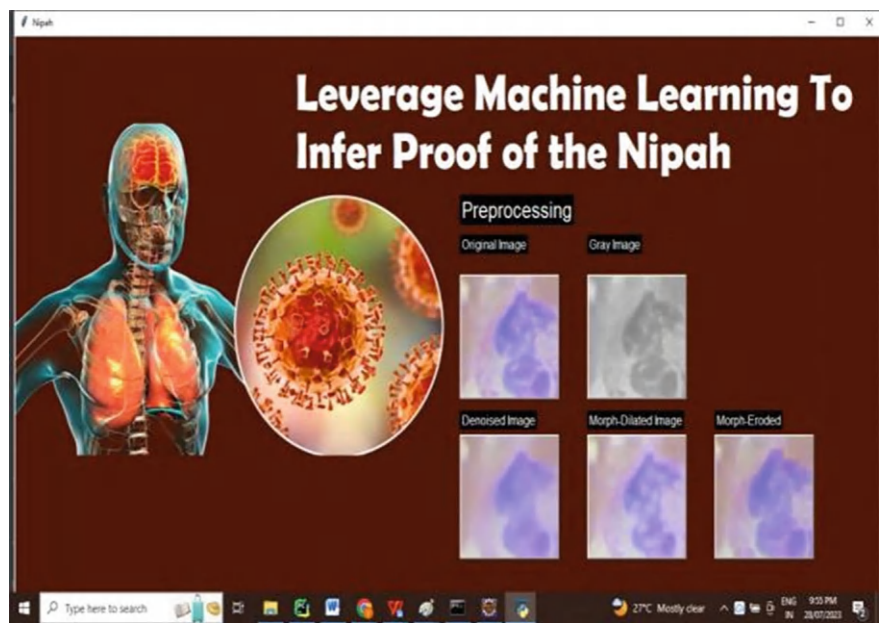


Fig. 5 Preprocessing

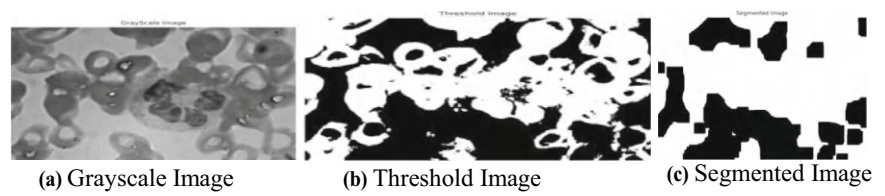


Fig. 6 Segmentation is process of dividing a picture into smaller pieces, called regions or segments and group pixels into clusters based on their similarities where each cluster represents a segment

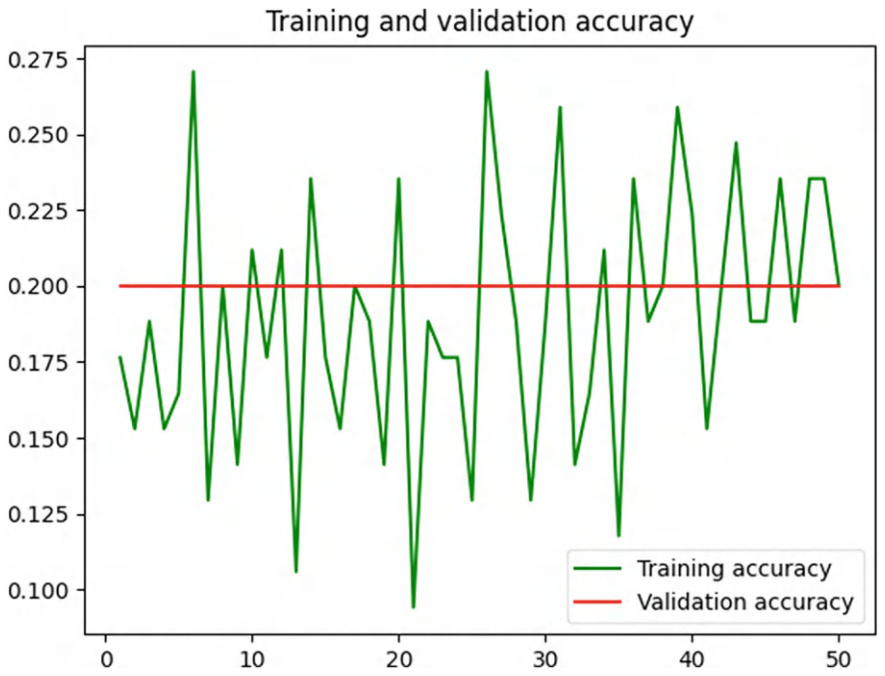
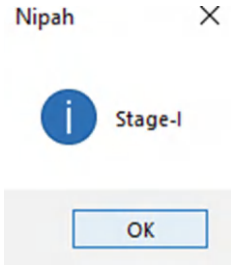


Fig. 7 Training and validation accuracies

Fig. 8 Prediction



9 Conclusion

Date palm tree sap is the primary vector for the transmission of Nipah virus from bats to people. If more people knew about these issues, maybe the focus would move from treatment to prevention. Random forest and decision tree are two machine learning approaches utilized to improve the accuracy of the models in this research. The model may be improved in future by using IoT.

References

1. Allen LJS, Brown VL, Jonson CB, Klein SL, Lavery SM, Magwedere K, Owen JC, Van Den Driessche P (2012) Mathematical modelling of viral zoonoses in wildlife. *Nat Resour Model* 25(1):5–51
2. Biswas MHA (2014) Optimal control of Nipah virus (NiV) infections: a Bangladesh scenario. *J Pure Appl Math Adv Appl* 12(1):77–104
3. Patil RJ, Mulage I, Patil N (2023) Smart agriculture using IoT and machine learning. *J Sci Res Technol* 1(3):47–59
4. Computer Engineering, University of Porto, Portugal on 28–29 June 2012
5. Biswas MHA, Paiva LT, de Pinho MDR (2014) A SEIR model for control of infectious diseases with constraints. *Math Biosci Eng* 11(4):761–784
6. Kumar N, Hanumanth N, Patil J (2023) Flight delay prediction based on aviation big data and machine learning. *J Sci Res Technol* 1(7):58–67
7. Parimala (2023) Machine learning technique for prediction of wind and rainfall using underwater measurement. *J Sci Res Technol* 1(6):124–135
8. Chua KB (2003) Nipah virus outbreak in Malaysia. *J Clin Virol* 26(3):265–275
9. Gulmeher R, Aiman U (2023) A novel approach to unveiling employee attrition patterns using machine learning algorithms. *J Sci Res Technol* 1(6):234–241
10. Breban R, Drake JM, Stallknecht DE, Rohani P (2009) The role of environmental transmission in recurrent avian influenza epidemics. *PLoS Comput Biol* 5(4):1–11
11. Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Applications of mathematics, no 1. Springer Verlag, New York
12. Sharma S, Sharma V (2023) Comparison of machine learning techniques in the diagnosis of erythematous squamous disease. *J Sci Res Technol* 1(4):1–9
13. Chua KB, Bellini WJ, Rota PA et al (2000) Nipah virus: a recently emergent deadly paramyxovirus. *Science* 288(5470):1432–1435
14. Geisbert TW, Daddario-DiCaprio KM, Hickey AC, Smith MA, Chan YP et al (2010) Development of an acute and highly pathogenic nonhuman primate model of Nipah virus infection. *PLoS ONE* 5(5):1–12

Comparative Analysis of Seizure Prediction Methods



R. Priyakanth, N. M. Sai Krishna, Viswanadham Ravuri, K. Buvika, M. Yamini, N. Harathi, and Y. Lakshmi

Abstract The onset of epileptic seizures is attributed to irregularities in the brain, which can have an impact on a person's overall well-being. Seizures come on suddenly, with no warning signs, thereby elevating the risk of fatality. Put simply, a seizure arises when a large number of brain cells become stimulated simultaneously. Nearly 1% of people worldwide experience epileptic seizures. The prediction of seizures indeed remains a complex challenge. While advancements in technology and research have improved our understanding, the limited amount of data available can hinder progress. There are various methods employed to predict seizures, such as techniques based on frequency, statistical assessment of EEG signals, non-linear dynamics involving chaos, and the use of intelligent engineered systems. Forecasting seizures prior to their onset is advantageous in terms of preventing seizures through medication. Contemporary computational tools, along with machine learning and deep learning methods, are presently being employed to anticipate seizures utilizing EEG. The research in this paper entailed a comparative analysis of diverse machine learning algorithms employed to train a model specifically designed for predicting episodes of seizures.

Keywords Seizures prediction · Colab · EEG data · Machine learning · Regression models

R. Priyakanth (✉) · N. M. Sai Krishna · V. Ravuri · K. Buvika · M. Yamini · N. Harathi · Y. Lakshmi

BVRIT HYDERABAD College of Engineering for Women, Hyderabad, Telangana, India

e-mail: priyakanth.r@bvrithyderabad.edu.in

N. M. Sai Krishna

e-mail: saikrishnakumar.n@bvrithyderabad.edu.in

V. Ravuri

e-mail: viswanadh.r@bvrithyderabad.edu.in

1 Introduction

About 70 million people out of the total population suffer from epileptic seizures, accounting to one sixth of global burden, 90% of these people reside in developing countries like India [1]. Estimates vary, but studies indicate that around 1.16 out of every 1000 individuals with epilepsy experience SUDEP annually [2]. Seizures can affect anyone regardless of age, gender, or ethnicity, and epilepsy is a chronic and severe non-communicable brain disorder with global prevalence. Daily life can be challenging for those with epilepsy, who must take precautions to manage their condition. Seizures can result in injury and pose risks to the patient and others, particularly in industries involving heavy machinery or when operating vehicles. While epilepsy cannot be cured, it can be managed with medication and other techniques. Researchers primarily use Electroencephalogram (EEG) signals to study abnormal brain activity during seizures in epilepsy studies. It occurs in three different phases, preictal, ictal and postictal [3].

Although the prevalence of epilepsy is slightly higher in males, both genders experience concerns about potential seizures in public situations. Epileptic individuals may feel vulnerable when traveling alone, driving, or swimming, limiting their freedom of movement. Researchers are currently exploring ways to mitigate the risks associated with epilepsy and enhance the quality of life for affected individuals [4]. They must exercise the proper caution in order to deal with this sickness. When a seizure happens, it may cause harm or even put the patient or others' lives in danger, especially when it comes to working with heavy machinery or operating automobiles. Although epilepsy is a permanent condition, it can be managed with drugs and other techniques.

2 Methodology

Classification methods (see Fig. 1) include various machine learning models and a regression method.

2.1 *Signal Acquisition and Pre-processing*

The EEG signals are collected from seizure and non-seizure patients and are pre-processed. The pre-processing stage here includes sampling the datasets. This sampled data is then utilized for the training stage.

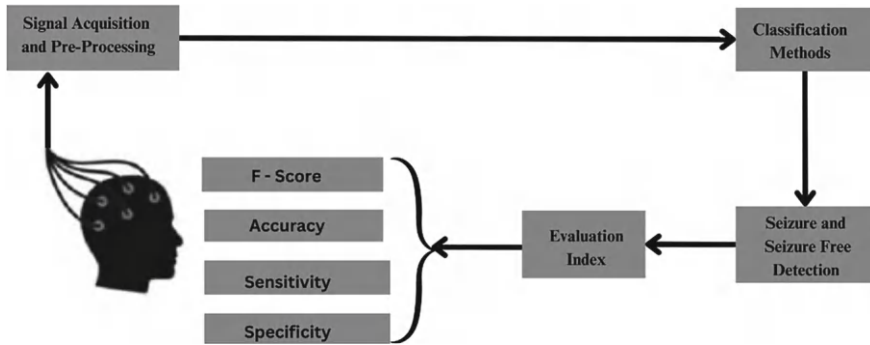


Fig. 1 Epileptic seizure prediction method

2.2 Classification Method

In this stage, the EEG signals that were sampled are segregated into training and testing samples. Various classification algorithms are applied to train the training samples.

2.3 Seizure and Seizure Free Decision

Once the training part is done the trained model will be further used for prediction. In the prediction stage, the trained model will be given a data sample from the testing samples. From the values in the testing sample the model will analyze and predict the seizure sample.

2.4 Evaluation Index

The evaluation of different classification models is done with different performance metrics after predicting the samples [5].

3 Software Requirements

Google Colab Notebook is used for training and prediction of our models. Colab enables users to build and run Python code in the web browser with free GPU access, simple sharing and no additional configuration setup.

Colab notebooks offer a convenient process of combining runnable code and well-formatted text, as well as visuals like images, LaTeX, HTML, and more, all in one file. These notebooks are saved in your Google Drive when you create them and can be easily shared with colleagues or acquaintances, granting them the ability to provide feedback or even make edits. Colab notebooks are operated on Colab, serving as Jupyter notebooks.

One of the most famous programming languages, Python is used to code Machine Learning Models. Python has versatile languages which support GUI. There are modules pre-defined in python, which can be used for carrying out actions like feature extraction, data-cleaning, data mining, etc. The user can completely utilize well-known Python packages to analyze and visualize data with Colab.

By using Colab, the user will have the ability to quickly import a collection of images, conduct training on a specific image classification model, and assess its performance, all with minimal code requirements. The execution of code in Colab notebooks takes place on Google's cloud servers, which means that irrespective of the capacity of the device, the user can access the computational prowess of Google hardware such as TPUs and GPUs through the browser.

Currently, Python has an outstanding reputation as one of the leading programming languages in the realm of machine learning, having replaced numerous languages in the industry. This success can be attributed to its wide range of libraries. By utilizing various libraries in Python, various models were implemented.

4 EEG Data Recording

The EEG signal data is categorized through the spectral components as alpha, beta, delta, and theta, which encompass an extensive sweep of spectral components. Within the EEG spectra, there are discernible wave shapes which are predominantly distributed across four different frequency bands: delta (0.5–4 Hz), theta (4–8 Hz), alpha (8–13 Hz), and beta (13–30 Hz). This study collected EEG data sets from various age groups who were admitted to Dicle University's Medical Faculty Hospital's neurology department for uncontrolled seizures and were diagnosed with epilepsy. The system utilizes the high-level programming language, LabView, to process EEG data obtained from 400 patients, with 200 diagnosed with epilepsy and 200 classified as normal. The data set consists of signals from both healthy and epileptic patients, collected using the advanced multifunction DAQ system. This system facilitates data handling in real-time and functions as a medium for transportation of data between the PC and signal processing system ensuring the accurate measurement and recording of EEG signals. Traditionally, diagnosing such diseases can take hours, but with this system, the process is greatly expedited, allowing for faster and more effective treatment [6].

Electroencephalogram (EEG) signals are utilized in sections to obtain valuable insights into brain activity. The signals are sampled quickly at 173 Hz and adhere to the internationally recognized array of electrodes with 10–20 in number. The

sampled EEG information undergoes precise 12-bit A-D conversion and is subsequently processed through a bandpass filter with the specified band of 0.53–40 Hz. This filtration allows accurate focus on the EEG signals that have clinical relevance. With this cutting-edge technology, researchers can access reliable EEG data and gain a deeper understanding of various brain-related disorders [7]. For this comparative study, EEG data is obtained from a 24 h recording of both individuals with epilepsy and those without. The primary focus is on specific bipolar EEG channels for the analysis. Testing the classifier employed in this study involved analysis of five hundred EEG samples that included surge and upsurge complexes, relics, and regular EEG background were considered [8]. The data set at hand is an initially processed and reorganized form of an often-used dataset that features the senses of epileptic seizures [9].

The data consisting of 4097 points was divided and randomly rearranged into 23 groups, with each group containing 178 points representing one second of EEG recording. Each data point represents a distinct moment in time. This process has generated 11,500 rows of information, with each row containing 178 data points representing one second of EEG recording. The final column of each row indicates the label y , which can take on one of five values (1, 2, 3, 4, or 5).

1. Recording of seizure activity.
2. Captured the EEG data from the region of the tumor.
3. Locating the tumor's location in the brain and capturing the EEG activity from the healthy brain region.
4. Eyes closed indicates that the patient had closed their eyes while the EEG signal was being captured.
5. Eyes open denotes that the subject maintained their eyes open during the recording of the brain's EEG data

Class 2, Class 3, Class 4, and Class 5 subjects are those who did not have epileptic seizures, whereas Class 1 subjects were the only ones who did. The reason for creating a .csv version of the data was to facilitate its accessibility. Although there are five classes in the dataset, most researchers have conducted binary classification, specifically distinguishing class 1 (epileptic seizure) from the other four. The UCI Machine Learning Repository was used to retrieve this dataset.

5 Predictions on Colab

Prediction results with the data set considered in Colab (see Figs. 2 and 3).

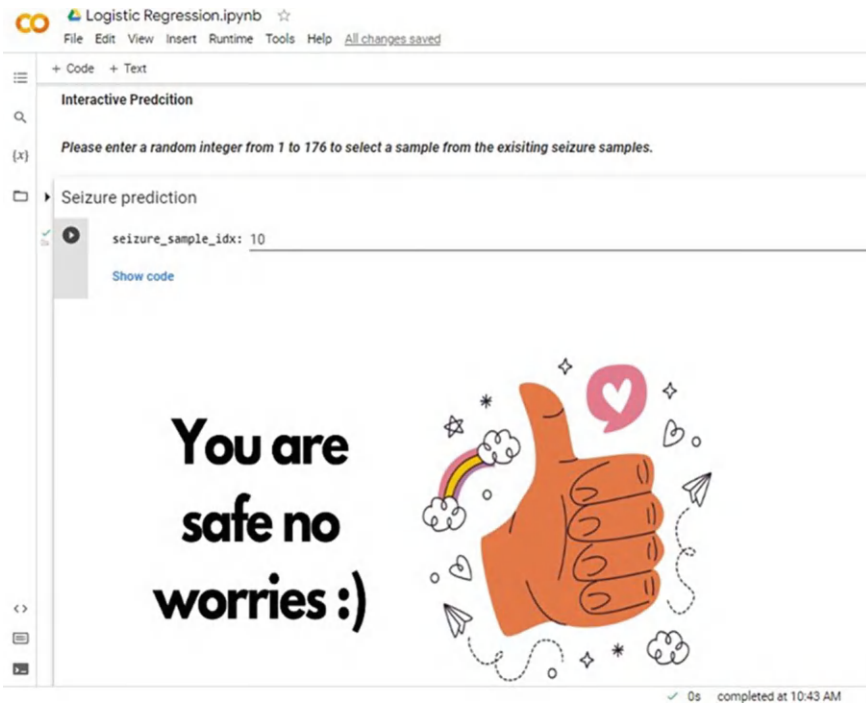


Fig. 2 Seizure sample prediction

6 Comparison of Various Machine Learning Algorithms Implemented for Seizure Prediction

Machine learning and Artificial Intelligence is a domain which is developing drastically in the industry. New innovations are made using machine learning and deep learning models. The algorithms which were considered in this comparative analysis are the most popular kind in the classification domain of machine learning. The comparative analysis of the models used to predict seizure is described as below.

6.1 Support Vector Machine Algorithm

The support vector machine, sometimes known as SVM, is one of the most well-known methods for supervised learning. Its algorithm is described below. The goal of the Support Vector Machine (SVM) technique is to determine the ideal line or decision boundary that permits the partitioning of the n-dimensional space into classes. This will enable users to rapidly categorize new data points in the future. This optimal decision boundary is referred to as a hyperplane in the scientific community. The

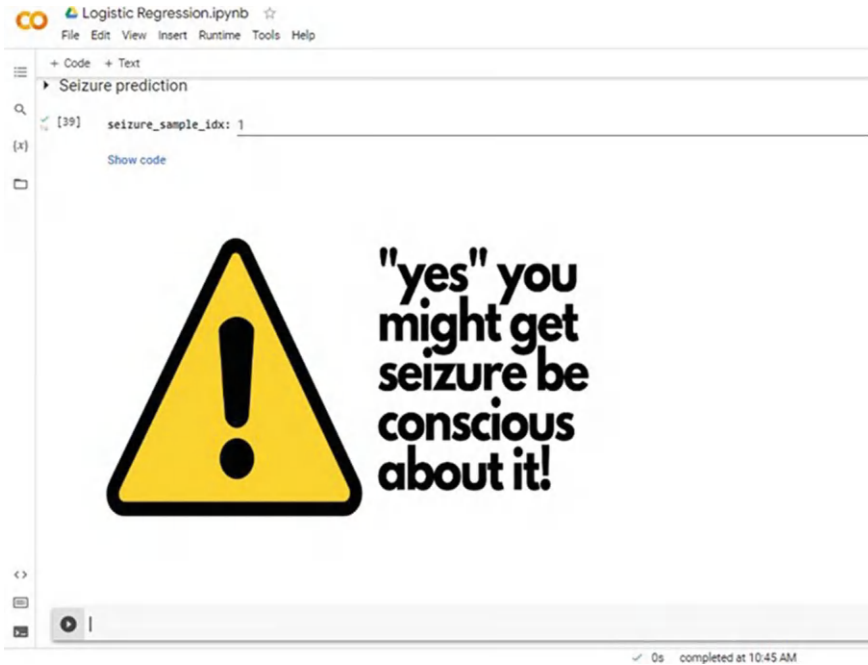


Fig. 3 Non-seizure sample prediction

SVM is responsible for selecting the extreme vectors and points that are used in the construction of the hyperplane. As a result of these very challenging conditions, the method in question is referred to as a “support vector machine.” [10, 11].

6.2 *K-Nearest Neighbor (KNN)*

In the K-Nearest Neighbor algorithm, the new data is assigned to the category that is most similar to the available categories. This is done on the assumption that there is similarity between the new data and the data that already exists. The algorithm makes a determination regarding the classification of a new data point based on how similar it is to the data that is already stored and used to make the determination. Because of this, the KNN algorithm is able to quickly classify newly acquired data into the most relevant category possible. The KNN technique can be used for classification problems as well as regression problems, although the majority of the time, it is employed for classification problems [12, 13].

6.3 *Random Forest*

The Random Forest is a reliable machine learning algorithm that can perform various tasks, such as regression and classification. As an ensemble method, a Random Forest model comprises many small decision trees known as estimators, each providing its own set of predictions. By using a Random Forest, the restrictions of a decision tree algorithm can be overcome, such as overfitting of datasets, resulting in improved accuracy. Additionally, the Random Forest generates predictions without necessitating extensive configuration within packages [14].

6.4 *Multi-layer Perceptron Classifier*

The MLPClassifier updates the parameters at each time step using iterative training. This is accomplished by computing the partial derivatives of the loss function with reference to the selected model parameters. In addition, it is possible to incorporate a regularization term into the loss function in order to cut down on the number of model parameters and prevent overfitting [15]. This implementation is capable of functioning with either dense or sparse arrays containing floating-point data. There are three layers that make up the MLPClassifier, but one of them is concealed. When there is more than one hidden layer within an artificial neural network, it is referred to as a deep artificial neural network. One example of a feedforward artificial neural network is the MLPClassifier, which can be found in [16].

6.5 *Decision Tree*

The supervised-learning method named Decision Tree is applicable to both classification problems as well as regression problems, but it is mostly used for classification. It works as a tree-structured classifier, with core nodes indicating a dataset's properties, branches presenting decision rules, and leaf nodes describing outcomes. Decision Nodes are utilized for making choices and possess multiple branches, while Leaf Nodes are the outcomes of those decisions and do not possess additional branches [17].

6.6 *Logistic Regression*

The procedure known as logistic regression is used for classifying data in order to determine the likelihood that something will be successful or unsuccessful. When the dependent variable is binary, such as 0/1, true/false, or yes/no, it is permissible to use

this method. Logistic regression examines the association among a given collection of labeled data in order to classify the data into discrete categories. This is accomplished by using the data as a given. It begins by constructing a linear relationship based on the dataset that is at hand, and then it adds a non-linear component in the form of the sigmoid function. This approach is predicated on the sigmoid function, in which the output is a probability and the input can take any value between a minus infinity and a plus infinity.

Each of the machine learning models is analyzed by calculating different parameters. These parameters include train time complexity, prediction efficiency, accuracy, and computational complexity. These parameters are selected according to the application; here accuracy, latency and ease of implementation were the final features of interest which play a key role in real-time working. Every ML model has a separate accuracy formula; a few of these analysis parameters are also part of the ML model library. Logistic regression has $O(m)$ space and test time complexity, KNN model's space and test time complexity are analyzed using formula $O(m \times n)$, where m indicates the number of features and n indicates the number of training examples. Support vector machine (SVM) has $O(n \times m)$ and $O(n' \times m)$, test time and space complexity, n' here is the number of support vectors. The test time and space complexity for decision tree are $O(m)$ and $O(\text{tree depth})$, for random forest it is given as $O(m \times k)$ and $O(k' \times \text{depth of tree})$ where k is the neighbors and k' denotes number of trees. Multi-layer-perceptron has complexity as $O(n^2)$, n denoting the neurons.

7 Results

See Table 1.

7.1 Bar Graph Analysis of Model Performance

Five best classification models and a regression model were explored (see Table 1), using these prediction process of seizure attack is performed. The performance of these models (see Fig. 4) is evaluated in terms of train time complexity, prediction efficiency, computational complexity, and accuracy. Every model has its own advantages and disadvantages; the performance might not just be because of the model's incapacibilities, but also because of the inconsistency in the data used. Logistic regression has good level of accuracy and prediction with less computational complexity whereas the others have to compromise between accuracy and prediction.

Table 1 Comparison table for algorithms implemented

Name of the model	Train time complexity	Prediction efficiency	Accuracy	Computational complexity
Support vector machine	$O(nd^2)$ $n \rightarrow$ no. of data points $d \rightarrow$ dimension	Low	98.2	1,628,400 $m \rightarrow 177$ $n \rightarrow 9200$
K-nearest neighbors	$O(nd)$ $n \rightarrow$ no. of data points $d \rightarrow$ no. of features	Average	91.78	1,628,400 $m \rightarrow 177$ $n \rightarrow 9200$
Random forest	$O(nd)$ $n \rightarrow$ size of random forest $d \rightarrow$ depth of random forest	Low	91.70	17,700 $m \rightarrow 177$ $k \rightarrow 100$
Multi-layer perceptron	$O(n^2)$ $n \rightarrow$ no. of neurons	Low	97.13	22,500 $n \rightarrow 150$ n here is max number of neurons in a layer
Decision tree	$O(n*\log n*d)$ $n \rightarrow$ no. of data points $d \rightarrow$ no. of features	Average	89.98	177 $m \rightarrow 177$
Logistic regression	$O(nd)$ $n \rightarrow$ size of random forest $d \rightarrow$ depth of random forest	Better	82.47	177 $m \rightarrow 177$

8 Conclusion

AI/ML is one of the booming technologies of the current world. So many applications developed using these technologies for commercial usage. It would help save the lives of many if these technologies are used for predicting health issues like epileptic seizures. This paper is a trial to use the AI/ML models for analyzing the models to predict seizures.

As an extension to the analysis, the best model according to accuracy, SVM can be used to develop a IoT based device which would alert the patient about the attack, using which they can position themselves in a safer place or be informed about their condition. The device can also have add-on features such as alarming the care-takers, analyzing the occurrence, and suggesting precautions.

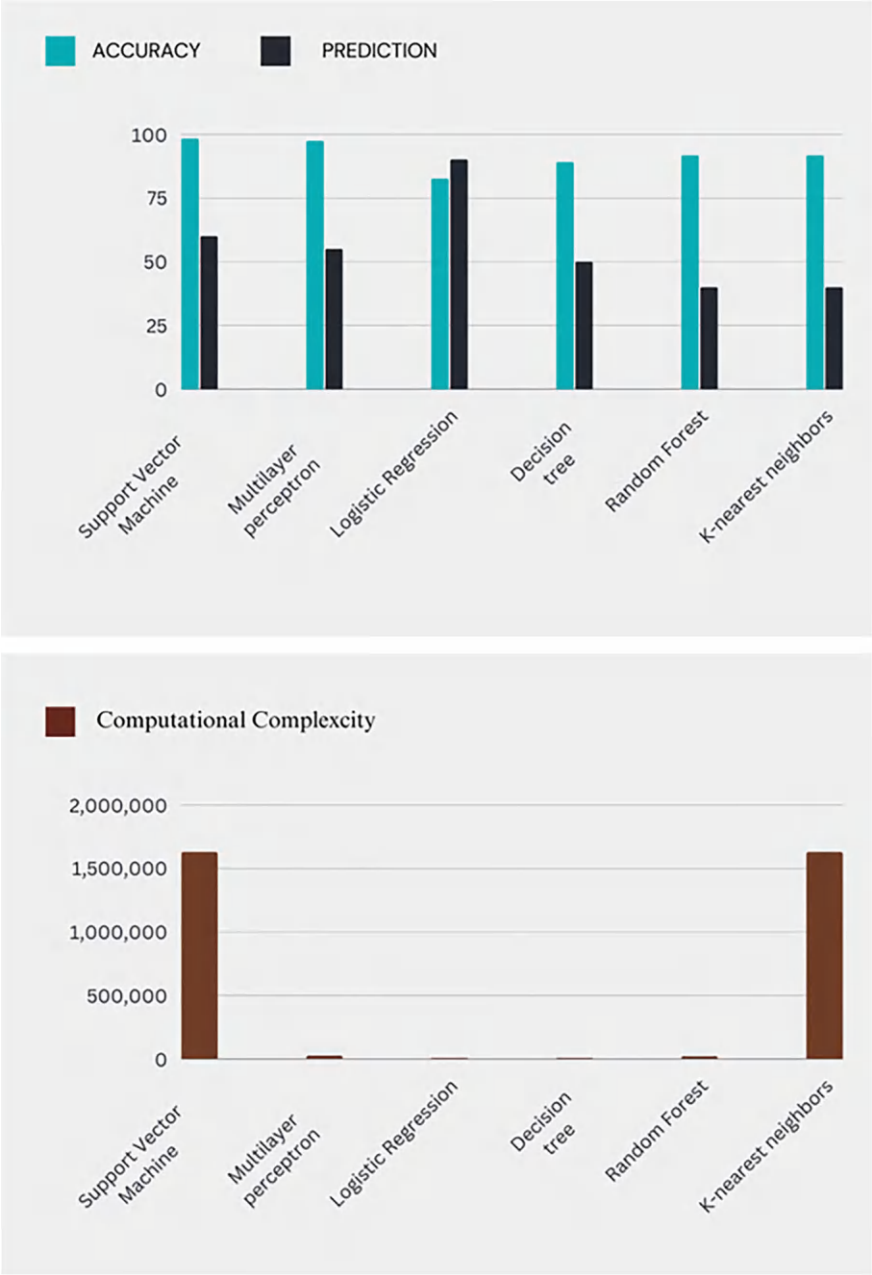


Fig. 4 Bar graph for model performance

References

1. Gururaj G, Satishchandra P, Amudhan S (2015) Epilepsy in India I: epidemiology and public health. *Ann Indian Acad Neurol* 18(3):263. <https://doi.org/10.4103/0972-2327.160093>
2. Thurman DJ, Hesdorffer DC, French JA (2014) Sudden unexpected death in epilepsy: assessing the public health burden. *Epilepsia* 55(10):1479–1485. <https://doi.org/10.1111/epi.12666>
3. Natu M, Bachute M, Gite S, Kotecha K, Vidyarthi A (2022) Review on epileptic seizure prediction: machine learning and deep learning approaches. *Comput Math Methods Med* 2022:1–17. <https://doi.org/10.1155/2022/7751263>
4. Parvez MZ, Paul M (2016) Epileptic seizure prediction by exploiting spatiotemporal relationship of EEG signals using phase correlation. In: *IEEE transactions on neural systems and rehabilitation engineering*, vol 24, no 1. IEEE Engineering in Medicine and Biology Society, pp 158–168. <https://doi.org/10.1109/tnsre.2015.2458982>
5. Yang Y, Zhou M, Niu Y, Li C, Cao R, Wang B, Yan P, Ma Y, Xiang J (2018) Epileptic seizure prediction based on permutation entropy. *Front Comput Neurosci* 12. <https://doi.org/10.3389/fncom.2018.00055>
6. Shahidi Zandi A, Tafreshi R, Javidan M, Dumont GA (2013) Predicting epileptic seizures in scalp EEG based on a variational Bayesian Gaussian mixture model of zero-crossing intervals. *IEEE Trans Biomed Eng* 60(5):1401–1413 [Internet] [cited 2023 Dec 3]. Available from: <https://pubmed.ncbi.nlm.nih.gov/23292785/>
7. Kaggle (2018) Machine learning algorithms for epileptic seizures [Internet]. Kaggle.com. [cited 2023 Dec 3]. Available from: <https://www.kaggle.com/code/harunshimanto/machine-learning-algorithms-for-epileptic-seizures>
8. Islam MK, Rastegarnia A, Yang Z (2016) A wavelet-based artifact reduction from scalp EEG for epileptic seizure detection. *IEEE J Biomed Health Inform* 20(5):1321–1332 [Internet] [cited 2023 Dec 3]. Available from: <https://pubmed.ncbi.nlm.nih.gov/26186797/>
9. UCI machine learning repository [Internet]. Uci.edu [cited 2023 Dec 3]. Available from: <https://archive.ics.uci.edu/dataset/388/epileptic+seizure+recognition>. Gandhi R (2018) Support vector machine—introduction to machine learning algorithms. Towards Data Science
10. Gandhi R (2018) Support vector machine—introduction to machine learning algorithms [Internet]. Towards Data Science [cited 2023 Dec 3]. Available from: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
11. Panda R, Khobragade PS, Jambhule PD, Jengthe SN, Pal PR, Gandhi TK (2010) Classification of EEG signal using wavelet transform and support vector machine for epileptic seizure diction. In: *2010 International conference on systems in medicine and biology*. IEEE, pp 405–408
12. Uddin S, Haque I, Lu H, Moni MA, Gide E (2022) Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Sci Rep* 12(1) [Internet] [cited 2023 Dec 3]. Available from: <https://doi.org/10.1038/s41598-022-10358-x>
13. Manjusha M, Harikumar R (2016) Performance analysis of KNN classifier and K-means clustering for robust classification of epilepsy from EEG signals. In: *2016 International conference on wireless communications, signal processing and networking (WiSPNET)*. IEEE
14. Basri A, Arif M (2021) Classification of seizure types using random forest classifier. *Adv Sci Technol Res J* 15(3):167–178 [Internet] [cited 2023 Dec 3]. Available from: <http://www.astrj.com/Classification-of-Seizure-Types-Using-Random-Forest-Classifer,140542,0,2.html>
15. Bisong E (2019) Logistic regression. In: *Building machine learning and deep learning models on Google Cloud platform*. Apress, Berkeley, CA, pp 243–250
16. Keijsers NLW (2010) Neural networks. In: *Kompolti K, Metman LV (eds) Encyclopedia of movement disorders*. Elsevier, San Diego, CA, pp 257–259

17. Charbuty B, Abdulazeez A. Classification based on decision tree algorithm for machine learning. J Appl Sci Technol Trends 2(1):20–28 [Internet] [cited 2023 Dec 3]. Available from: <https://jastt.org/index.php/jasttpath/article/view/65>. Gupta P (2021) Decision trees in machine learning [Internet]. Towards Data Science (2017) [cited 2023 Dec 3]. Available from: <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>

Enhancing Crop Yield Through Rainfall Analysis Using Machine Learning Techniques



Spandana N. Raj, K. Amuthabala, and Shantala Devi Patil

Abstract Rainfall prediction and crop yield estimation are critical agricultural planning and management factors. Climate change has an increasing impact on agriculture's function in developing nations like India and many crops. Although there are several methods or techniques for calculating agricultural production, their precision is not up to pace. The literature study indicates that there are no appropriate treatments or technological advancements for the aforementioned issue. The over-mechanization of agriculture was being pushed by this technique. The use of machine learning techniques has demonstrated encouraging results in recent years in improving the accuracy of these forecasts. In order to forecast rainfall and estimate agricultural productivity, this study evaluates the efficacy of two machine learning algorithms. The implementation of the ridge classification technique and the K-Nearest Neighbors' (KNN) classification algorithm reflects the complex relationship between rainfall patterns and agricultural yields. To improve the performance of the models, several feature engineering and data pretreatment techniques are used.

Keywords KNN classification · Ridge classification · Machine learning algorithms · Prediction

S. N. Raj (✉) · K. Amuthabala · S. D. Patil
School of Computer Science Engineering, Reva University, Bengaluru, India
e-mail: spandanaraj634@gmail.com

K. Amuthabala
e-mail: amuthabala@gmail.com

S. D. Patil
e-mail: shantaladevi.patil@reva.edu.in

1 Introduction

The agriculture sector is now very concerned about climate change and its effects on weather patterns. The success of agricultural practices significantly depends on the calculation of crop yield and the prediction of rainfall. Making forecasts based on previous trends and handling complex environmental data have shown machine learning algorithms to be useful tools. In this research, we would like to use the K-Nearest Neighbors (KNNs) and Ridge Classifier strong machine learning algorithms to forecast rainfall and crop yield, respectively. Farmers and stakeholders will gain useful information from the integration of these algorithms that will help them improve productivity and improve agricultural practices.

K-Nearest Neighbors (KNNs) are a quick and effective machine learning method for classification and regression issues. KNN can analyze previous rainfall data to provide accurate predictions about future rainfall patterns in the field of rainfall prediction. KNN calculates the closest neighbors and predicts the rainfall for a particular area and time period by finding similar past weather conditions.

A common regularization method for classification jobs is the Ridge Classifier. To anticipate crop yields, we will use the Ridge Classifier algorithm to examine historical crop yield data and climatic variables, such as rainfall and crop type. This model will allow farmers to forecast crop yields for various crops, helping them select the best crops, allocate resources efficiently, and create comprehensive farming strategies.

The agricultural community will greatly benefit from the execution of the rainfall predicted and crop prediction projects using the KNN and Ridge Classifier algorithms. Accurate predictions of rainfall will help farmers plan ahead for irrigation, fertilization, and insect management, improving the health and yield of their crops. Crop yield predictions will also help with waste reduction, resource allocation, and overall agricultural sustainability.

2 Literature Review

The following articles are relevant to the estimation of the literature on rainfall predicted for crop yield improvement.

The author [1] investigates the creation of a decision support system for crop yield optimization through balanced nutrient management. The importance of nutrient balance in farming is discussed, various decision support systems for crop management are shown, and the potential of technological solutions for sustainable agriculture is highlighted.

The author [2] suggests a better computational model that successfully integrates a higher level of quality assurance which can be produced by using an integrated optimization model that addresses all quality degradation issues. The suggested model for distributed data warehouse quality assurance is developed using an analytical methodology.

The article [3] focuses on predicting crop yield using machine learning methods, through the use of numerous algorithms and past data. Incorporating weather data and pertinent information, the study demonstrates how machine learning improves precise crop yield estimates. For the benefit of farmers and others, this facilitates resource allocation, crop planning, and sustainability.

The author [4] uses random forest to predict crop yields by analyzing historical yield data, meteorological factors, and other relevant features. The study demonstrates the effectiveness of this technique in forecasting agricultural productivity, highlighting its significance for optimizing farming planning and resource allocation. This aids farmers and decision-makers in enhancing crop production and overall sustainability.

The article [5] speaks about comparing GAN models for NLP, picture generation, and translation, as well as comparing GAN models based on activation, optimization, and loss functions. This chapter also lists the numerous picture and NLP datasets that are accessible for study.

The article [6] focuses on wheat crop yield prediction through machine learning. By analyzing historical yield records, meteorological data, and relevant variables, it explores various machine learning algorithms for accurate forecasts. The study highlights these algorithms' effectiveness in predicting wheat crop productivity and emphasizes their role in optimizing resource allocation and agricultural sustainability. This aids farmers and stakeholders in making informed decisions for enhanced wheat cultivation.

The article [7] study uses information on soil moisture and machine learning to forecasts crop yield. The study investigates various forecasting algorithms that incorporate soil moisture and pertinent variables to predict crop productivity. The study emphasizes how accurate these forecasts were by measuring model performance. For better irrigation, resource management, and general agricultural results, this has practical ramifications.

The author [8] focuses on the use of Sentinel-2 satellite data to extract pertinent agricultural properties and applies machine learning algorithms for yield prediction. The study shows how satellite data may be used to precisely estimate the productivity of monocot crops by evaluating the models' performance. In order to help farmers and other stakeholders maximize crop management practices, resource allocation, and total crop yield enhancement, the study emphasizes the significance of this method in precision agriculture.

The paper [9] addresses precise crop prediction by integrating moisture and water considerations through machine learning. It explores incorporating moisture-related parameters into machine learning algorithms to enhance yield anticipation. The study underscores the effectiveness of this approach in predicting agricultural outcomes, stressing its value for sustainable farming. It provides valuable insights for optimizing water management, resource allocation, and overall crop yield, benefiting farmers and decision-makers.

The article [10] provides a machine learning method for forecasting agricultural production and recommending the most effective fertilizer application. The study illustrates the value of this approach in delivering insightful information to improve

crop productivity through precise fertilizer application by analyzing crop and soil data. Putting an emphasis on resource efficiency and sustainable farming methods supports precision agriculture.

This study [11] identifies the soil fertility element and, consequently, the appropriate crop to cultivate for the highest yield. This encouraged the development of software that determines the soil's amounts of nitrogen (N), phosphorous (P), and potassium (K) by first evaluating a variety of soil fertility properties like moisture, electrical conductivity, pH, and others. Using machine learning and the gradient descent optimization method for algorithmic classification, the data collection is reviewed and the best crop-to-plant ratio is projected.

3 Proposed Methodology

The K-Nearest Neighbors' classification and ridge classification are the techniques that are frequently used to create machine learning applications.

3.1 *K-Nearest Neighbors' Classification*

KNN is a technique for supervised learning that is mainly used for solving classification-related issues. A new data point is given a class label based on the majority class among its k closest neighbors in feature space using this basic classification technique. The algorithm determines the separations between data points, chooses the k closest neighbors, and then, using a majority vote determines the new point's class. KNN can be sensitive to the selection of k and distance measures and is intuitive and non-parametric (Fig. 1).

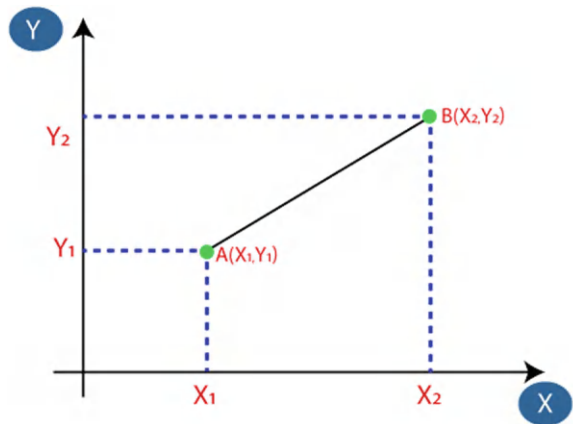
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

This equation calculates the Euclidean distance (d) between two points. Where, the coordinates of one point are represented by x_1 and y_1 , and those of another point by x_2 and y_2 in a two-dimensional space.

3.2 *Ridge Classification*

It is the most important concept of machine learning. It is a technique for avoiding overfitting the model by giving it extra data. The machine learning model may perform well on training data but poorly on test data on occasionally. The model is said to be overfitted when it can't predict the outcome while dealing with unknown

Fig. 1 K-Nearest Neighbor's classification algorithm



data by adding noise to the output. A regularization strategy can be used to address this problem.

3.3 Random Forest Regression

The supervised learning procedure makes use of the well-known machine learning algorithm random forest. It can be used to address problems involving regression and classification. It is built on the idea of ensemble learning, a method for incorporating various classifiers to solve challenging issues and enhance model performance.

3.4 Confusion Matrix

A confusion matrix provides a tabular format of the possible outcomes of predictions and results, that helps in visualizing the impacts of a classification task. It creates a table with all of the classifier's predicted and actual values (Fig. 2).

Some formulas are used to find the classification report they are:

Fig. 2 Table representing confusion matrix

		Actual Value	
Predicted value		Positive	Negative
	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F - score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

where

RSS represents the sum of squares of residuals.

TSS represents the total sum of squares.

Figure 3 explains the following steps:

- **Data Collection:** Gathering raw data from various sources.
- **Data preprocessing:** The process of preparing raw data for analysis by cleaning, converting, and organizing it. This involves activities such as dealing with missing numbers, outliers, and normalizing data.
- **Data Analysis:** Once the data has been preprocessed, you analyze it to gain insights, identify patterns, and draw conclusions. This is where you explore the data's characteristics and relationships.
- **Data Mining:** This is a subset of data analysis that involves searching for patterns or relationships within the data using techniques like clustering, association rules, classification, regression, etc. Data mining often helps to discover valuable information and knowledge hidden within the data.
- **Data Splitting:** After data analysis or data mining, you might split your dataset into training, validation, and testing sets for building and evaluating models.
- **Result Analysis:** Analyze the outcomes of your models, whether they are predictive models or other types of analyses.
- **Predict the Result:** Using the models you've built to make predictions on new or unseen data.

The primary objectives of the paper are listed as follows:

- Develop accurate machine learning models for predicting rainfall patterns.
- Create robust algorithms to estimate crop yields based on diverse factors.
- Empower small farmers with timely and precise data-driven insights.
- Enable informed decision-making, resource optimization, and risk mitigation.
- Enhance agricultural productivity and sustainability in changing conditions.

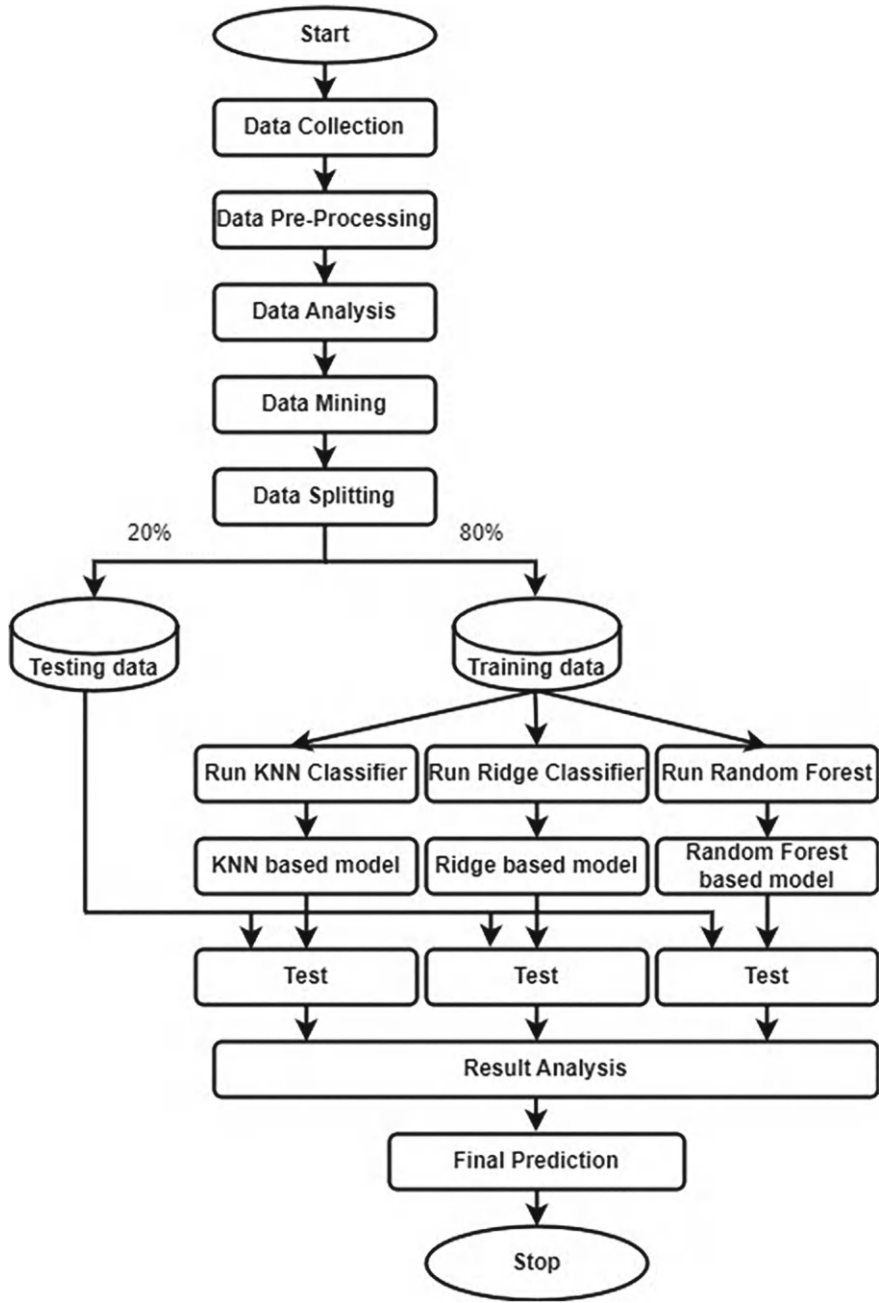


Fig. 3 Flow chart

Table 1 Dataset chosen for the proposed work

	State	Season	Crop	Area	Rainfall
0	Andaman and Nicobar Islands	Kharif	Arecanut	1254.0	2763.2
1	Andaman and Nicobar Islands	Kharif	Other Kharif pulses	2.0	2763.2
2	Andaman and Nicobar Islands	Kharif	Rice	102.0	2763.2
3	Andaman and Nicobar Islands	Whole year	Banana	176.0	2763.2
4	Andaman and Nicobar Islands	Whole year	Cashew nut	720.0	2763.2

The primary goal of the analysis is to automatically test and forecast the best crop for a specific rainfall, as stated and illustrated above. The paper simplifies and improves the traditional farming process.

3.5 Dataset

Table 1 represents the list of dataset attributes including State, Season, Crop, Area, and Rainfall.

4 Result

The proposed study uses the Python framework to predict rainfall and crop yield which helps farmers in improving crop yield. The necessary dataset has 74,976 rows and 7 columns with the variables such as state, year, season, crop, area, production, and rainfall. This work enhances the rainfall and crop yield in various states (Figs. 4, 5, 6, 7 and 8).

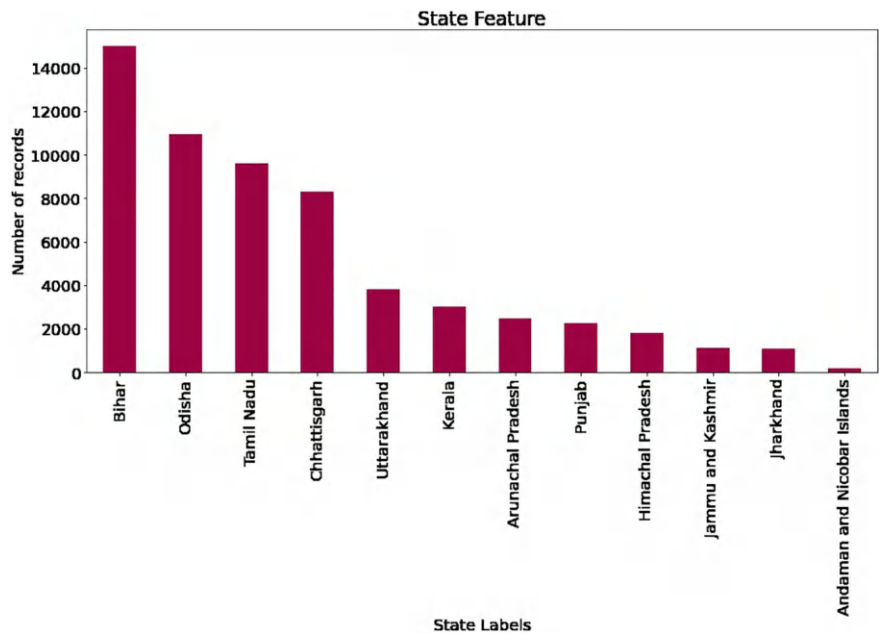


Fig. 4 Vertical bar graph represents state features with the number of records over different states

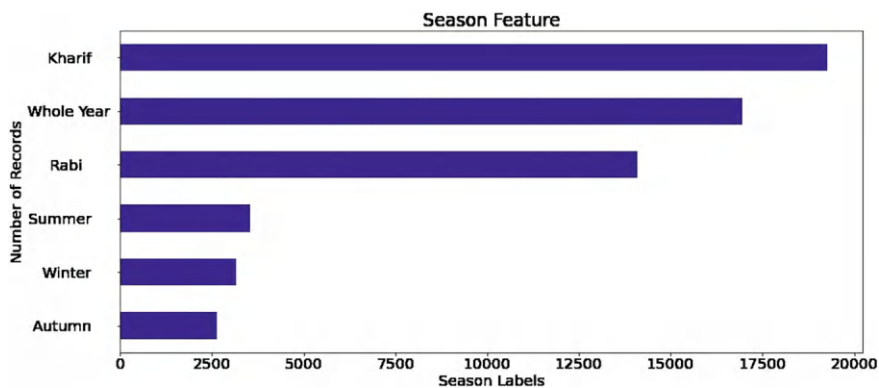


Fig. 5 Horizontal bar graph representing the season features with the number of records over different seasons

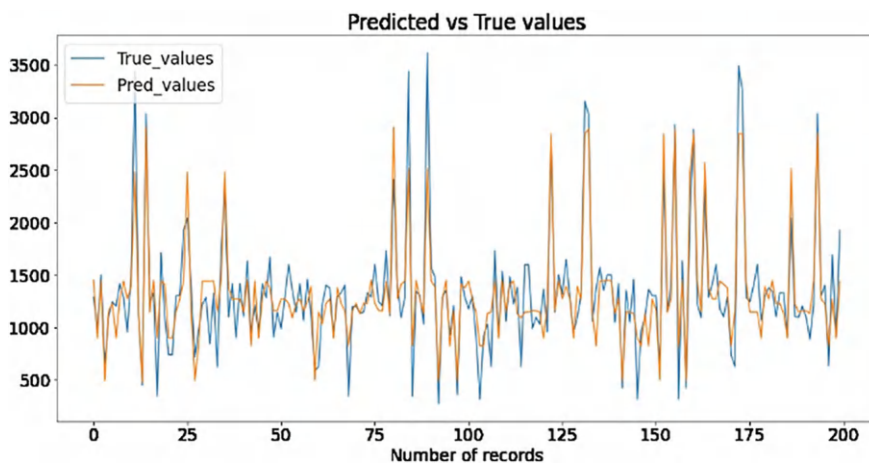


Fig. 8 Graph shows a comparison between the original value and the prediction

5 Conclusion

In this study, we analyze agriculturally acceptable rainfall in order to forecast a crop that can be farmed profitably and productively based on precipitation. The research will be carried out using a system based on machine learning built on the intention of combining the latest technological and computational advances into this paper. In summary, our project focused on leveraging machine learning techniques to predict rainfall and enhance crop yield. We compared the effectiveness of K-Nearest Neighbors (KNNs) and Ridge Classifier, converting unsupervised data to supervised using K-means clustering. Through the confusion matrix, we evaluated classifier performance. Additionally, we employed random forest regression, achieving accurate rainfall prediction validated by the R-squared (R^2) score. Our work offers a valuable pathway for optimizing agriculture in a changing climate, blending data insights and algorithms to ensure sustainable crop production. A better knowledge of rainfall will enable the farmer to plant the appropriate crop at the appropriate time of year to increase the yield of the crop.

Acknowledgements The authors wish to express their gratitude to REVA University for providing the facilities to carry out the research.

References

1. Sanjana V, Mundada MR (2019) Decision support system to agronomically optimize crop yield based on nitrogen and phosphorus. In: 2019 4th International conference on computational systems and information technology for sustainable solution (CSITSS). IEEE

2. Amuthabala P, Santhosh R (2019) Robust analysis and optimization of a novel efficient quality assurance model in data warehousing. *Comput Electr Eng* 74:233–244
3. Ranjani J et al (2021) Crop yield prediction using machine learning algorithm. In: 2021 4th International conference on computing and communications technologies (ICCCCT). IEEE
4. Suresh N et al (2021) Crop yield prediction using random forest algorithm. In: 2021 7th International conference on advanced computing and communication systems (ICACCS), vol 1. IEEE
5. Thirumagal E, Saruladha K (2021) GAN models in natural language processing and image translation. In: *Generative adversarial networks for image-to-image translation*. Academic Press, pp 17–57
6. Kumar D et al (2022) Wheat crop yield prediction using machine learning. In: 2022 International conference on data analytics for business and industry (ICDABI). IEEE
7. Paudel S et al (2022) Prediction of crop yield based-on soil moisture using machine learning algorithms. In: 2022 2nd International conference on technological advancements in computational sciences (ICTACS). IEEE
8. Rananavare LB, Chitnis S (2022) Monocot crop yield prediction using Sentinel-2 satellite data. In: 2022 IEEE 3rd global conference for advancement in technology (GCAT). IEEE
9. Shreyas S, Bhat S, Prasad SH (2022) Effective crop prediction considering water and moisture needs using machine learning techniques. In: 2022 IEEE North Karnataka subsection flagship international conference (NKCon). IEEE
10. Jeevaganesh R, Harish D, Priya B (2022) A machine learning-based approach for crop yield prediction and fertilizer recommendation. In: 2022 6th International conference on trends in electronics and informatics (ICOEI). IEEE
11. Prathiksha SV, Raj SN, Naveen S, Ashwinkumar UM (2023) Crop yield prediction of cotton using optimization technique. In: 2023 International conference on smart systems for applications in electrical sciences (ICSSSES). Tumakuru, India, pp 1–6. <https://doi.org/10.1109/ICSSSES58299.2023.10200233>

Performance Evaluation of Machine Learning Algorithms for Breast Cancer Detection in Women



B. Chaitrashree, A. Manasa, Shridhar B. Devamane, Reshma D'Souza, S. Prerana, and K. Pooja

Abstract Cancer is sorted by widespread and atypical development of cells. This unrestricted maturation of cells donates to the tumor evolution. Each year, the incidence of breast cancer is escalating significantly making it a crucial disease amid women widespread creating it a remarkable common health issue in today's mankind. Consequently, advancements in the forecasting and the detection of cancer are crucial for promoting a wholesome well-being. Therefore, achieving a fine-grained accuracy in cancer fidelity is requisite for strengthening treatment interventions and the overall viability of patients. Algorithms are employed to train a model on the dataset of patients. Feature selection is performed, and cancer stage is predicted using selected features. Owing to breast cancer, numerous women pass away worldwide. The recommended model incorporates multiple Machine Learning (ML) methods, including Logistic Regression (LR), Random Forest Classifier (RF), Artificial Neural Network (ANN), XG Boost Algorithm.

Keywords Artificial Neural Network · XG -Boost classifier · Logistic Regression · Machine learning · Random Forest Classifier

1 Introduction

Tumors can be either malignant or benign, with the former spreading and invading supplementary fragments of the body. The cancer phase designates the coverage of expansion and dissemination, encompassing non-invasive, incipient phase and

B. Chaitrashree (✉) · A. Manasa · S. B. Devamane · R. D'Souza · S. Prerana · K. Pooja
Department of Computer Science and Engineering, Global Academy of Technology, Bengaluru, India
e-mail: chaitrashreebasavaraj@gmail.com

S. B. Devamane
e-mail: shridhar.devamane@gat.ac.in

R. D'Souza
e-mail: reshmadsouza@gat.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025
N. R. Shetty et al. (eds.), *Advances in Artificial Intelligence and Machine Learning*,
Lecture Notes in Electrical Engineering 1335,
https://doi.org/10.1007/978-981-96-3105-6_4

progressed infiltrative breast cancer. Breast cancer prompts as the quick proliferation of typical breast particles preminent to the maturation of the irregular and aberrant cells and is accountable for the establishment of a tumor in the breast tissue signaling the occurrence of cancer. Regardless the reality that breast cancer scatters to adjacent lymph nodes, it has the potential to metastasize to the other areas of the body such as the bones, liver, lungs, or brain provoking in stage IV or metastatic breast cancer. Reverting after the primary therapy, it might again reoccur in the same breast or in the nearby lymph nodes which is pointed out to as local or regional recurrence. Nevertheless, if it returns in other regions of the body, it is considered as distant and metastatic reoccur. If the tumor initiates to promptly percolate to other units of the body, the disease probably would be irremediable when figured out. The timely determination of the ailment would be supreme for the forecasting process. The indication may be fragile during the starting point, but it would be mandatory to ascertain the disease shortly because the likelihood of persistence fiercely rises if spotted priorly. Machine learning and deep learning techniques can assist physicians in developing diagnostic verdict. The suggested system for forecasting breast cancer consists of traditional ml approach. We have used three models, namely Logistic Regression, Random Forest Classifier, and XG Boost Algorithm, and a deep learning procedure, Artificial Neural Network is employed to train and assess the dataset. The phases of the suggested system encompass the feature-selection process, database partitioning, model optimization and training, and model evaluation. The data sample encompasses 30 properties and one class tag. These metrics offer insights into the characteristics of cell nuclei as observed in the breast image snippet, with a binary class label: 0 indicating a benign condition and 1 denoting malignancy, thus enabling the classification of breast cancer as either benign or malignant. The key strengths of utilizing feature-selection algorithms are that they enable us to discover the most critical features in a dataset. The data repository is subdivided into two constituents: training set and testing set. The training set has been fine-tuned and trained, and the solutions have been documented for each model. The models undergo examination using the testing dataset, and the results of the testing phase were documented for each individual model. The investigative analysis was carried out for the dataset and visualized through pair plot, count plot, bar plot, and heat map. In this, several studies conducted by researchers utilizing diverse algorithms regarding breast cancer identification are outlined. We have used multitudinous techniques and discovered Logistic Regression showed highest accuracy whereas the Artificial Neural Network showed least accuracy.

2 Related Work

Arpita Joshi and Ashish Mehta [1] performed a provisional survey of the classification outcomes generated by different techniques on the cancer dataset which was available on UCI repository involving 699 samples. They employed a range of methods to achieve the highest possible accuracy in early-stage breast cancer prediction. The

result demonstrated the KNN was the most proficient technique compared to SVM, Random forest and decision tree algorithms.

Anji Reddy Vaka, Badal Soni, K. Sudheer Reddy [2] proposed a strategy which showed how the tumor cells multiplied. They took dataset which was prepared by specialized doctors and gained various insights about the dataset. Early detection of the breast cancer is possible which would control the death rate caused by this disease.

Monica Tiwari, Rashi Bharuka, Praditi Shah, and Reena Lokare [3] proposed a modality which showed how to undertake the data present in Kaggle. They have used deep learning techniques in addition to machine learning methods and differentiated the reliability between them. The study demonstrated that the accuracy obtained using CNN was 97.3% which gave positive outcome when contrasted with machine learning algorithms.

Anastraj, T. Chakravarthy, K. Sriram [4] conducted an appraisal in juxtaposition of assorted predictive modeling algorithms including backpropagation method, Artificial Neural Network conducted a relative assessment of myriad machine learning algorithms inclusive of ANN and SVM, on the original dataset. They utilized a profound convolutional neural network in conjunction with the ALEXNET architecture, facilitating the extraction of distinctive features while scrutinizing both benign and malignant tumors. The emulation results illustrated that SVM was the most efficient method leading to better outcomes (94%).

J. Sivapriya, V. Aravind Kumar, S. Siddharth Sai, and S. Sriram [5] conducted an analysis on the dataset. They have conducted research and represented various patterns in the data by using data visualization. They have used matplotlib and seaborn libraries for visualizing the data. They concluded that Random Forest Classifier was the best typifier when scrutinized to other algorithms.

Mohammed Amine Naji, Sanaa El Filali Kawtar Aarika, EL Habib Benlahmar, Rachida Ait Abdelouhahid, Olivier Debauche [6] conducted breast cancer prediction by employing five distinct algorithms, including Random Forest, Logistic Regression, Support Vector Machine, Decision Tree and K-Nearest Neighbors. Evaluated their effectiveness using metrics like the confusion matrix, accuracy, and precision, achieving an accuracy rate of approximately 97.2%, aligning with our model's predicted accuracy.

Md. Milon Islam, Md. Rezwanul Haque, Hasib Iqbal, Md. Munirul Hasan, Mahmudul Hasan, and Muhammad Nomani Kabir [7] completed an investigation using a breast cancer dataset, employing five distinct supervised machine learning approaches, focusing on metrics such as accuracy, sensitivity, specificity, precision, negative predictive value, and the false-negative rate. We found that the accuracy difference compared between the work we did and the work carried out by them was high.

Mohammad Monirujjaman Khan, Somayea Islam, Srobani Sarkar, Fozayel Ibn Ayaz, Md. Mursalin Kabir, Tahia Tazin, Amani Abdulrahman Albraikan, and Faris A. Almalki [8] used computer-aided detection and diagnosis technologies for the timely screening of the disease which facilitated longevity of humankind. Their work showed highest correctness in Logistic Regression algorithm which was 98%.

Nikita Rane, Rucha Kanade, Jean Sunny, Prof. Sulochana Devi [9] inculcated diverse strategies like ensemble methods, agglutination analysis, and data mining algorithms. They compared six machine learning algorithms by partitioning the dataset to testing phase and training phase and used SQLAlchemy toolkit which is an object relational mapper and obtained highest accuracy in SVM algorithm.

Chang Ming, Valeria Viassolo, Nicole Probst-Hensch, Pierre O. Chappuis, Ivo D. Dinov, and Maria C. Katapodi [10] have worked on ML-based algorithms for estimation against the standard practices such as, utilized both the breast cancer risk assessment tool and the breast and ovarian analysis of disease incidence and carrier estimation algorithm models, achieving an accuracy level of approximately 88%.

3 Algorithms Description

The algorithms considered for the implementation are discussed below.

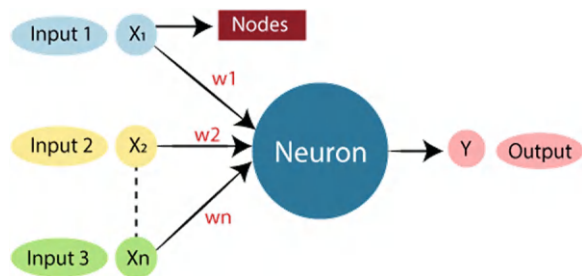
3.1 Artificial Neural Network

Artificial Neural Network is a machine learning algorithm whose computational model which consists of interconnected units, or synthetic neurons, arranged in tiers. Here the information is processed through the network in which each neuron receives the inputs, processes them, and passes the results to the next layer. ANN algorithm will be able to recognize the patterns, through which it will be able to make the predictions and solve the complex problem which process is called training. The training procedure encompasses fine-tuning the synaptic strengths of neuron-to-neuron connections in response to the training data, providing the network to normalize the new and unseen data to make accurate predictions (Fig. 1).

Artificial Neural Network includes three tiers among these components are Input Layer, Hidden Layer, Output Layer.

- **Input Layer:** It apprehends the input from the programmer in several ways from the outside world.

Fig. 1 Exemplar of an Artificial Neural Network assumes the visual representation depicted above



- **Hidden Layer:** It is the stratum positioned amidst the input and the output layers, which is used to perform various computations to find the obscured features and patterns.
- **Output Layer:** It is the layer, when the input undergoes through a sequence of modifications inside the hidden layer is outputted.

$$\sum_{i=1}^n W_i * X_i + b.$$

The transfer function can be represented using the above formula which determines the sum of the inputs and includes a bias.

3.2 *Logistic Regression*

Logistic Regression is the statistical framework that foretells the likelihood of an event unfolding, hinging on the independent variables values incorporated in the examination. It is a kind of regression analysis which takes dependent variables as the categorical variables which takes only two possible values which are '0' and '1'. The logistic function is renowned for its alias, the 'Sigmoid function,' employed to convert the linear combination of independent variables into a probability value, gracefully confined within the range of 0 to 1.

AWS has defined Logistic Regression as the important regression analysis in the field of machine learning, as the models using Logistic Regression analysis help organizations to gain actionable insights from their business data. Here we have relied highly on the Logistic Regression for the proper presentation of data, and to make use of the model in more powerful way, we have selected the important features from the available dataset using the backward elimination and recursive elimination techniques. Logistic Regression is also contemplated as a discriminative model which methods to strive between the classes or the categories in addition, Logistic Regression maximizes the log likelihood function to regulate the beta coefficients of the model which substitutes slightly under the circumstances of machine learning. Training phase duration in Logistic Regression focuses to find the optimal values for the weights which maximizes the likelihood of the observed data which is usually done using the optimization algorithms encompassing techniques like the descent of gradients. When the model is honed, anticipations may be derived by determining the likelihood of the affirmative category for fresh input data and setting a demarcation point to allocate the data to either of the dual categories. Logistic Regression is having various alternatives such as multiclass Logistic Regression for controlling the problems with more than two classes, and normalized Logistic Regression assists to alleviate overfitting by containing regularization terms into the objective function.

3.3 *XG -Boost Algorithm*

The XG Boost Algorithm stands as a cutting-edge paradigm within the realm of machine learning, exhibiting extensive versatility in handling tasks encompassing both regression and classification domains. It is renowned for its excellent anticipating accuracy and elasticity, making it favored among the researchers, data scientists, and machine learning practitioners. It is “extreme Gradient Boosting” algorithm which is the extension of the gradient boosting method, which amalgamates an ensemble of weak prediction models which are typically of decision trees into a stronger predictive model. XG Boost initiates several improvements over traditional gradient boosting algorithms, leading to enhanced performance and efficiency of the model. It has become well-known due to its capability to control the large datasets and structured handling of missing values without preprocessing the data values significantly. Besides XG Boost has integral assist for parallel processing, making it enable to train models on large datasets in an appropriate amount of time. XG Boost is used in diversity of applications, which includes the recommendation systems, and it is also highly personalized which allows for calibration of various model parameters to optimize the performance and increase the efficiency. XG Boost is implemented through Gradient Boosted decision trees, in which the decision trees are generated in consecutive forms. XG Boost can be regarded as the ideal combination of the software and hardware optimization approach to yield condensing results using minimum computing resources in lesser amount of time. It is an enhanced application of gradient boosting that incorporates the robustness of both the gradient boosting and the integrated decision trees. Certain key characteristics and the advantages include tree pruning which removes the unnecessary branches from the decision trees which helps to reduce the complications involved in the final model and improves its regulation.

3.4 *Random Forest Classifier*

The Random Forest Classifier stands as a formidable entity within the domain of machine learning, known for its prowess in weaving complex data into predictive insights which comes under the type of supervised learning technique and belonging to the illustrious ensemble learning lineage, the Random Forest Classifier embodies the ingenious concept of amalgamating numerous decision trees to tackle intricate problems and enhance model performance. Widely cherished, it finds its dominion in both classification and regression analyses. The Random Forest Classifier’s essence lies in constructing a coalition of decision trees, each sculpted from a randomly selected segment of the training data. Notably, at each branch of the decision tree, a fortuitous subset of attributes assumes the role of potential splitters, introducing a sprinkle of unpredictability that fosters diversity and fends off the peril of overfitting.

Some of the advantages of the Random Forest Classifier includes robustness to overfitting which helps in the improvement of the generalization performance, versatility which handles categorical and continuous features without the much requirement of the extensive data preprocessing. It can also construct a logical forecasting without hyper-parameter tuning. Random Forest Classifier is used in considerable ways, as it lowers the matter of overfitting in decision trees.

4 Experimental Setup

The experiments were done using Google Colab, which is a web-based collaborative coding platform model centered around Jupyter Notebook, facilitating the execution of python notebooks. The dataset was secured from kaggle. The data gathering is being the initial step followed by data wrangling and omission of vacant entries, formulating test and training samples from the acquired data. Thirty features have been used for the evaluation. Each cell nucleus contributes ten distinct numerical attributes, encompassing calculations for mean, standard deviation, and peak values, thereby culminating in an intricate collection of 30 distinctive properties for the image. The dataset consists of 569 observations and 30 fields that possess all entities. Here we consider two target variables, namely malignant and benign, and we retrieve the information of cancer dataframe, finding that the data is clean and no missing values or null values. In order to homogenize the magnitudes of all features, we've contemplated the adoption of the StandardScaler technique available within the Scikit-Learn Library (Fig. 2).

4.1 Dataset Description

This dataset subsumes aspects gleaned from digitized fine needle aspirate (FNA) images of breast masses. The inherent traits predominantly center around cellular nucleus attributes within a tridimensional realm, as elucidated by K. P. Bennett and O. L. Mangasarian in a 1992 treatise concerning resilient linear programming discrimination. The dataset is obtainable via the UW CS ftp server. Furthermore, it can also be accessed through the UCI machine learning repository. The dataset comprises an identification number, diagnosis (malignant or benign), and ten genuine-valued characteristics calculated for every cellular nucleus. These attributes encompass radius, texture, perimeter, area, smoothness, compactness, concavity, concave vertices, symmetry, and fractal dimension.

Every attribute figure is transcribed with a precision of four significant digits. There are no absent attribute values within the dataset. The distribution of categories in the dataset encompasses 357 cases classified as benign and 212 cases categorized as malignant, mirroring the essence of the diagnoses related to breast masses.

				cancer_df.isnull().sum()	
#	Column	Non-Null Count	Dtype	mean radius	0
0	mean radius	569 non-null	float64	mean texture	0
1	mean texture	569 non-null	float64	mean perimeter	0
2	mean perimeter	569 non-null	float64	mean area	0
3	mean area	569 non-null	float64	mean smoothness	0
4	mean smoothness	569 non-null	float64	mean compactness	0
5	mean compactness	569 non-null	float64	mean concavity	0
6	mean concavity	569 non-null	float64	mean concave points	0
7	mean concave points	569 non-null	float64	mean symmetry	0
8	mean symmetry	569 non-null	float64	mean fractal dimension	0
9	mean fractal dimension	569 non-null	float64	radius error	0
10	radius error	569 non-null	float64	texture error	0
11	texture error	569 non-null	float64	perimeter error	0
12	perimeter error	569 non-null	float64	area error	0
13	area error	569 non-null	float64	smoothness error	0
14	smoothness error	569 non-null	float64	compactness error	0
15	compactness error	569 non-null	float64	concavity error	0
16	concavity error	569 non-null	float64	concave points error	0
17	concave points error	569 non-null	float64	symmetry error	0
18	symmetry error	569 non-null	float64	fractal dimension error	0
19	fractal dimension error	569 non-null	float64	worst radius	0
20	worst radius	569 non-null	float64		

Fig. 2 Dataset with no missing values

The libraries imported are pandas for data modification or research, numpy for arithmetic computation, matplotlib and seaborn for data mapping, graphical analysis, and visual exploration. Scikit-learn package is used for data scrutiny. The model is built using the dataset and classification methodologies. Our approach revolves around a roster of techniques, featuring Logistic Regression, Random Forest Classifier, decision trees, XG Boost, Ada Boost Classifier, Support Vector Machines, and Artificial Neural Networks. The supreme-performing model undergoes rigorous optimization, ultimately serving as the oracle for determining the risk of breast cancer in females. This proactive assessment empowers timely interventions, potentially sparing them from critical conditions or even mortality.

As depicted in Fig. 3, we have tried to visualize the data using data visualization libraries which helps us to understand the dataset in better way. Within our toolkit, we’ve wielded the Seaborn pairplot visualization, a masterstroke for unveiling intricate correlations between dataset variables. This artistic rendering offers a panoramic view, orchestrating an eloquent data symphony that harmonizes and synthesizes voluminous information into a visual opus.

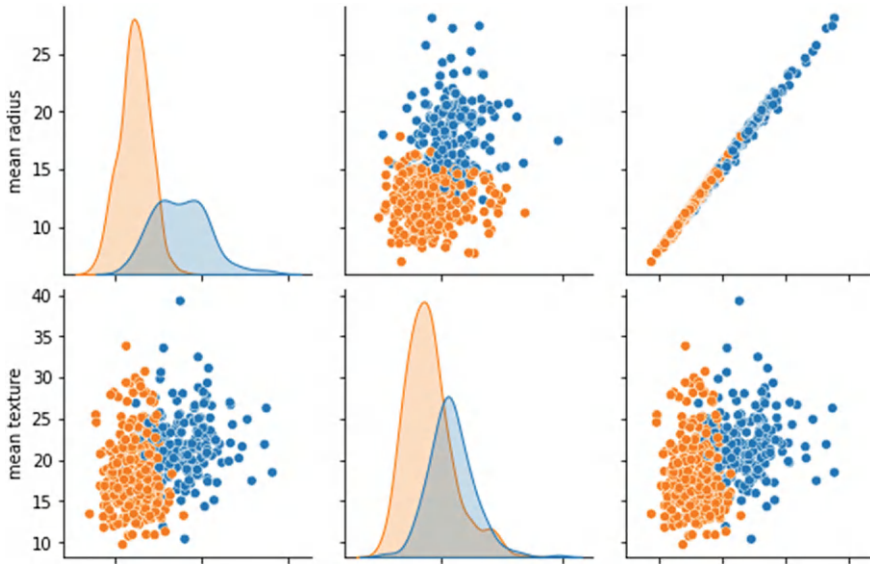


Fig. 3 Data visualization using pairplot

5 Results and Discussion

As previously indicated, we have carried out the experiments with algorithms as illustrated, Random Forest Classifier, XG Boost classifier, Logistic Regression and Artificial Neural Network. We have imported `accuracy_score`, `confusion_matrix` from `scikit-learn` library. In our notebook, we have endeavored a plethora of algorithms, but in our paper, we have predominantly talked about the four algorithms mainly Random Forest Classifier, XG Boost classifier, Logistic Regression and Artificial Neural Networks. For each algorithm, we comply with resembling procedure, i.e., we commence with importing the required modules. We instantiate the classifier by invoking the constructor. This initializes the constructor with default parameters. The next step would be to educate the classifier by invoking `fit()` function and providing the training data and their corresponding labels as inputs. This step enables the classifier to grasp patterns and correlations within the data. Once the classifier is honed, we set it to work, foretelling outcomes on the test data through the `predict()` function, storing these predictions in a designated variable. To gauge the classifier's efficacy, we scrutinize these predictions against the ground truth labels. This evaluation unfolds through the deployment of the `accuracy_score()` function from the `sklearn.metrics` module. The resulting accuracy score provides indication of the classifier's effectiveness in handling the test data.

Then the model is trained using the input data that has been normalized. Undergoing the process of standard scaling, which bears the moniker z-score normalization, entails a metamorphosis of the data, aligning it to possess a mean of zero and

a variance of one. The scaling technique ensures that all features or variables are on similar scale and mitigates the impact of outliers. By training a model with standard scaled data, it allows for more equitable comparisons between different features and prevents bias toward features with larger magnitudes. This approach can enhance the performance and convergence of certain machine learning algorithm, particularly those that are sensitive to the scale of input variables. In summary, training with standard scaled data entails training a machine learning model using input data that has undergone standard scaling to achieve consistent and comparable feature scales. Formula for accuracy, precision, and recall is stated below.

Accuracy = $\frac{TP + TN}{TP + TN + FP + FN}$

Precision = $\frac{TP}{TP + FP}$

Recall = $\frac{TP}{TP + FN}$

Quantifies the model’s precision by accounting for true positives, true negatives, false positives, and false negatives. The results of each of these algorithms are briefly summarized below. AG Boost classifier has got the highest accuracy of 98% among all the algorithms that we have experimented. The Random Forest Classifier attained the second-highest accuracy level, registering an impressive 97%. The Logistic Regression has got the accuracy of 96%. Artificial Neural Network has got quite less accuracy with 93%, and it concludes whether the tumor is malignant or benign (Table 1).

The countplot serves as an observational symphony, harmonizing the presence of data within categorical variables through a bar-chart-inspired visual narrative.

Here within the confines of the count plot visualization we have considered two classes which are the target variables considered in the starting of the section which are benign and malignant. We have considered the parameter of count for the x-axis and the target parameter for the y-axis which we refer for the depiction and then

Table 1 Discussion on results

Algorithms	Accuracy (%)
Support vector classifier	93
Logistic Regression	96
Naïve Bayes classifier	94
Decision tree classifier	94
AdaBoost classifier	94
Artificial Neural Networks	93
Random Forest Classifier	97

observe the pattern which is highlighted. The main observation observed in this visualization is comparing the categorical variables based on the countplot parameters (Fig. 4).

The count plot for feature mean radius is depicted in Fig. 5.

Data visualization is accomplished using heat map wherein around ten features are taken into consideration, and the value represents the correlation between the considered features in Fig. 6.

Fig. 4 Data visualization using count plot

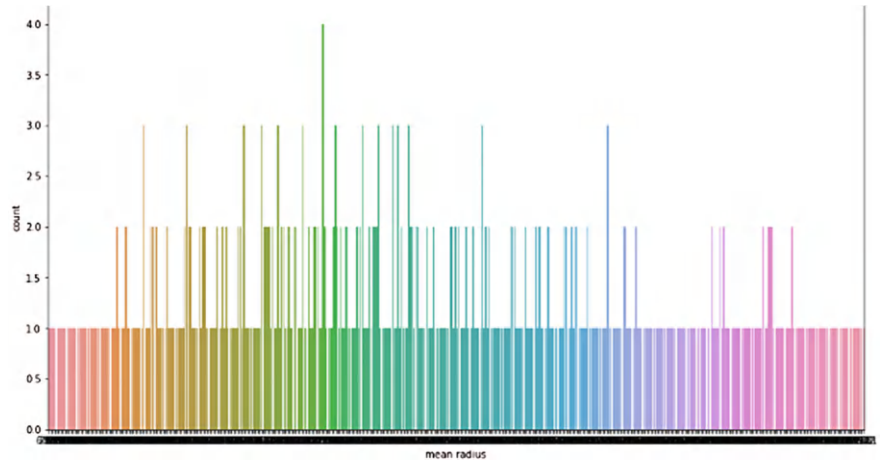
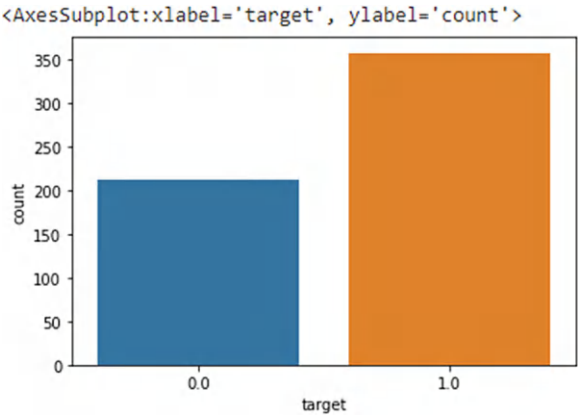


Fig. 5 Count plot for feature mean radius

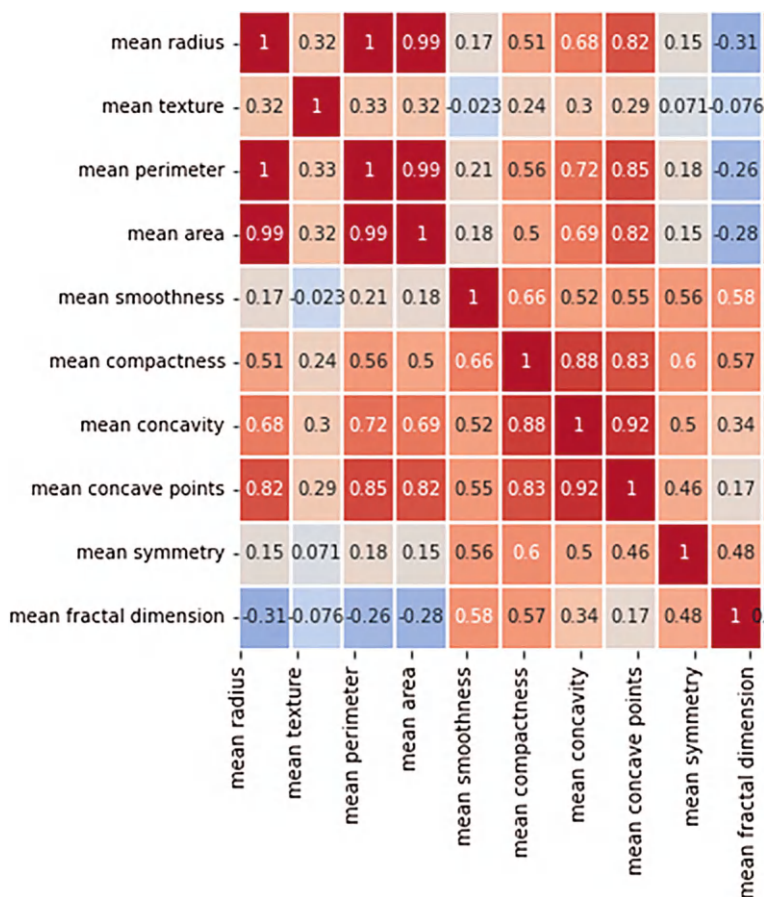


Fig. 6 Data visualization using heat map

6 Conclusion

The machine learning approaches proposed for the breast cancer prediction could predict the early detection of the disease which could help to slow down the progress of the disease and reduce the mortality rate through suitable therapeutic interventions at the correct time. The model has used several machine learning classification algorithms such as Logistic Regression, XG Boost Classifier, Random Forest Classifier, ANN, Decision Tree Classifier, Naïve Bayes, Ada Boost Classifier, and Support Vector Machines. The highest accuracy is observed in Random Forest Algorithm which is 97.36%. Even though our study pitches in the valuable insights into the study using machine learning, several limitations should be conceded. The dataset utilized is exhibiting an imbalance between benign and malignant cases, potentially influencing model performance. As inscribing class imbalance strategies, we have utilized

tree-based algorithms, which have shown the highest accuracy rate. Addition to this, the paucity of comprehensive clinical validation serves as a constraint, underscoring the imperative for heightened collaboration with healthcare practitioners to substantiate the real-world applicability of our findings. Future scope of this study involves integrating multi-modal data, such as genetic information and clinical histories which could lead to thorough exhaustive models. Additional amelioration avenues lie ahead as we fine-tune the model through adept algorithmic training, heightening our ability to forecast breast cancer in its nascent phases, and extending a guiding hand for swift intervention. This paper classifies the breast cancer whether it belongs to benign or malignant.

References

1. Muhammet Faith AK (2020) A comparative analysis of breast cancer detection and diagnosis using data visualization and machine learning applications
2. Rawat RM, Panchal S, Singh VK, Panchal Y (2020) Breast cancer detection using k-nearest neighbors, logistic regression and ensemble learning
3. Rawal R (2020) Breast cancer prediction using machine learning
4. Karthik S, Srinivasa Perumal R, Chandra Mouli PVSSR (2019) Breast cancer detection classification using deep neural networks
5. Gardezi SJS, Elazab A, Lei B, Wang T (2019) Breast cancer detection and diagnosis using mammographic data: systematic review
6. Sivapriya J, Aravind Kumar V, Siddarth Sai S, Sriram S (2019) Breast cancer prediction using machine learning
7. Anastraj K, Chakravarthy T, Sriram K (2019) Breast cancer detection either beningn or malignant tumor using deep convolutional neural network with machine learning techniques
8. Wadkar K, Pathak P, Wagh N (2019) Breast cancer detection using ANN network and performance analysis with SVM
9. Le H (2018) Using machine learning models for breast cancer detection
10. Joshi A, Mehta A (2017) Comparitive analysis of various machine learning techniques for diagnosis of breast cancer

Sarcasm Detection in Hindi Social Media Text Using HindiBERT-LSTM and XGBoost Models



Y. Indira Priyadarshini, K. Mahesh Babu, Y. Bhargavi, Y. Dharani, Y. Ramya, and P. Mani Chandana

Abstract Identifying sarcasm is challenging because it depends on various contextual and non-verbal information. To accurately understand the intended meaning, it's essential to recognize sarcasm in statements, particularly on open platforms like Facebook, Twitter, WhatsApp, etc., where people express themselves freely. Sarcasm detection in Hindi languages is challenging in natural language processing (NLP) due to the richness of annotated datasets and insufficient resources. Hindi, the fourth-most popular language in the world, remains unexplored in sarcasm detection. An automated system is necessary for sarcasm detectors in textual data, and considerable researchers have proposed several sarcasm detection algorithms. Since English is the most often used language in social networking groups, scholars use this language most to identify sarcasm in written content. There is a lack of research on sarcasm detection in various Asian languages, including Hindi, Telugu, Tamil, Urdu, and Bengali. In this paper, we suggested a model that uses ensemble machine learning (ML) and a BERT-based-LSTM model to detect sarcasm in Hindi text. Both models perform satisfactorily in terms of precision, recall, F-score, and accuracy, although the HindiBERT-LSTM (Long Term-Short Memory) model achieves higher performance overall. This model's accuracy is 96% on average.

Keywords Machine learning · Deep learning · Sarcasm · Natural language processing · HindiBERT · Social media

Y. Indira Priyadarshini (✉) · K. Mahesh Babu · Y. Bhargavi · Y. Dharani · Y. Ramya · P. Mani Chandana
Computer Science and Engineering, Ravindra College of Engineering
for Women, Kurnool, India
e-mail: indusai.yashu@gmail.com

Sarcasm is more difficult to detect than sentiment or other text categories in text data. Recently, much attention has been given to detecting sarcasm in written text, and some researchers have used text with emojis. Sarcasm detection has many uses in daily life, including figuring out the user's mindset and whether it is positive or negative, automating feedback systems, modifying business or marketing strategies based on customer responses, improving chatbot performance, and more [1]. Social media have transformed how we interact with each other [2], resulting in a lot of written communication that can be analyzed for valuable insights. Popular platforms like Facebook [3], Twitter [4], and WhatsApp [5] are widely used for communication and expressing opinions. This has led to the study of verbal expressions such as humor, irony, sarcasm, aggression, and hate becoming an important area of research. Twitter is a highly popular social media platform that has gained immense popularity worldwide [6–8]. It is a direct communication channel for social activists, politicians, and celebrities to reach the people. As more people share their thoughts on social media, sentiment analysis and opinion mining have become more important [9]. Detecting sarcasm in social media text is essential for fully comprehending sentiment and grasping the underlying meaning behind statements. There has been significant research on identifying sarcasm in English because ample resources are available for research in this field, with large annotated and carefully curated datasets readily available also. Some approaches have been concerned with specific languages, such as Chinese [10], Czech [11], Dutch [12], Indonesian [13]. Still, it is also crucial to analyze sarcasm in Hindi tweets because the language is widely used [14]. As a result, most researchers have chosen to focus on this language for their studies [15]. Hindi is the fourth-most spoken language in the world among low-resourced languages, with 490 million speakers worldwide, especially from India [16–18]. Lately, Hindi has gained popularity on social media platforms like India, with people frequently posting messages and comments. The rise of regional language used on social media presents a promising opportunity for information mining, and analyzing sarcasm in Hindi tweets is necessary for many Hindi speakers [15].

NLP researchers are working on using computational methods to detect these expressions automatically [19]. Using ML and NLP techniques, sarcasm recognition identifies a human's sarcastic feelings-based solely on their ideas and interpretation. Detecting sarcasm can be challenging due to various factors such as situational context, cultural background, and personal traits, and identifying them through text or speech [20], audio [21], or facial cues can be complex for humans. For machines, it is an even more complicated task. Nonetheless, many scholars have been employing models to automatically detect sarcasm from several types of inputs, including text, audio, video, and images [22].

We aim to contribute to the field of sarcasm detection work significantly. We are addressing the challenges and opportunities in this field and wish to promote more study and the creation of efficient methods, models, and datasets for precisely identifying sarcasm in Hindi tweets. The knowledge we have gathered can enhance NLP systems, leading to better sentiment analysis, opinion mining, and general comprehension of user-generated content in Hindi-speaking communities.

This paper incorporates recent research on detecting sarcasm in text corpus or documents. The primary contribution of our study:

- (1) We have compiled a dataset of Hindi text from online sources.
- (2) We developed a special HindiBERT-LSTM model and used HindiBERT as an embedding.
- (3) A comparison of the abilities of ML and DL to recognize sarcasm in Hindi text is presented.
- (4) Other significant contributions include Improving model performance and developing precise detection systems.

To achieve this, we have organized the article into several sections. Section 2 describes a thorough assessment of the literature, while Sect. 3 covers the methodology of our work. We have also included information about the performance of our model in Sect. 4. Lastly, we conclude our work.

1 Related Work

Sarcasm detection, a sentiment analysis component, is still challenging in NLP, which constantly improves. Sarcasm conveys one's emotions when discussing or commenting on a topic but serves a distinct purpose [8]. Sarcasm is a form of language that involves saying something but intending the opposite meaning, often through irony or mockery. Detecting sarcasm in the text is crucial, particularly on open platforms (such as Twitter, Facebook, WhatsApp), where individuals can express themselves, and it helps understand a text's true sentiment and meaning [23]. Identifying sarcasm in the text is becoming increasingly crucial for various industries, including security and commercial services. To contribute to this field, scholars [9] present the first dataset of tweets that are code-mixed between English and Hindi, annotated for the presence of sarcasm and irony. A baseline supervised classification system is also presented using this dataset, with an average F-score of 78.4 obtained by employing a random forest classifier and tenfold cross-validation. A study [24] analyzed Indian citizens' sentiments during the 2019 Lok Sabha election using Twitter data. Using Transfer Learning to handle the unsupervised sentiment analysis, they created an automatic tweet analyzer and utilized Linear Support Vector Classifiers and TF-IDF methodology for processing textual data. The model accurately predicted the BJP party's victory with a vote share difference of approximately 19%, which aligned closely with the election results. The study found that around 37.4% of the votes favored the BJP party, while about 19.5% favored the INC party.

Recent work [19] introduced a Twitter corpus for training unique word embeddings and a Hinglish dataset specifically tagged for sarcasm recognition. They propose a DL method employing bilingual word embeddings built from FastText, and Word2Vec approaches to address the problem of sarcasm identification in Hindi-English code-mixed tweets [25]. They experimented with deep learning models,

including CNNs, LSTMs [26], and Bi-directional LSTMs. The DL they developed performed better than any currently available models, and the attention-based Bi-directional LSTMs achieved the highest accuracy of 78.49%. Bharti et al. [15] suggested a context-based pattern for detecting sarcasm in Hindi tweets, which involves looking for contradictions between a tweet and the news context. The researchers used Hindi news articles from the same time as the tweet to achieve an accuracy rate of 87%. Their approach proved more effective than other state-of-the-art techniques for detecting sarcasm in Hindi tweets. Authors [25] proposed a pattern-based approach to detect sarcasm using Twitter data. Four feature sets with specific sarcasm patterns are presented to classify tweets as sarcastic or non-sarcastic. The proposed feature sets are evaluated for additional cost classifications using 9,104 tweets containing sarcasm and not dataset. Support Vector Machine (SVM), *K*-Nearest Neighbors (KNN), and Random Forest (RF) classifiers are used, with the Random Forest classifier obtaining the highest accuracy of 81% and an *F1* score of 79.00%. Although SVM can detect sarcasm with high accuracy and narrow down sentiment analysis, the actual results of all sarcastic tweets may vary.

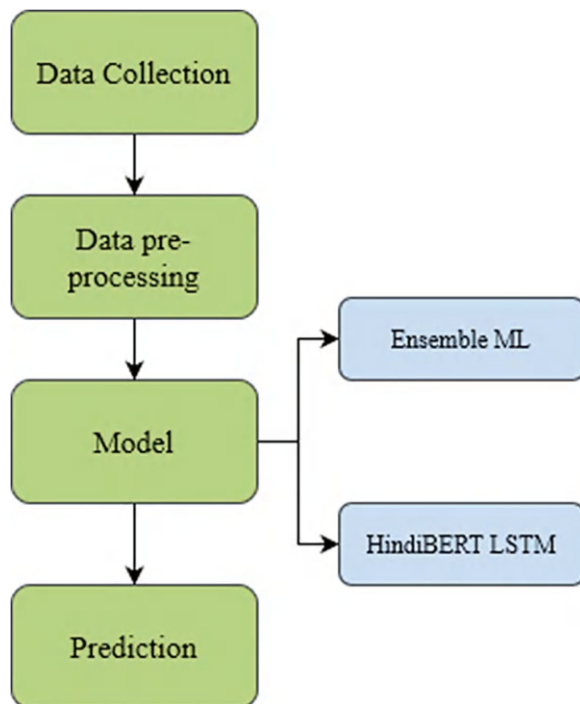
After reviewing the entire literature, we've discovered some limitations, challenges, and lackings:

- There has been limited research in Indian languages, such as Hindi, Tamil, and Bengali, particularly concerning detecting sarcasm in Hindi tweets [27]. This is due to insufficient annotated datasets and limited labeled data for training and testing, making it difficult to apply traditional methods [15]. More resources make it easier for researchers to process these tasks effectively [28, 29].
- Translating Hindi words into English is a challenging task that involves a two-step process. Firstly, the Hindi words are transliterated from English, and then the sentences are translated word-by-word into English [30].
- The dataset available for sarcasm detection needs to be more standardized. It needs to be sufficiently large and acceptable to effectively enhance models' performance in this field. It is worth noting that the amount of research conducted in the sarcasm detection domain needs to be increased [1].
- Identifying sarcasm had become more challenging in recent years, particularly since 2018, when this subject gained traction. There is a growing need for additional research and labeled data, especially for Hindi [31].

2 Methodology

For conducting the research, we followed a few procedures. Data collecting was one of the first tasks in the study process. The dataset was then preprocessed after that HindiBERT and combinations of TF-IDF and word n-gram are used to convert the feature vector from the raw data. Finally, we have developed our categorization model. Figure 1 depicts the steps in the research process. As follows are descriptions of each type:

Fig. 1 Steps we have taken in our research



- **Data Collection:** For this study, the dataset was employed from Kaggle, an open-source platform, and the raw dataset has over 16,000 tweets (including both sarcastic and non-sarcastic) to work on Sarcasm Detection in Hindi. The dataset contains 6,051 sarcastic tweets and 10,128 non-sarcastic tweets. The tweets were posted between January 1, 2012, and June 23, 2020

In addition, using web scraping techniques [32], we collected data on social media platforms like Facebook and Twitter and gathered 12,000 data, both sarcastic and non-sarcastic. Twitter provides an API to gather necessary data and retrieve data using this API, using specific keywords to make the data collection process easier for our research [33]. The collected data was then labeled through manual standards.

The raw dataset is a precious asset for researchers interested in detecting sarcasm in Hindi. The dataset is a foundation for creating and assessing effective models, algorithms, and techniques for sarcasm detection in Hindi tweets. The presence of this dataset inspires progress in sarcasm detection and sentiment analysis in the Hindi language. Figure 2 displays the total number of each class in the dataset. Figure 3 displays the data scraping procedure.

- **Data preprocessing:** In NLP, text preparation techniques are vital in preparing data for analysis and prediction. Data mining extracts patterns from a large number of datasets. Data preprocessing is the most important part of data

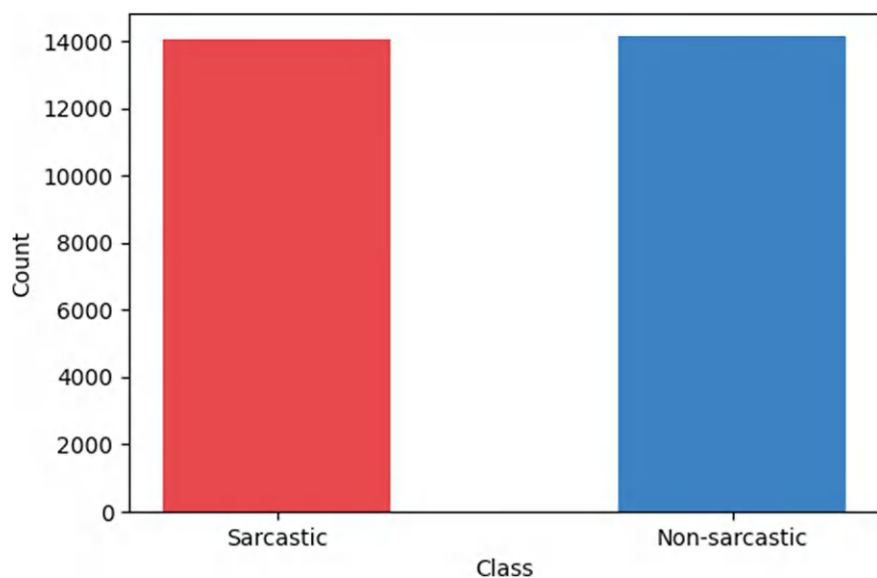


Fig. 2 Total counts of sarcastic and non-sarcastic instances

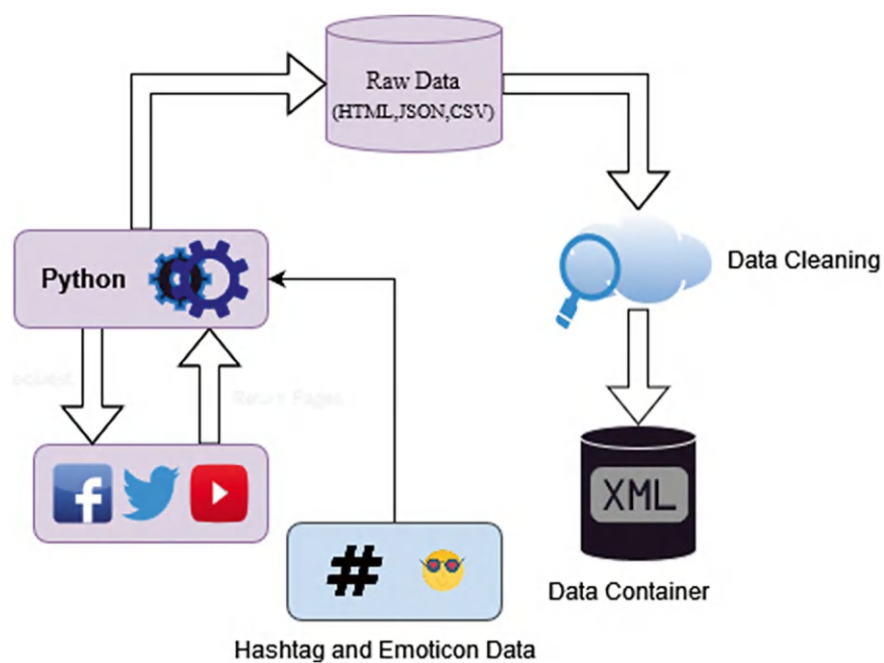


Fig. 3 Data collection using scraping

mining, where data is organized and transformed [34]. After collecting the raw data, the preprocessing technique has been implemented. These techniques involve cleaning, transforming, lowercasing, and standardizing to prepare the text data for modeling purposes and analysis.

Here are some commonly used text preparation techniques include:

- (1) **Cleaning:** To ensure the accuracy of NLP algorithms, it is crucial to clean the text data by removing any noise, irrelevant information, and inconsistencies. To get only text-based information, it is imperative to eliminate any HTML tags, hashtags, and URLs, also convert all textual data to lowercase, remove punctuation marks (such as extra dots, commas, hyphens, or question marks), spell checking, remove duplicates, abbreviations, and acronym from the textual data.
- (2) **Emoji and stopword remover:** Stopwords frequently appear in a text but do not convey significant meaning. Special characters, symbols, emojis and emoticons, numeric values, and removing stopwords can reduce the data's complexity and help simplify the text data and bring more effective textual content.
- (3) **Tokenization:** Dividing text data into smaller units, known as tokens, helps organize and understand it better. These tokens can be individual words, complete sentences, or smaller components such as characters or subwords. The process of tokenization assists in comprehending the structure of the text.
- (4) **Stemming and normalization:** In text analysis, words can be reduced to their base or root form through lemmatization or stemming.

Lemmatization considers the word's context and converts it to its base form, while stemming applies simpler rules to truncate words to their root form. These techniques help improve accuracy by reducing word variations. Additionally, we normalized the data by converting the text to its appropriate platform.

- **Proposed model:** For our investigation, two models have been put forth. We utilized two feature engineering techniques, one of them HindiBERT [35] and another TF-IDF [23]. For DL models, Bert feature engineering approaches are utilized, and for ML models, TF-IDF. Our study initially used an ensemble of ML and BERT base DL models. The two proposed model is described below:
HindiBERT-LSTM Mode: Our DL model is created using LSTM [36] and HindiBERT [35]. Hindi textual sequences are employed as the input data for the classification process. The HindiBERT model receives the input sequences and uses them to produce contextualized word embeddings for each word in the input sequence. The HindiBERT basic model captures the semantic meaning of the words and their contextual information. Table 1 displays the HindiBERT's parameters. The LSTM layer receives the contextual word embeddings from the HindiBERT base model. The embeddings' sequential data is processed by the LSTM layer, which also discovers patterns and dependencies between the sequence's elements. Following that, a dense layer with softmax activation

Table 1 Hyperparameters of HindiBERT

Parameters	Value
Model	Monsoon-nlp/Hindibert
Length	500
Class	2

receives the output from the LSTM layer. The dense layer produces the final predicted probabilities for each class in the classification task by applying a fully connected operation on the sequential representation from the LSTM. A probability distribution over the classes, reflecting the likelihood of each class given the input sequence, is the output of the dense layer. These probabilities represent the predictions made by the model for the classification job. The parameters of the LSTM layer are shown in Table 2. Accordingly, in the suggested model, information moves from the input sequences through the HindiBERT base model, the LSTM layer, and then the dense (Output) layer. Through the hierarchical processing of the layers, the model learns the correlations and patterns in the input sequences, ultimately producing the expected probabilities for classification. Our study also used the sigmoid activation function to classify the probability. But the best outcome was produced by the softmax activation. The model architecture is shown in Fig. 4.

Ensemble method: To compare our results, we looked at the dataset’s effectiveness for ML modeling; we used an ensemble ML model. We suggested a boosting ensemble method in our research. After using several boosting ensemble methods, XGBoost (eXtreme Gradient Boosting) [37] performed the best in our situation. It is a gradient-boosting implementation that has been optimized. It trains the model using gradient descent to minimize a loss function using a tree-based model as the base model. Regularization and early halting, two overfitting prevention methods included in XGBoost, aid in managing the model’s complexity and preventing it from becoming too closely tailored to the training set of data. The types and values of the parameters utilized heavily influence the performance of ensemble models. So we used the hyper-tuning [38] strategy to find the best parameters for our model. Table 3 displays the value and the XGBClassifier parameters utilized during the hyper-tuning procedure. The word n-gram and TF-IDF feature engineering methods are used in XGBClassifier. The TF-IDF makes it possible to accurately evaluate each word’s weight because the IDF considers how frequently a term appears in various documents. Sarcasm is typically used in context. Therefore, it might be challenging to identify sarcasm in some cases without understanding the

Table 2 Hyperparameters of LSTM layer

Parameters	Value
Activation	Softmax
Learning rate	1e-4
Batch size	32
Epoch	100

Fig. 4 Proposed LSTM model architecture

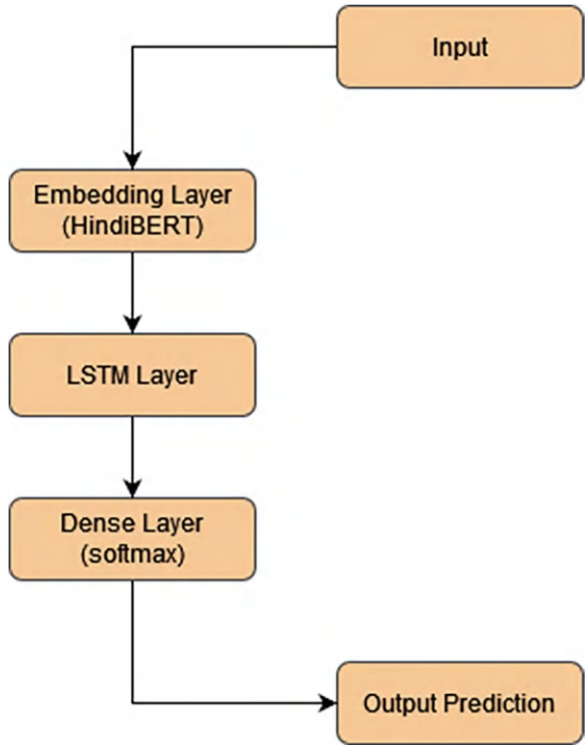


Table 3 Parameters and values of XGBClassifier during hyper-tuning

Parameter	Value
Learning_rate	0.2, 0.1, 0.05, 0.01
Max_depth	3, 5, 7
Min_child_weight	1, 3, 5
Gamma	0.0, 0.1, 0.2
Subsample	0.6, 0.8, 1.0

whole context of a remark. Because of this, we merged n-grams with TF-IDF [23]. The following n words or characters make up this. In our work, we used the word n-grams.

3 Result and Discussion

HindiBert is used as an embedding layer in the DL. The combination of TF-IDF and the word n-gram is utilized as feature engineering in the ensemble model to detect sarcasm in Hindi from social media text. We have evaluated the

Table 4 Performance of our proposed models

Evaluation metrics	HindiBERT-LSTM (%)	Ensemble (%)
Precision	91.54	87.98
Recall	92.96	87.54
<i>F</i> -score	90.54	86.97
Accuracy	95.54	88.65

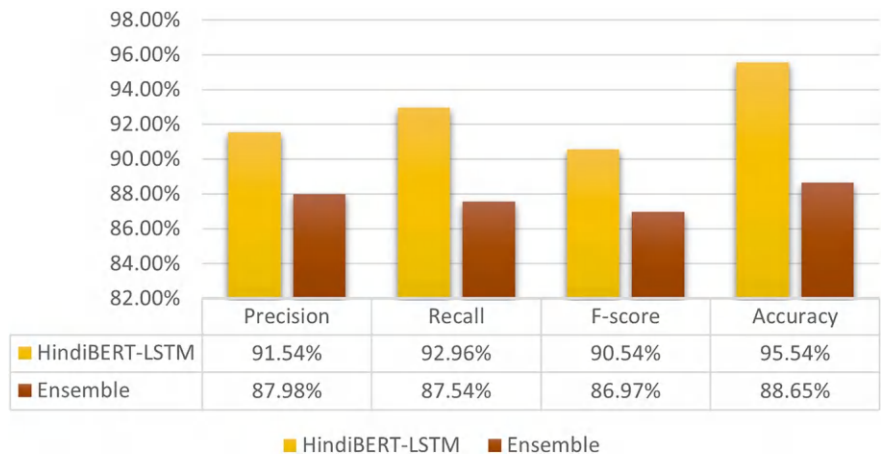


Fig. 5 Performance of our proposed models

performance of our suggested model after deploying it. To assess the effectiveness, we take four evaluation parameters into account. Precision, recall, F-score, and accuracy were calculated on the test dataset. We shall now discuss the performance. The performance of our suggested model is displayed in Table 4. Figure 5 depicts the performance graphically as well. According to data from Table 4, our HindiBERT-LSTM model has a precision of 91.54%, meaning that 91.54% of the cases predicted to be sarcastic were. This high level of precision indicates how well the model recognizes sarcastic sentiment in Hindi text. The precision of 87.98% obtained with the ensemble model is a little lower. Recall, often called sensitivity or true positive rate, gauges how well a model can spot sarcastic expressions. According to this metric, our HindiBERT-LSTM model successfully recognizes 92.96% of the dataset’s sarcastic instances or recall. This strong recall demonstrates the model’s efficacy in capturing and identifying sarcastic sentiment. The recall for the ensemble model is 87.54%, which is somewhat less. With an F-score of 90.54%, our HindiBERT-LSTM model demonstrates its ability to strike a harmonic balance between recall and precision. The *F*-score offers a balanced assessment of the model’s performance because it integrates precision

Table 5 True labels and predicted labels by HindiBERT-LSTM

Data	English translation	Actual label	Predicted label
कामयाब व्यक्ति अपने चेहरे पर दो ही चीजें रखता है- मुस्कराहट और खामोशी	Successful people keep only two things on their faces— smile and silence	Non_sarcastic	Non_sarcastic
आदि से अनंत तक बस यही रही है परंपरा कायर भोगे दुख सदा वीर भोग्य वसुंधरा	From beginning to end this is the only tradition Cowardly suffer	Non_sarcastic	Non_sarcastic
आलू से सोना बनाने की तकनीक खोजने वाले महान वैज्ञानिक की चिंताओं को सिरे से खारिज नहीं किया जा सकता	The concerns of the great scientist who discovered the technique of making gold from potatoes cannot be dismissed outright	Sarcastic	Non_sarcastic
आज तो चीन की ख़ैर नहीं, टुकड़े-टुकड़े कर देंगे हमारे TT चैनल वाले ! शाम तक कड़ी निंदा भी हो जाएगी, देख लेना	Today China is not well, our TV channels will break it into pieces! There will be strong condemnation till evening, see	Sarcastic	Sarcastic
आदि से अनंत तक बस यही रही है परंपरा कायर भोगे दुख सदा वीर भोग्य वसुंधरा	From beginning to end this is the only tradition Cowardly suffer	Non_sarcastic	Sarcastic

and recalls into one metric. This shows that the model can minimize false positives and negatives while accurately predicting sarcastic sentiment. The F-score of the ensemble model is 86.97%, which is somewhat less. The completeness of the model's predictions is measured by accuracy. Our HindiBERT-LSTM model successfully classifies 95.54% of the cases in the dataset with an accuracy score of 95.54%. This high level of accuracy demonstrates the model's general efficacy in sarcasm detection. The accuracy of the ensemble model is 88.65%. The importance of combining the strength of HindiBERT and LSTM architectures for sarcasm detection in Hindi text is highlighted by the improved performance of our proposed HindiBERT-LSTM model compared to the ensemble model. The DL model's improved precision, recall, F-score, and accuracy demonstrate how well it can recognize and categorize sarcastic sentiment in texts written in Hindi. These outcomes highlight the usefulness and potential of our suggested method for Hindi sarcasm recognition tests. The predicted label by the models and the actual label are displayed in Table 5.

4 Conclusion

Social media has grown in popularity as a platform for individuals to share their views and opinions in recent years, facilitating the development of activities like opinion mining and emotion analysis. Detecting sarcasm through linguistic analysis is highly subjective and challenging for linguistic approaches. Sarcasm heavily

depends on context, so we use textual context to identify sarcastic text accurately. In natural language processing, the challenge of sarcasm recognition from Hindi text offers both potential and obstacles. Detecting sarcastic sentiment in Hindi is difficult when there isn't enough data for training and testing. We achieved substantial advancements in sarcasm identification by utilizing DL and ML techniques, and we saw promising results in our ability to identify sarcastic text accurately. It's impressive that HindiBERT-LSTM achieved a 95.54% accuracy rate in detecting sarcasm in Hindi text. This shows that combining pre-trained language models like HindiBERT with recurrent neural networks such as LSTM can effectively capture the complex linguistic patterns and contextual cues associated with sarcasm. However, there is room for improvement, particularly in enhancing performance with a larger dataset. Utilizing a larger dataset could provide additional context and lead to improved accuracy.

References

1. Mane S, Khatavkar V (2023) Researchers eye-view of sarcasm detection in social media textual content. arXiv preprint [arXiv:2304.08582](https://arxiv.org/abs/2304.08582)
2. Kumar A, Kumar MR, Shribala N, Singh N, Gunjan VK, Arif M et al (2022) Dynamic wavelength scheduling by multiobjectives in obs networks. J Math 2022
3. Das D, Clark AJ (2018) Sarcasm detection on Facebook: a supervised learning approach. In: Proceedings of the 20th international conference on multimodal interaction: adjunct, 2018, pp 1–5
4. Zhang M, Zhang Y, Fu G (2016) Tweet sarcasm detection using deep neural network. In: Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers, 2016, pp 2449–2460
5. Avvaru A, Vobilisetty S, Mamidi R (2020) Detecting sarcasm in conversation context using transformer-based models. In: Proceedings of the second workshop on figurative language processing, 2020, pp 98–103
6. Rajadesingan A, Zafarani R, Liu H (2015) Sarcasm detection on twitter: a behavioral modeling approach. In: Proceedings of the eighth ACM international conference on web search and data mining, 2015, pp 97–106
7. Rudra Kumar M, Gunjan VK (2022) Peer level credit rating: an extended plugin for credit scoring framework. In: ICCCE 2021: proceedings of the 4th international conference on communications and cyber physical engineering. Springer, pp 1227–1237
8. Pawar N, Bhingarkar S (2020) Machine learning based sarcasm detection on twitter data. In: 2020 5th international conference on communication and electronics systems (ICCES). IEEE, pp 957–961
9. Swami S, Khandelwal A, Singh V, Akhtar SS, Shrivastava M (2018) A corpus of English-Hindi codemixed tweets for sarcasm detection. arXiv preprint [arXiv:1805.11869](https://arxiv.org/abs/1805.11869)
10. Liu P, Chen W, Ou G, Wang T, Yang D, Lei K (2014) Sarcasm detection in social media based on imbalanced classification. In: Web-age information management: 15th international conference, WAIM 2014, Macau, China, June 16–18, 2014. Proceedings 15. Springer, pp 459–471
11. Ptáček T, Habernal I, Hong J (2014) Sarcasm detection on Czech and English twitter. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers, 2014, pp 213–223
12. Liebrecht CC, Kunneman FA, van Den Bosch AP (2013) The perfect solution for detecting sarcasm in tweets# not

13. Lunando E, Purwarianti A (2013) Indonesian social media sentiment analysis with sarcasm detection. In: 2013 International conference on advanced computer science and information systems (ICACSIS). IEEE, pp 195–198
14. Bedi M, Kumar S, Akhtar MS, Chakraborty T (2021) Multi-modal sarcasm detection and humor classification in code-mixed conversations. *IEEE Trans Affect Comput* 2021
15. Bharti SK, Babu KS, Raman R (2017) Context-based sarcasm detection in Hindi tweets. In: 2017 Ninth international conference on advances in pattern recognition (ICAPR). IEEE, 2017, pp 1–6
16. Dhir R, Mishra SK, Saha S, Bhattacharyya P (2019) A deep attention based framework for image caption generation in Hindi language. *Computación y Sistemas* 23(3):693–701
17. Kumar M, Rani Z (2019) Energy-efficient scheduling of cloud data center servers, pp 1769–1772 (Online). Available: <https://doi.org/10.35940/iji-tee.k1759.0981119>
18. Hughes H (2009) 1000 languages: the worldwide history of living and lost tongues. *Ref Rev* 23(2):27–27
19. Aggarwal A, Wadhawan A, Chaudhary A, Maurya K (2020) Did you really mean what you said? Sarcasm detection in Hindi-English code-mixed data using bilingual word embeddings. *arXiv preprint arXiv:2010.00310*
20. Sun Y, Zhang H, Yang S, Wang J (2022) Efafn: An efficient feature adaptive fusion network with facial feature for multimodal sarcasm detection. *Appl Sci* 12(21):11235
21. Bharti SK, Gupta RK, Shukla PK, Hatamleh WA, Tarazi H, Nuagah SJ (2022) Multimodal sarcasm detection: a deep learning approach. *Wirel Commun Mob Comput* 2022:1–10
22. Das D, Clark AJ (2018) Sarcasm detection on flickr using a CNN. In: *Proceedings of the 2018 international conference on computing and big data*, 2018, pp 56–61
23. Kumar AA, Kaur P, Sharma M, Tripathi K (2021) Sarcasm detection in English and Hindi sentences
24. Khare A, Gangwar A, Singh S, Prakash S (2022) Sentiment analysis and sarcasm detection of Indian general election tweets. *arXiv preprint arXiv:2201.02127*
25. Mathur P, Shah R, Sawhney R, Mahata D (2018) Detecting offensive tweets in Hindi-English code-switched language. In: *Proceedings of the sixth international workshop on natural language processing for social media*, 2018, pp 18–26
26. Dwaram JR, Madapuri RK (2022) Crop yield forecasting by long short-term memory network with Adam optimizer and huber loss function in Andhra Pradesh, India. *Concurr Comput Pract Exp* 34(27):e7310
27. Patil PK, Kolhe S (2022) Marathisarc: a marathi tweets dataset for automatic sarcasm detection of marathi tweets. *Grenze Int J Eng Technol (GIJET)* 8(2)
28. Bharti SK, Sathya Babu K, Jena SK (2017) Harnessing online news for sarcasm detection in Hindi tweets. In: *International conference on pattern recognition and machine intelligence*. Springer, 2017, pp 679–686
29. Prasad M, Babu AR, Krishna AV, Kumar MR (2019) Reciprocal repository for decisive data access in disruption tolerant networks. *Int J Innov Technol Explor Eng* 9(1):4430–4434
30. Sharma A, Kabra A, Jain M (2022) Ceasing hate with moh: hate speech detection in Hindi–English codeswitched language. *Inf Process Manage* 59(1):102760
31. Afyati A, Azhari A, Sari AK, Karim A (2020) Challenges of sarcasm detection for social network: a literature review. *JUITA: Jurnal Informatika* 8(2):169–178
32. Hirschey JK (2014) Symbiotic relationships: pragmatic acceptance of data scraping. *Berkeley Tech LJ* 29:897
33. Maruf AA, Biplob MNH, Khanam F (2022) Covid-19 vaccine sentiment detection and analysis using machine learning technique and NLP. In: *International conference on machine intelligence and emerging technologies*. Springer, pp 401–414
34. Alasadi SA, Bhaya WS (2017) Review of data preprocessing techniques in data mining. *J Eng Appl Sci* 12(16):4102–4107
35. Gupta A, Sukumaran R, John K, Teki S (2021) Hostility detection and covid-19 fake news detection in social media. *arXiv preprint arXiv:2101.05953*

36. Kumar A, Narapareddy VT, Srikanth VA, Malapati A, Neti LB (2020) Sarcasm detection using multihead attention based bidirectional LSTM. *IEEE Access* 8:6388–6397
37. Chen Z, Jiang F, Cheng Y, Gu X, Liu W, Peng J (2018) Xgboost classifier for DDoS attack detection and analysis in SDN-based cloud. In: 2018 IEEE international conference on big data and smart computing (bigcomp). IEEE, 2018, pp 251–256
38. Li Y, Shen Y, Jiang H, Zhang W, Li J, Liu J, Zhang C, Cui B (2022) Hyper-tune: towards efficient hyper-parameter tuning at scale. arXiv preprint [arXiv:2201.06834](https://arxiv.org/abs/2201.06834)

Enhancing Breast Cancer Prediction: A Comparative Analysis of CNN and Ensemble Models with Data Balancing Techniques



Y. Indira Priyadarshini, K. Mahesh Babu, Katika Sai, B. Supriya, Ala Chaitanya Kumari, and Epuru Vyshnavi

Abstract Breast cancer causes a significant number of annual deaths and is the principal type of cancer that results in the deaths of women worldwide. Developing cancer prediction and diagnosis is crucial for maintaining a healthy life. It is imperative to achieve high accuracy in cancer prediction to improve treatment options for patients and increase their chances of survival. Healthcare datasets are readily accessible, encouraging researchers to use them for study. So, many researchers work on it, and limitations still exist. In this work, a performance comparison between an Ensemble and a proposed Convolutional Neural Networks (CNNs) Models is conducted to overcome the shortcomings of single-model performance on breast cancer prediction. Also, we have applied data balancing techniques to get more accurate results. The results found that the proposed CNN Model had a higher level of accuracy, reaching 97.65%. In comparison, the Ensemble Model achieved an accuracy of 96.21%. Additionally, the proposed CNN Model demonstrated a precision of 95.21%, a recall of 96.43%, and an F -score of 96.35%, while the Ensemble Model had a precision of 94.13%, a recall of 95.76%, and an F -score of 95.50%. The proposed CNN Model presented superior accuracy and a more even balance between precision and recall, proving its effectiveness in detecting positive breast cancer cases, and the goal is to reduce differences in diagnoses and improve accuracy by merging these models.

Keywords Breast cancer · Machine learning · Deep learning · CNN · Ensemble · NLP

Y. Indira Priyadarshini (✉) · K. Mahesh Babu · K. Sai · B. Supriya · A. C. Kumari · E. Vyshnavi
Computer Science and Engineering, Ravindra College of Engineering for Women, Kurnool, India
e-mail: indusai.yashu@gmail.com

K. Sai
e-mail: indusai.yashu@gmail.com

1 Introduction

The World Health Organization (WHO) predicts that there will be 19.3 million cases of cancer in 2025 [1]. Breast cancer was first recorded in Egypt in 3000 BC [2]. Approximately 14 million individuals annually experience various forms of cancer, predominantly resulting from developing malignant tumors, and breast cancer is the second most rampant cancer type detected in women [3]. Cancer is a malignant growth that can be unpleasant when a single cell divides unchecked, which causes tumors. The two classifications for cancer detection are benign and aggressive. Rapid growth and invasion cause tissue damage in a malignant tumor [4]. Breast cancer is the name given to the malignant tissue developing in the breast. Increased breast mass, altered breast size and shape, altered breast skin tone, soreness in the breasts, and modifications to the breast's genetic makeup are all examples of breast cancer symptoms [5]. Breast cancer is the second superior cause of death for women globally, behind heart disease, and more than 8% of women experience it at some point [6]. Every year, more than 500,000 women are diagnosed with breast cancer, pursuant to the WHO's annual report. Medical professionals advise undergoing mammography to detect cancer early, which can decrease the chance of mortality by 20–40% [7].

Considerable analyses have shown that detecting breast cancer early and providing appropriate treatment can significantly reduce breast cancer mortality rates in the long run [8]. Medical professionals have used mammography alongside ultrasound as an additional imaging technique [9]. While magnetic resonance imaging (MRI) can detect small lesions that mammography cannot, it is expensive and has low specificity, potentially leading to overdiagnosis [10]. Positron emission tomography (PET) is the reliable method for determining a tumor's spread and response to treatment [11]. Recently, microwave imaging (MI) techniques have been recommended as a safe and cost-effective alternative to mammography for detecting breast cancer [12]. The classification of benign and malignant tumors using different breast tumor measurements rather than standard diagnostic lab tests like breast biopsies, positron emission tomography, and magnetic resonance imaging is also referred to as a data mining problem focusing on breast cancer prediction [13–15].

Data mining [16] techniques are used for prediction and categorization, with classification being one of the primary methods. Classification techniques are helpful in several domains as they can assist in forecasting disease diagnoses and conducting logical tests [17]. Medical data mining often uses classification techniques and various diseases have been predicted using the data mining approach [18]. Medical or clinical data pertains to information gathered from routine patient care or clinical trial plans related to human health. This includes electronic health records that contain patient health information. With the help of AI-based technique [19], the information can be effectively processed and analyzed to provide clear output for end users [20].

ML, with its subcategory and DL [21], has shown promising outcomes in the healthcare domain in recent times [22]. Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), Multi-layer perception (MLP), K-Nearest

Neighbor (kNN), Naive Bayes (NB), and Decision Tree (DT) are gaining popularity alongside the growth of data mining in the information systems field, and using these ML tools with appropriate hyperparameters can aid in efficiently identifying tumors [23]. In contrast, Numerous Deep Learning algorithms (CNNs), Recurrent Neural Networks (RNNs), Gated Recurrent Unit (GRU), and Long Short-Term Memory Networks (LSTM) [24]) are available; however, not all have been explored for detecting breast cancer [1]. Our main research contributions focus on predicting breast cancer and the significance of our study:

- We balance our dataset by utilizing the data balancing technique.
- We have put forward a CNN Model and achieved the anticipated level of accuracy.
- We created an Ensemble Model and compared it with our proposed CNN Model to demonstrate its effectiveness further.

To structure this paper, we have divided it into several sections. Firstly, in Sect. 2, we will review the various approaches to detecting breast cancer. Following this, we will outline our methodology statement in Sect. 3. Section 4 will present the experimental outcomes of the models we have used. Finally, we will conclude by presenting our findings.

2 Related Work

In the following section of this article, we discuss two aspects of earlier studies on breast cancer detection. Firstly, we would like to present some of the latest work done by ML-based techniques, and lately, we will discuss deep learning-based work on predicting breast cancer and several classifier types employed to determine accuracy.

One of the biggest issues facing people in developing countries is cancer. Some cancer types currently have no available treatments, breast cancer is one of the more prevalent cancers, and early detection is crucial to its treatment [25]. ML is widely used for predicting breast cancer, as it can detect critical features from medical data sets. Recently, more attention has been given to using ML methods for medical diagnosis [26]. Using ML models for determining breast cancer results in the quick identification of tumors [27]. We have obtained valuable insights regarding data collection: Scholars have gathered data from various sources, including the Wisconsin Breast Cancer dataset (WBCD) [28], Wisconsin Diagnostic Breast Cancer [29], and the U.S. National Cancer Institute's Surveillance, Epidemiology, WEKA [30] and End Results (SEER) [13]. Upon thoroughly reviewing the literature, we have discovered some significant findings that most papers have utilized: SVM, KNN, LR, DT, MLP, XGBoost [5], and NB in the implementation. Conversely, the Weighted Area Under the Receiver Operating Characteristic Curve Ensemble (WAUCE) [13] is implemented by some researchers. Scrutinizing the outcomes, the highest accuracy gained from the ML-based method is 100% [31], 99.28% [28], 98.01% [25], and 97.89% [13], where the LR model, including limited features, produced the highest accuracy.

The deep learning-based methodology [22] is a strategy that shows promise for the early identification of breast cancer. After evaluating the entire literature, we identified that Genetically Optimized Neural Networks (GONNs) [32], LSTM, GRU, CNN, and RNN were used the most. Bhardwaj et al. [32] evaluated GONN and backpropagation models based on classification accuracy, sensitivity, specificity, confusion matrix, ROC curves, and AUC under ROC curves. The results showed that the GONN model achieved a perfect accuracy of 100% in tenfold cross-validation, while the classical and backpropagation models achieved higher accuracies of 99.63% and 100%, respectively. Another study found that LSTM and GRU algorithms were highly effective, surpassing an accuracy of 99%. These models demonstrated exceptional performance in the classification task [1].

Islam et al. [33] evaluated various ML and DL algorithms using the WBCD from the UCI database, including SVM, KNN, RF, ANNs [11], and LR. Performance was measured using various metrics, and the findings revealed that ANNs achieved the highest accuracy, 98.57%. Meanwhile, SVM obtained an accuracy of 97.14%. Another study [34] analyzed a dataset of 4902 patient records to predict breast cancer survival using ML and deep learning approaches. MLP, RF, and DT classifiers achieved 88.2%, 83.3%, and 82.5% accuracy rates, respectively.

Most research efforts are centered around utilizing ML techniques to identify breast cancer cases. However, DL also holds significant importance in this area.

3 Methodology

We took several actions to perform the research. The procedures we used in this study are detailed in full in this section. Our study framework is shown in Fig. 1.

A. Dataset Description and Preprocessing

The UCI ML repository served as the source of the dataset [35]. 569 instances and 32 attributes make up the dataset. We start by removing the Unnamed: 32 columns. The dataset is more crucial than any one attribute. To create the prediction system, we considered various features from that dataset. The dataset comprises two classes. In the research paper's data preprocessing phase, we took several critical actions to guarantee the data's quality and fit for our breast cancer prediction objective. We checked the dataset for missing or null values to ensure the data was complete and prepared for further analysis. We also examined how samples were distributed throughout the various classes and found that the dataset was unbalanced. We used random oversampling [36] to address the class imbalance problem.

We also used one-hot encoding to transform the categorical values in our dataset into a binary representation, as our dataset comprised categorical variables. Thanks to this translation, we successfully included the category data in our ML model. A binary feature was used to represent each category within a categorical variable, which helped to capture any relevant linkages and patterns related to each category.

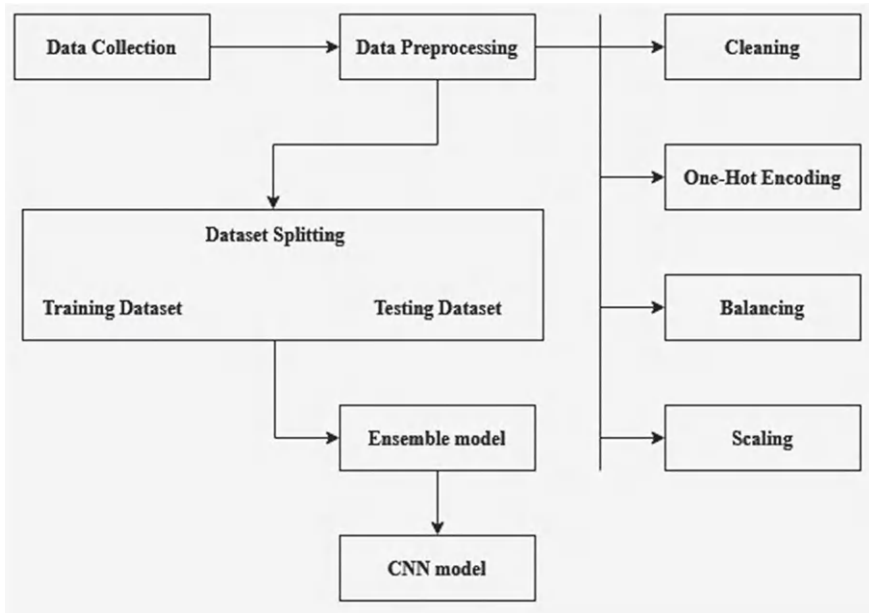


Fig. 1 Framework of our research

We scaled the features in our data to guarantee the best possible feature representation. To standardize the feature values and bring them within the same scale, feature scaling is a vital step in preprocessing. This stage aids in preventing the dominance of some features over others in the learning process because of their bigger scales. We made sure that every feature contributed proportionally to the learning process of the model by using a feature transformation approach, such as standardization or min-max scaling, which increased model performance and stability. Our preprocessed data was then divided into training and testing sets. The model was trained using 70% of the data, while the remaining 30% was used for testing. Figure 2 illustrates the correlation matrix for the dataset's features. By highlighting the strength and direction of their linear association, the correlation matrix sheds new light on the relationships between the various elements in the dataset.

B. Model

- Ensemble Learning:** Boosting model was employed to reduce training errors [37]. It makes a group of weak students become strong students. Boosting is an iterative strategy in which each iteration aims to categorize the data incorrectly classified in the previous iteration accurately. The combined forecasts from each separate model are weighted to produce the final projection. Overfitting is a problem that is specifically addressed by Boosting algorithms like XGBoost. This aids in concentrating the model's attention on the trickier examples, leading to a

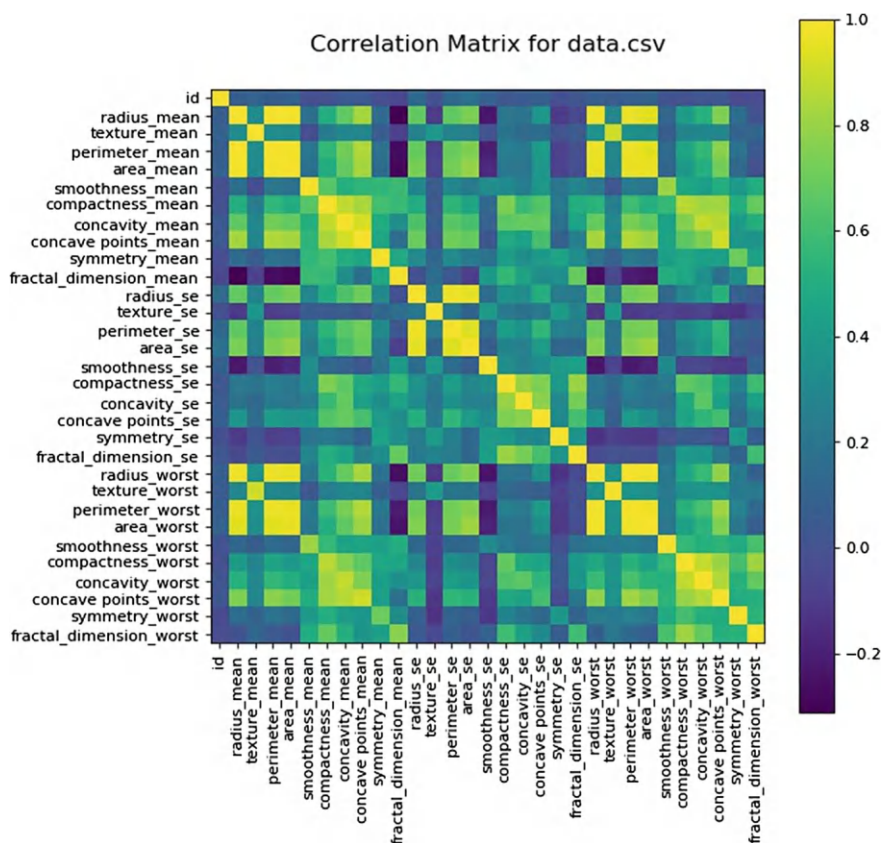


Fig. 2 Correlation matrix for the dataset's features

more reliable model. Gradient Boosting has been enhanced in XGBoost (eXtreme Gradient Boosting). It trains the model using gradient descent to minimize a loss function using a tree-based model as the base model. Regularization and early halting, two overfitting prevention methods included in XGBoost, aid in managing the model's complexity and preventing it from becoming too closely tailored to the training set of data. The parameter selection has a significant impact on XGBClassifier performance. The performance of the XGBClassifier can be improved with the use of suitable parameters and values. The XGBClassifier parameters used in our research are displayed in Table 1.

- **Proposed CNN Model:** In this study, we suggested a DL architecture designed exclusively for predicting breast cancer [38]. CNNs have been used greatly in the model architecture to capture significant patterns and features from the input

Table 1 Parameters of XGBClassifier

Parameter	Value	Parameter	Value
n estimators	200	Colsample bytree	0.8
Learning rate	0.1	Gamma	0.1
Max depth	6	Reg lambda	1.0
Subsample	0.8	Random state	42
Min child weight	1	–	–

data. The model had numerous layers to ensure efficient learning and top performance, and the training procedure was carried out across 50 iterations. The one-dimensional Convolutional layer, which served as a feature extractor with 32 filters and a 2 kernel size, was the first layer in the architecture. The model’s ability to learn complicated representations was improved by adding non-linearity and using the Rectified Linear Unit (ReLU) activation function. Following the initial Convolutional layer, batch normalization was implemented to normalize the activations. This promotes consistent training and accelerates convergence. A dropout (DO) layer with a rate of 0.2 is operated to prevent overfitting. This layer randomly turns off 20% of the neurons during training. Using 64 filters and a 2 kernel size, a second Convolutional layer was added to the model to improve further its capacity to capture pertinent features. The ReLU activation function was used to incorporate non-linearity and extract higher-level representations. After this layer, activations were batch normalized to promote steady training and lessen the effects of covariate shift. A DO layer with a rate of 0.5 was used to randomly disable 50% of the neurons during training to prevent overfitting. The multidimensional feature maps were reduced into a one-dimensional vector by adding a flattened layer after the convolutional layers. This modification made it easier to reach the completely interconnected levels. A Dense layer with 64 neurons and ReLU activation was added to capture complicated correlations within the learned characteristics. This layer made the extraction of higher-level representations and the facilitation of decision-making processes possible. After the fully connected layer, a second DO layer with a rate of 0.5 was added to reduce overfitting, allowing for increased generalization and model robustness. The Dense layer comprised the architecture’s top layer, which had a single neuron and sigmoid activation function. This layer gave the output prediction, representing the probability of developing breast cancer. The model architecture is summarized in Table 2.

4 Result and Analysis

The Ensemble Model that has been implemented and the proposed model are evaluated using various evaluation criteria.

We compute the Precision (Prc), Recall (Rec), *F*-score, and Accuracy (Acc) to evaluate the outcomes. In this section, we’ll talk in-depth about the outcomes.

Table 2 Parameters and layers used in the CNN architecture

Layer	Parameter(s)
One-dimensional Convolution	Filters: 32, Kernel size: 2
DO	Rate: 0.2
One-dimensional Convolution	Filters: 64, Kernel size: 2
DO	Rate: 0.5
Dense	Neurons: 64, Activation: ReLU
DO	Rate: 0.5
Dense	Neurons: 1, Activation: Sigmoid

Two sections will be used to present them. Before discussing the conclusion of the proposed model, we will first discuss the Ensemble Model’s performance. Finally, we shall compare the models’ accuracy.

• **Performance of Ensemble Model**

From Table 3, our Ensemble Model’s Prc was estimated to be 94.13%. Out of all cases anticipated to be positive, Prc is the percentage of positive situations that were accurately identified. A higher Prc score reflects the model’s accuracy in identifying positive breast cancer cases by showing a lower rate of erroneous positive predictions. The Rec was determined to be 95.76%. A higher Rec value indicates a lower rate of false negative predictions, demonstrating the model’s effectiveness in capturing positive breast cancer cases. Additionally, we produced the *F*-score, which combines Prc and Rec into a single statistic to provide a fair evaluation of the model’s performance. Our Ensemble Model produced an *F*-score of 95.50%. These evaluation criteria show our Ensemble Model’s excellent performance in predicting breast cancer. The low rate of false positive predictions indicated by the high precision score of 94.13% reduces the likelihood of a false positive diagnosis. The high recall score of 95.76% demonstrates the model’s ability to identify positive cases of breast cancer accurately. Additionally, the *F*-score of 95.50% indicates a balanced performance when both precision and recall are considered. These findings support the Ensemble Model’s potential for accurate breast cancer prediction and demonstrate its value in advancing medical diagnosis and decision-making.

• **Performance of CNN Model**

From Table 4, our CNN Model’s Prc was estimated to be 95.21%. Out of all cases anticipated to be positive, Prc is the percentage of positive situations that were

Table 3 Precision, recall, *F*-score of Ensemble Model

Evaluation metrics	Value (%)
Prc	94.13
Rec	95.76
<i>F</i> -score	95.50

Table 4 Precision, recall, *F*-score of CNN model

Evaluation metrics	Value (%)
Prc	95.21
Rec	96.43
<i>F</i> -score	96.35

accurately identified. A greater accuracy score demonstrates the model’s capacity to precisely detect positive breast cancer cases by indicating a reduced rate of erroneous positive predictions. The Rec was determined to be 96.43%. A higher Rec value means a lower rate of false negative predictions, demonstrating effectiveness in capturing true breast cancer cases. In addition, we calculated the *F*-score, which provides a fair assessment of the model’s performance by combining Prc and Rec into a single statistic. Our CNN Model earned an *F*-score of 96.35%. The model’s capacity to simultaneously achieve high precision and high recall is shown by the *F*-score (Figs. 3 and 4).

• **Comparison of Models**

The CNN model outperformed the Ensemble Model regarding Prc, Rec, and *F*-score when we compare the two models. The Ensemble Model had a Prc value of 94.13%, but the CNN Model had a Prc value of 95.21%. This suggests that the CNN Model was more accurate in identifying positive breast cancer cases because it had a lower rate of erroneous positive predictions. Additionally, the Ensemble Model’s Rec value was 95.76%, but the CNN Model’s Rec value was 96.43%. This suggests that the CNN Model was more successful in identifying genuine positive breast

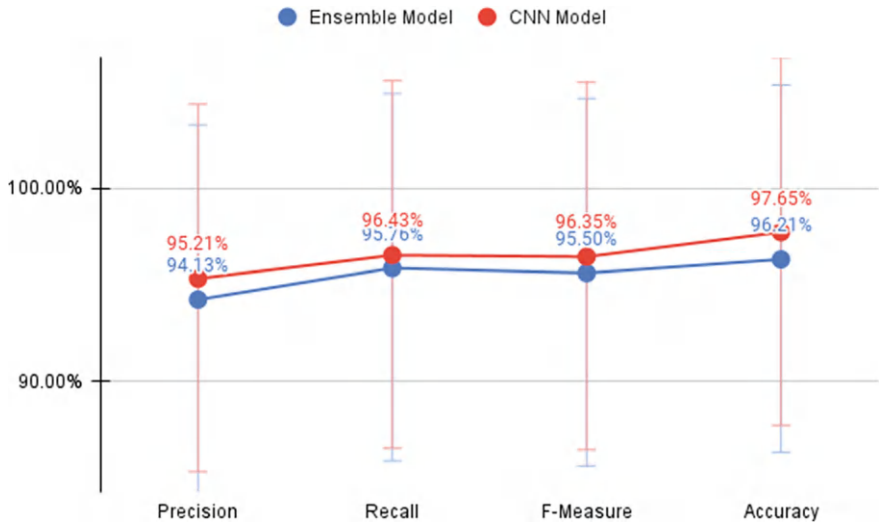


Fig. 3 Performance of all models

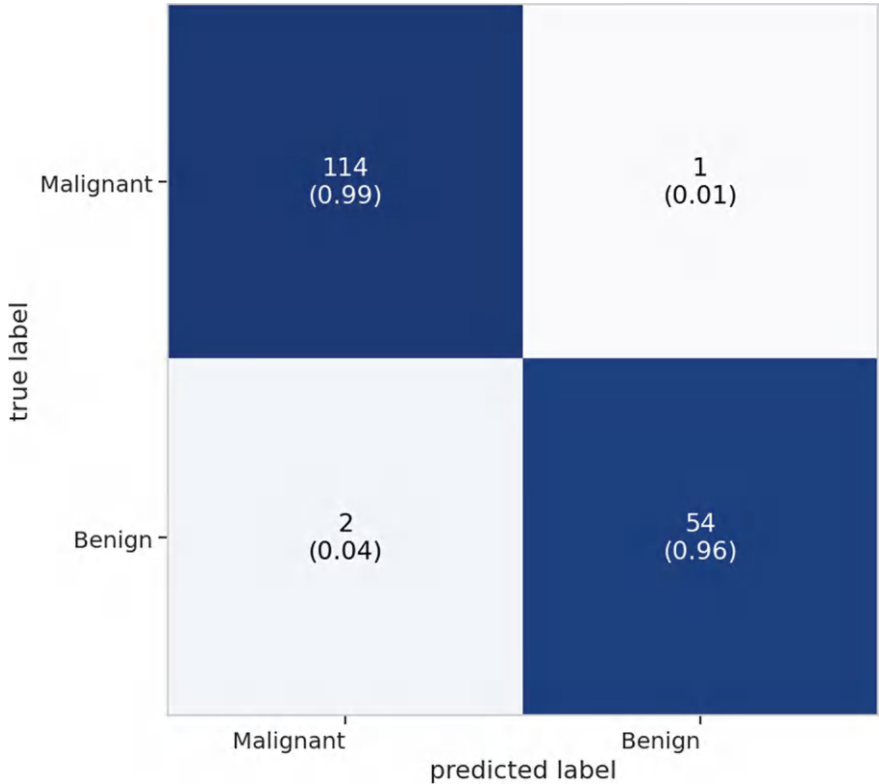
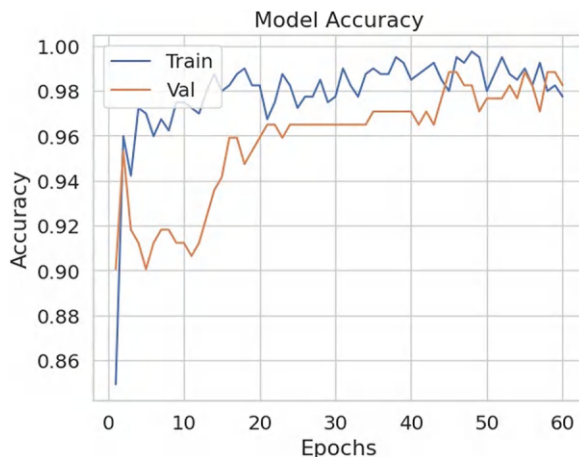


Fig. 4 Confusion matrix

cancer cases and had a lower rate of erroneous negative predictions. Additionally, the Ensemble Model’s *F*-score of 95.50% was surpassed by the CNN Model’s *F*-score of 96.35%. The CNN Model’s greater *F*-score denotes a balanced performance that considers precision and recall simultaneously. From Fig. 5, the percentage of successfully predicted cases out of all instances that the CNN Model predicted was 97.65%. This high accuracy demonstrates the model’s capacity to forecast the existence of breast cancer accurately, adding to its dependability in clinical decision-making. On the other hand, the Ensemble Model showed a 96.21% level of accuracy. It demonstrates the model’s ability to make correct predictions while being significantly less accurate than the CNN Model. When comparing the two models, we find that the CNN Model fared better in accuracy than the Ensemble Model. The Ensemble Model’s accuracy was 96.21%, while the CNN Model’s accuracy was 97.65%, demonstrating a higher degree of accuracy in predictions. The training and validation figures show the performance of our breast cancer prediction models during training. The charts show how performance metrics have changed across several epochs, giving information about the models’ ability to generalize and learn. The training and validation accuracy curves are shown in Fig. 5.

Fig. 5 Train and validation accuracy



5 Conclusion

Breast cancer is one of the major causes of mortality for women globally, and early detection is key to enhancing survival rates. Medical diagnosis relies heavily on accurately and effectively classifying breast cancer data, which is crucial for ensuring proper treatment and care.

The initial goal of this work is to show breast cancer accurately and to achieve this, we employed an Ensemble learning approach using a Boosting model to minimize training errors and also used the CNN Model. The dataset utilized for this study was obtained from the UCI ML repository. To compare the effectiveness of the Ensemble and CNN Models in predicting breast cancer, we evaluated using various evaluation criteria, including Precision, Recall, *F*-score, and Accuracy, to assess their outcomes. Upon comparing the two models, our proposed CNN Model outperformed the Ensemble Model, which indicates that the proposed CNN Model was more accurate in identifying positive breast cancer cases, as it exhibited a lower rate of erroneous positive predictions.

References

1. Ghosh P, Azam S, Hasib KM, Karim A, Jonkman M, Anwar A (2021) A performance based study on deep learning algorithms in the effective prediction of breast cancer. In: 2021 International joint conference on neural networks (IJCNN). IEEE, 2021, pp 1–8
2. Qasem A, Abdullah S, Sahran S, Hussain RI, Ismail F (2017) An accurate rejection model for false positive reduction of mass localisation in mammogram. *Pertanika J Sci Technol* 25:49–62
3. Sun Y-S, Zhao Z, Yang Z-N, Xu F, Lu H-J, Zhu Z-Y, Shi W, Jiang J, Yao P-P, Zhu H-P (2017) Risk factors and preventions of breast cancer. *Int J Biol Sci* 13(11):1387
4. Key TJ, Verkasalo PK, Banks E (2001) Epidemiology of breast cancer. *Lancet Oncol* 2(3):133–140

5. Islam T, Kundu A, Islam Khan N, Chandra Bonik C, Akter F, Jihadul Islam M (2021) Machine learning approaches to predict breast cancer: Bangladesh perspective. In: International conference on ubiquitous computing and intelligent information systems. Springer, pp 291–305
6. U. D. of Health, C. f. D. C. Human Services, Prevention, and N. C. Institute (1999–2008) Us cancer statistics working group: United States cancer statistics—1999–2008 incidence and mortality web-based report
7. Berry DA, Cronin K, Plevritis S, Fryback D, Clarke L, Zelen M, Mandelblatt J, Yakovlev A, Habbema J, Feuer E (2005) Cancer intervention and surveillance modeling network (CISNET) collaborators. Effect of screening and adjuvant therapy on mortality from breast cancer. *N Engl J Med* 353(17):1784–1792
8. Wang L (2017) Early diagnosis of breast cancer. *Sensors* 17(7):1572
9. Ozmen N, Dapp R, Zapf M, Gemmeke H, Ruiter NV, van Dongen KW (2015) Comparing different ultrasound imaging methods for breast cancer detection. *IEEE Trans Ultrason Ferroelectr Freq Control* 62(4):637–646
10. Hassan AM, El-Shenawee M (2011) Review of electromagnetic techniques for breast cancer detection. *IEEE Rev Biomed Eng* 4:103–118
11. Xu P, Peng Y, Sun M, Yang X (2015) SU-E-I-81: targeting of HER2-expressing tumors with dual PET-MR imaging probes. *Med Phys* 42(6Part6):3260–3260
12. Chen T, Artis F, Dubuc D, Fournie J, Poupot M, Grenier K (2013) Microwave biosensor dedicated to the dielectric spectroscopy of a single alive biological cell in its culture medium. In: 2013 IEEE MTT-S international microwave symposium digest (MTT). IEEE, 2013, pp 1–4
13. Wang H, Zheng B, Yoon SW, Ko HS (2018) A support vector machine-based ensemble algorithm for breast cancer diagnosis. *Eur J Oper Res* 267(2):687–699
14. Kumar M, Rani Z (2019) Energy-efficient scheduling of cloud data center servers, p 1769–1772 (Online). Available: <https://doi.org/10.35940/ijitee.k1759.0981119>
15. Prasad M, Babu AR, Krishna AV, Kumar MR (2019) Reciprocal repository for decisive data access in disruption tolerant networks. *Int J Innov Technol Explor Eng* 9(1):4430–4434
16. Lisboa PJ, Vellido A, Tagliaferri R, Napolitano F, Ceccarelli M, Martín-Guerrero JD, Biganzoli E (2010) Data mining in cancer research [application notes]. *IEEE Comput Intell Mag* 5(1):14–18
17. Ghani MU, Alam TM, Jaskani FH (2019) Comparison of classification models for early prediction of breast cancer. In: 2019 International conference on innovative computing (ICIC). IEEE, 2019, pp 1–6
18. Maimon O, Rokach L (2005) Data mining and knowledge discovery handbook, vol 2, no 2005. Springer
19. Rudra Kumar M, Gunjan VK (2022) Peer level credit rating: an extended plugin for credit scoring framework. In: ICCCE 2021: proceedings of the 4th international conference on communications and cyber physical engineering. Springer, 2022, pp 1227–1237
20. Chaurasia V, Pal S (2020) Applications of machine learning techniques to predict diagnostic breast cancer. *SN Comput Sci* 1(5):270
21. Lakshmi TN, Jyothi S, Kumar MR (2021) Image encryption algorithms using machine learning and deep learning techniques—a survey. In: Modern approaches in machine learning and cognitive science: a walkthrough: latest trends in AI, vol 2. Springer, 2021, pp 507–515
22. Akselrod-Ballin A, Chorev M, Shoshan Y, Spiro A, Hazan A, Melamed R, Barkan E, Herzl E, Naor S, Karavani E et al (2019) Predicting breast cancer by applying deep learning to linked health records and mammograms. *Radiology* 292(2):331–342
23. Gupta P, Garg S (2020) Breast cancer prediction using varying parameters of machine learning models. *Procedia Comput Sci* 171:593–601
24. Dwaram JR, Madapuri RK (2022) Crop yield forecasting by long short-term memory network with Adam optimizer and Huber loss function in Andhra Pradesh, India. *Concurr Comput: Pract Exp* 34(27):e7310
25. Ak MF (2020) A comparative analysis of breast cancer detection and diagnosis using data visualization and machine learning applications. In: Healthcare, vol 8, no 2. MDPI, 2020, p 111

26. Jabbar MA (2021) Breast cancer data classification using ensemble machine learning. *Eng Appl Sci Res* 48(1):65–72
27. Amrane M, Oukid S, Gagaoua I, Ensari T (2018) Breast cancer classification using machine learning. In: 2018 Electric electronics, computer science, biomedical engineerings' meeting (EBBT). IEEE, 2018, pp 1–4
28. Kumari M, Singh V (2018) Breast cancer prediction system. *Procedia Comput Sci* 132:371–376
29. Naji MA, El Filali S, Aarika K, Benlahmar EH, Abdelouhahid RA, Debauche O (2021) Machine learning algorithms for breast cancer prediction and diagnosis. *Procedia Comput Sci* 191:487–492
30. Asri H, Mousannif H, Al Moatassime H, Noel T (2016) Using machine learning algorithms for breast cancer risk prediction and diagnosis. *Procedia Comput Sci* 83:1064–1069
31. Battineni G, Chintalapudi N, Amenta F (2020) Performance analysis of different machine learning algorithms in breast cancer predictions. *EAI Endorsed Trans Pervasive Health Technol* 6(23)
32. Bhardwaj A, Tiwari A (2015) Breast cancer diagnosis using genetically optimized neural network model. *Expert Syst Appl* 42(10):4611–4620
33. Islam MM, Haque MR, Iqbal H, Hasan MM, Hasan M, Kabir MN (2020) Breast cancer prediction: a comparative study using machine learning techniques. *SN Computer Science* 1:1–14
34. Kalafi E, Nor N, Taib N, Ganggayah M, Town C, Dhillon S (2019) Machine learning and deep learning approaches in breast cancer survival prediction using clinical data. *Folia Biol* 65(5/6):212–220
35. Wolberg W, Mangasarian O, Street N, Street W (1995) Breast cancer Wisconsin (diagnostic), UCI Machine Learning Repository. <https://doi.org/10.24432/C5DW2B>
36. Gupta P, Varshney A, Khan MR, Ahmed R, Shuaib M, Alam S (2023) Unbalanced credit card fraud detection data: a machine learning-oriented comparative study of balancing techniques. *Procedia Comput Sci* 218:2575–2584
37. Qiu Y, Liu Y-F, Shu X, Qiao X-F, Ai G-Y, He X-J (2023) Peritumoral radiomics strategy based on ensemble learning for the prediction of Gleason grade group of prostate cancer. *Acad Radiol*
38. Majumdar S, Pramanik P, Sarkar R (2023) Gamma function based ensemble of CNN models for breast cancer detection in histopathology images. *Expert Syst Appl* 213:119022

Supervised Machine Learning Approach for Diabetes Detection and the Impact of Data Balancing Methods



K. Seshadri Ramana, N. Asra Shaheen, S. Safa Chowdary,
Shaik Afroz Jaha, Syeda Uzma Tasneem, and Talari Renuka

Abstract Millions of people are affected by the prevalent disease of diabetes worldwide, and women are the population group most affected by this condition. Having diabetes raises the chances of developing other health illnesses such as heart disease, kidney disease, nerve damage, and blindness due to the damage it causes to blood vessels. Accurately diagnosing diabetes by analyzing relevant data remains a major challenge. Modern healthcare research uses innovative and advanced technologies to diagnose individuals and detect their illnesses based on clinical data. Machine learning (ML) is a technology that can improve the accuracy of diagnosis and detection. In this work, we use the female Pima Indians Diabetic Dataset employing several supervised learning algorithms such as Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), and Logistic Regression (LR) to detect diabetes. Additionally, we used GridSearchCV for hyperparameter tuning to enhance the model's performance and experimented with various parameters. We evaluate the F1 score, precision, and recall for all supervised learning techniques. The predictive accuracy was 92.17%, which was the best accuracy achieved by RF, and in comparison with other research, it showed great promise for addressing the limitations in this field. RF performed exceptionally well and demonstrated its superiority over other ML classifiers. Furthermore, the proposed study aims to enhance the quality of life for people who have diabetes and also foster research development.

Keywords Diabetes mellitus · Machine learning · Hyperparameter tuning · Pima Indian Diabetes Dataset · Healthcare

K. Seshadri Ramana (✉) · N. Asra Shaheen · S. Safa Chowdary · S. A. Jaha · S. U. Tasneem · T. Renuka

Computer Science and Engineering, Ravindra College of Engineering for Women, Kurnool, India
e-mail: ramana.kothapalli@gmail.com

1 Introduction

According to the World Health Organization (WHO), 415 million individuals are affected with diabetes, also called diabetes mellitus (DM), and are at risk of potentially losing their lives due to the condition [1]. According to a study by the International Diabetes Federation, the number of individuals with diabetes is consistently rising [2]. In developing countries like India, diabetes is a serious condition that has gained significant attention and is a serious issue. Diabetes is a chronic disease that influences metabolism and occurs when a person's blood sugar levels are insufficient due to either inefficient insulin production or the body's inability to respond correctly to insulin, or both [3]. It is a medical circumstance in which the body cannot appropriately metabolize blood sugar, resulting in high glucose levels in the bloodstream, known as hyperglycemia [4]. Insulin is a hormone that primarily controls glucose absorption from the bloodstream into the body's muscle and fat cells, but not the cells of the central nervous system. Any insulin deficiency or insensitivity in its receptors can lead to various forms of diabetes [5]. Untreated diabetes can lead to various complications and long-term effects, including coma, stroke [6], failure of the kidneys and eyesight, cardiovascular and cerebral vascular dysfunction, diseases of peripheral vascular, difficulty in sexual functioning, joint failure, weight loss, ulcers, negative impacts on immunity [7]. A contributing factor to the worldwide increase in obesity with diabetes is the consumption of high-calorie, high-fat foods, and reduced physical activity due to the convenience of new technologies that offer instant access to entertainment, education, communication [8], and shopping [9].

According to the WHO and the American Diabetes Association (ADA), there are four primary categories of diabetes [10]. **Type 1 Diabetes:** This particular form of diabetes makes up around 5–10% of all cases and is identified by a deficiency in insulin production within the body [11].

Type 2 Diabetes: The most common type of diabetes is this one, accounting for approximately 90% of cases, resulting from a combination of metabolic disorders and insulin resistance.

Gestational Diabetes (GDM): This form of diabetes shares similarities with type 2 diabetes, and during pregnancy, hormonal changes can lead to a type of diabetes that affects around 4% of pregnancies and also the risk of diabetes in newborns.

Rare Specific Diabetes: Several types of diabetes are caused by specific genetic or metabolic disorders.

Analyzing diabetes data can be challenging due to most medical data's nonlinear, abnormal, correlation-structured, and complex nature [12]. When it comes to detecting diabetes, scholars often rely on the Pima Indians Diabetes Database for their studies [13]. Pima County, situated in southern Arizona, is well-known globally for being extensively studied in diabetes research; this phenomenon is not limited to Native Americans but is observed worldwide. The Pima Indian dataset is widely

used and contains information about patients with diabetes and provides information on both positive and negative cases of type 2 diabetes [14].

Each hospital has its own unique set of basic and medical information, and it is important to extract valuable information from this data to aid in future medical analysis and diagnosis. Healthcare studies have recently utilized machine learning [15] techniques to enhance diabetes diagnosis and detection by analyzing clinical data. ML allows computers to learn from experiences and inputs, improving their ability to predict disease outcomes accurately [16]. On the other hand, deep learning [17, 18] is based on how the human brain functions and can resolve complex issues and process enormous amounts of data [19], learn intricate tasks, and handle unlabeled data with minimal human input [20].

Our research primarily centers on the Pima Indians Diabetes Database and the significance of our study:

- To balance the data set for the research, using the data balancing technique.
- Despite utilizing a baseline classifier, several strategies were used to select the classifier's best parameter.
- In this work, improving performance is a key factor. Our study surpassed earlier research to identify diabetes in the Pima Indian dataset.

To achieve our objective, we have separated the article into different sections. We followed a specific methodology to conduct the research explained in Sect. 3. A thorough literature review should be conducted in Sect. 2 to identify any gaps in knowledge. Section 4 will cover the results of our experiments using the models we employed. Finally, we will reveal our discoveries.

2 Related Work

Type 2 diabetes is a widespread and complex illness, especially in tropical regions like India. As a result, scientists have been working diligently to create an ML model that can accurately diagnose this disease. The proposed method involves using Goldberg's Genetic algorithm with Pima Indians Diabetes Dataset [14] during the preprocessing phase, which enhances the accuracy of existing diagnostic techniques. A Multi-Objective Evolutionary Fuzzy Classifier is also utilized in the dataset, focusing on achieving the maximum rate of classifier and minimum rules. During training, the algorithm achieved a classifier rate of 83.0435% with NSGA II, reducing the number of features from 8 to 4 [1]. According to [21, 22] focused on predicting type 2 diabetes (T2DM) using different neural network-based methods. The analysis utilized the Pima Indian Diabetes dataset and two approaches. By utilizing feature selection and feeding the selected features into an MLP neural network classifier, the first approach achieved an accuracy of 85.153% in data recovery. The second approach utilized noise reduction with k-means clustering by feature selection. The resulting features were used with RF, LR, and MLP neural network classifiers; the classifier with the highest accuracy achieved was 77.08%.

An automated diagnosis system was developed by combining the Pima Indians Diabetes Dataset with a Three-Layered Artificial Neural Network (ANN). The logistic activation function was utilized, and the Quasi-Newton method was employed for training, and a logistic activation function was utilized, and the Quasi-Newton method was employed for training. The model's performance was evaluated using a cumulative gain plot, and the maximum gain score was used to assess its quality [3]. The clinical dataset was employed for training and testing using k-fold cross-validation. Performance metrics such as Area Under the Curve (AUC), Classification Accuracy (CA), F1 score, precision, and recall were evaluated for all the algorithms. An open-source platform called Orange 3.24.1 was used to generate the results, which utilizes Python libraries [16]. To predict the presence of diabetes, an ensemble of ML classification algorithms, including LR, KNN, DT, SVM, NB, and RF, were used to combine features extracted from different datasets, and when analyzed using Jupyter Notebook, the experimental according to the results, the RF approach achieved the highest accuracy in classification of 89.58% [7]. Scholars [23] also utilized the Pima Indian Diabetes (PIDD) Dataset to predict the onset of diabetes based on diagnostic methods and gained a 94% accuracy rate with LR, 79% with NB, and 69% with KNN algorithms. Yahyaoui et al. [11] proposed a Decision Support System (DSS) that predicts diabetes using ML techniques and compared conventional ML methods with deep learning approaches. SVM and RF classifiers were used for conventional ML, while a full CNN was utilized for deep learning using the Pima Indians Diabetes Database. The database comprises 768 samples with eight features, of which 500 were labeled non-diabetes and 268 as diabetes. The overall accuracy achieved using DL, SVM, and RF was 76.81%, 65.38%, and 83.67%, respectively.

In predicting diabetes in females of Pima Indians heritage, various supervised learning algorithms and deep learning, including Classification Tree, SVM [5], K-Nearest Neighbors (kNN), NB, RF [9], neural network (NN), AdaBoost (AB) [12], LR, CNN [4], ANN [3] were used. Observing the outcomes, we find that everyone is not accurate as the others. LR produced the best performance with an average accuracy of 94% in predicting diabetes in females of Pima Indian heritage [23].

After thoroughly reviewing the literature, we have identified several limitations, challenges, and shortcomings. Let us now discuss the following:

- The data available for the Pima Indians Diabetes Dataset is quite limited. It has a small sample size, imbalanced classes, missing and outdated data, and few instances. This can lead to a lack of data availability and make it difficult to interpret the model.
- Finding the features that help models learn representations most effectively is the main difficulty in feature engineering. Manually filtering and evaluating each feature in a diabetes dataset could be time-consuming.
- We have discovered a significant fact: The vast majority of papers that use the Pima Indian Diabetes Dataset have not achieved good accuracy (Fig. 1).

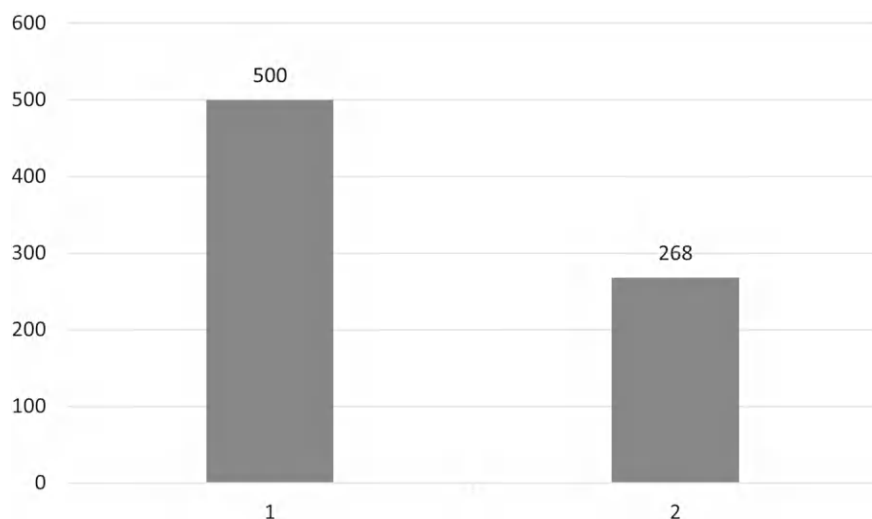


Fig. 1 Number of samples each class before data balancing

2.1 Methodology

2.2 Dataset

This dataset originates from the National Institute of Diabetes and Digestive and Kidney Diseases. It aims to predict whether a patient has diabetes using specific diagnostic measures. The dataset was created by selecting instances from a large database according to strict criteria. All patients included in the dataset are Pima Indian women over 21 who have sought treatment at this facility. The dataset contains one dependent variable (the patient's likelihood of having diabetes) and several independent variables (such as BMI, insulin level, age, and number of previous pregnancies). The target variables in the dataset are 0 and 1. The figure illustrates how many samples are in each class. As we have stated, one of the main problems with this dataset is the imbalanced distribution of classes. Therefore, we used data balancing techniques to make the dataset balanced. Techniques for data balance are covered in Sect. 3.2 (Fig. 2).

2.3 Data Balancing

The model could be biased in favor of the dominant class when the data are unbalanced. This indicates that it may be less accurate at identifying minority classes and more accurate at identifying that specific class of the dominant class. A good Natural

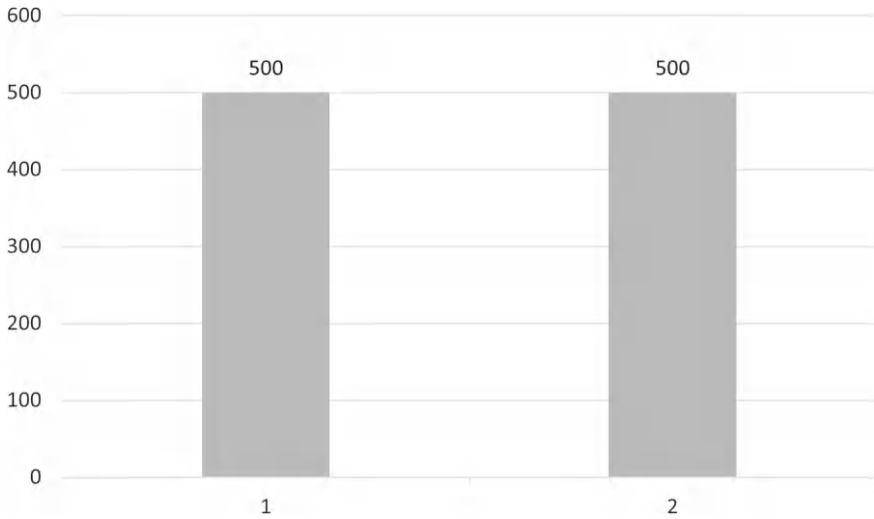


Fig. 2 Number of samples each class after data balancing

Language Processing (NLP) work strategy is data balance. We utilized RandomOverSampler in this instance to balance the dataset. The RandomOverSampler techniques are described below: Examples in the majority class have the label N_0 and examples in the minority class have the label N_1 , respectively. N_0 must equal N_1 for the dataset to be balanced. Calculate the difference between instances of the majority and minority classes after that: $\Delta N = N_0 - N_1$. Choose N occurrences at random with replacement from the minority class: Let M represent the collection of instances that belong to the minority class. Choose N instances at random from M to represent M_{selected} . Create a set of synthetic instances called $M_{\text{synthetic}}$ by duplicating each instance in M_{selected} N times. The original minority class instances (M) and the synthesized instances ($M_{\text{synthetic}}$) are combined to create the balanced dataset, which is then called M_{balanced} . The dataset is rebalanced using the aforementioned random oversampling technique, enabling the algorithms to learn more from both classes and enhance performance on tasks like diabetes diagnosis using the Pima Indians dataset.

2.4 Feature Scaling

In particular, in medical diagnostics like diabetes detection, feature scaling techniques are critical in prepping datasets for machine learning models. The StandardScaler, MinMaxScaler, MaxAbsScaler, RobustScaler, PowerTransformer, QuantileTransformer, Normalizer, and Binarizer, feature scaling approaches were used in this study. Each method has a particular function that enhances model performance in

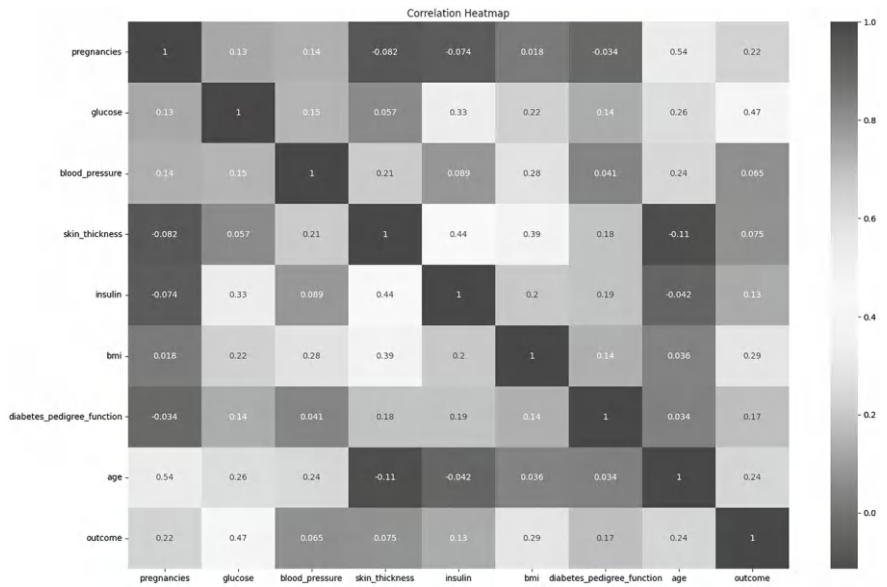


Fig. 3 Correlation matrix for the dataset’s features

diabetes detection. MinMaxScaler, RobustScaler, PowerTransformer, and MaxAbsScaler performed well in our research. MaxAbsScaler scales features to a range between -1 and 1 using the highest absolute value. By maintaining the data’s sign during scaling, this method makes sure that both positive and negative numbers are evenly represented. RobustScaler uses robust statistics to scale features, making it resistant to outliers. RobustScaler enhances resilience to extreme values and permits models to concentrate on the bulk of the data by lowering the impact of outliers. PowerTransformer employs power transformations to enhance the Gaussianity of feature distributions. This method handles the data’s non-normality, which is frequently preferred by many machine learning algorithms. Features are scaled by MinMaxScaler to a defined range, frequently between 0 and 1. MinMaxScaler helps models handle different scales by bringing features into a common range and preventing one characteristic from dominating others. Figure 3 illustrates the correlation matrix for the dataset’s features. By highlighting the strength and direction of their linear association, the correlation matrix sheds new light on the relationships between the various elements in the dataset.

2.5 Classifiers

To forecast the development of diabetes illnesses, we used DT, SVM, LR, and RF classifiers. The classifier is initially used without any parameters. After that, we

offer several parameters to improve the classifier's performance. Each classifier's hyperparameter tuning value and used parameter are mentioned below:

- **DT:** The DT model's tree structure can represent the procedure for classifying instances according to features. This can be seen as a group of rules that follow the "if-then" logic or probability distributions that rely on certain features and categories. The DT is employed in our research for classification [24]. In this classifier, we made use of various parameters. We used the `max_depth` argument first. It regulates the tree's maximum number of tiers. Lower values can result in underfitting, while higher values can lead to overfitting. We test the values 3, 5, and 7 in our example. `min_samples_split` is the minimum number of samples required to split an internal node. It determines the minimum number of samples a node must have before it can be split. In the example, we try values 2, 5, and 10. `Auto`, `sqrt`, and `log2` were the values we used for the `max_features` parameter. `Auto` uses all features, `sqrt` uses the total feature count's square root, and `log2` uses the total feature count's base-2 logarithm. We applied values 1, 2, and 3 for `min_samples_leaf`. Entropy and Gini were employed as criteria to assess a split's quality.
- **LR:** The LR model has been widely used in many fields, including the biological sciences. When categorizing data objects into groups is the goal, the LR approach is used [25]. In LR, the target variable is usually binary, consisting of data classified as 1 or 0, indicating whether a patient is positive or negative for diabetes. Our LR algorithm strives to select the most accurate model that captures the connection between our predictor and target variables, all while ensuring it is diagnostically reasonable. In our study, the values for the penalty parameter were l1 and l2. 'l1' denotes L1 regularization (Lasso), while 'l2' is L2 regularization (Ridge). To prevent overfitting by lessening the influence of individual features, smaller values of C enhance the regularization strength. They were 0.01, 0.1, 1.0, and 10.0. We utilized `liblinear` and `saga` as solvers; `liblinear` is a decent option for small datasets, while `saga` is appropriate for large datasets and can handle both L1 and L2 regularization. Increases in `max_iter` may be required if the algorithm doesn't converge, upping the maximum number of iterations for the solver to reach convergence. When using this parameter, we set this correspondents value to 100, 500, and 1000.
- **SVM:** We rigorously tuned the hyperparameters to enhance the performance of our SVM. SVMs are effective ML models for classification problems, but they perform best when the right hyperparameters are chosen [26]. We used the well-known `GridSearchCV` method, which thoroughly searches a grid of predetermined hyperparameters. The regularization parameter C, the kernel type kernel, and the kernel coefficient gamma were the three main hyperparameters that were the subject of our inquiry. To find the best setting for each hyperparameter that maximizes the accuracy of the classifier, we looked at a range of possible values. A tenfold cross-validation approach was used in the grid search to assess the model's performance thoroughly. `GridSearchCV` determined that the optimal hyperparameters were the `C = 1`, `kernel = "rbf"`, and `gamma = "scale"`. These hyperparameters

produced the test set's best accuracy and outperformed other setups in terms of performance. The SVM classifier's performance is crucially boosted by our hyperparameter tuning procedure, which guarantees that it produces trustworthy and accurate detections for our diabetes classification assignment.

- **RF:** We used a thorough hyperparameter tuning method to enhance the RF classifier's detection capabilities. Multiple Decision Trees are used in RF, an ensemble learning technique, to increase detection accuracy. To systematically explore the hyperparameter space in our hyperparameter tuning, we used the GridSearchCV approach. The primary hyperparameters on which we concentrated were (`n_estimators`), (`max_depth`), (`min_samples_split`), and (`max_features`). To robustly assess performance, we applied tenfold cross-validation. Following a thorough investigation, it was determined that the optimal set of hyperparameters to use was `n_estimators = 500`, `max_depth = 10`, `min_samples_split = 2`, and `max_features = "sqrt."` This means we used a forest of 500 trees, each with a maximum depth of 10, for our RF classifier. A minimum of two samples were needed to decide whether to split an internal node, and the algorithm took the square root of the total number of characteristics into account when determining the appropriate split. In comparison with other settings, we improved the accuracy of our diabetes classification task using this hyperparameter tuning approach. Utilizing these particular hyperparameter settings, the optimized RF model shows promise in predicting diabetes using the ensemble of Decision Trees.

3 Result and Discussion

We initially looked at four evaluation criteria to gauge how well the applied model, feature scaling, and data balancing strategies performed. Precision (`prc.`), Recall (`Rce.`), *F*-score, and Accuracy (`Acc.`) have all been calculated. Different classifiers, feature scaling, and data balance techniques have been used. So, to evaluate the entire experiment, three research questions (RQ) must be considered. We employed macro-averaging, which averages data.

RQ1: What is the performance after data balancing?

RQ2: What are the outcomes of different feature scaling techniques applied?

RQ3: How much has the performance increased after hyperparameter tuning the classifier?

RQ1: Role of Data Balancing Technique

The results of data balancing approaches are displayed in Figs. 4, 5, 6, and 7. In this section, we want to emphasize the value of data balancing strategies, so we chose the best results from the implemented feature scaling techniques. The precision, recall, *F*-score, and accuracy metrics may vary when dealing with imbalanced datasets before and after performing data balancing methods. Without data balancing, a significant class imbalance can cause biased metrics toward the majority class.

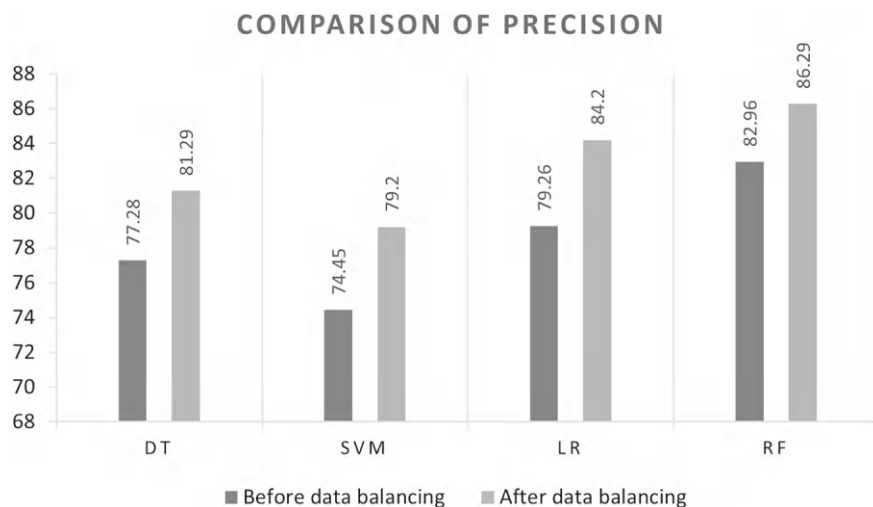


Fig. 4 Comparison of precision value

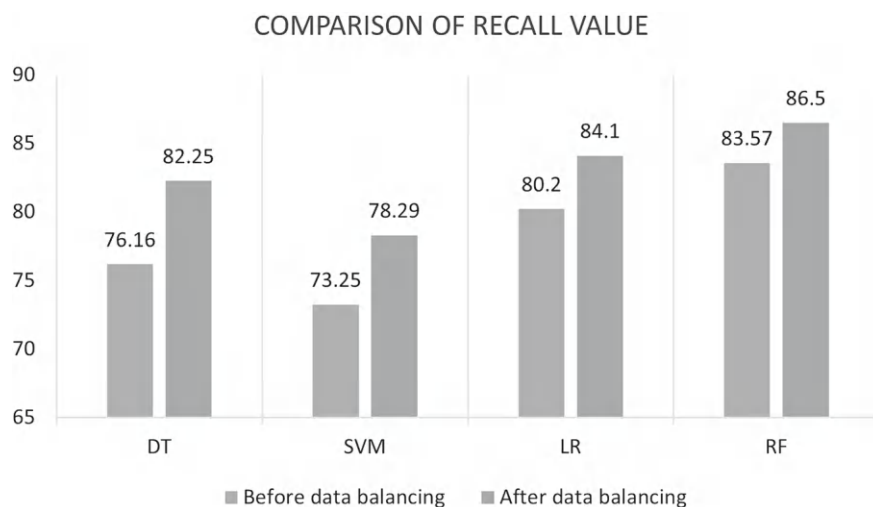


Fig. 5 Comparison of recall value

- **Precision:** Before balancing, the precision for the minority class may be low because the classifier tends to predict the majority class more often. However, balancing can improve precision for the minority class by helping the classifier to identify positive samples more accurately. We noticed significant improvements in multiple algorithms when comparing the precision rates before and after data balancing. The DT algorithm had a precision rate of only 77.20% before the data was balanced, but after balancing the data, its precision rate increased significantly

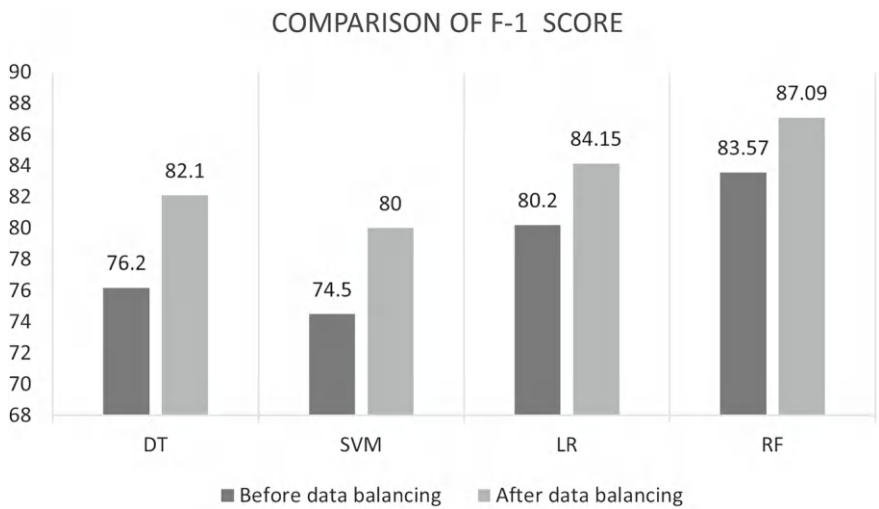


Fig. 6 Comparison of $F1$ -score

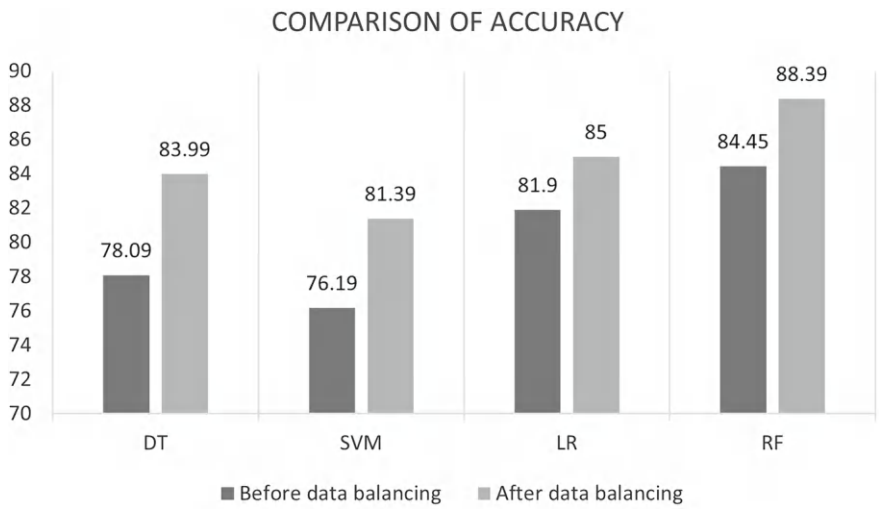


Fig. 7 Comparison of accuracy value

- to 81.29%. After balancing the data, the precision rates for SVM, LR, and RF algorithms increased to 4.75, 4.94, and 3.33, respectively.
- **Recall:** Before balancing, the minority class may have a low recall as the classifier might overlook positive samples while prioritizing the majority class. However, after balancing, the recall can increase for the minority class as the classifier becomes more attuned to positive samples. We observed notable enhancements

in various algorithms when comparing their recall values before and after implementing data balancing. Before balancing, the DT algorithm exhibited a recall value of 76.16%, but after balancing the data, its recall rate significantly improved to 82.25%. After the data was balanced, the recall rates for SVM, LR, and RF algorithms increased to 5.04, 3.9, and 2.93, respectively.

- **F-score:** The *F*-score measures the classifier’s performance that considers precision and recall and is calculated as the mean of these two values. However, the *F*-score for the minority class may be low initially due to low precision and recall. By balancing these factors, the *F*-score can improve, leading to better performance for the minority class. We have marked several algorithm’s significant enhancements when comparing the *F*-1 score rates before and after data balancing. The DT algorithm exhibited a *F*-1 value of 76.2% before the data balancing and increased significantly to 82.1% after the data was balanced. The precision rates for SVM, LR, and RF algorithms increased after the data was balanced, and rates are now 5.5, 3.95, and 3.52, respectively.
- **Accuracy:** Accuracy can be high before balancing because of the dominant majority class, but once balancing is done, the accuracy may slightly decrease as the classifier becomes more sensitive to the minority class and correctly identifies positive samples. We noticed significant improvements in the algorithm’s accuracy when we compared it before and after data balancing. The DT algorithm had an initial accuracy rate of 78.09%, but after balancing the data, it increased significantly to 83.99%. After balancing the data, the precision rates of SVM, LR, and RF algorithms have increased. Currently, the rates are 5.2, 3.1, and 3.94, respectively.

RQ2: Role of Feature Scaling Technique

Even though we used nine feature scaling strategies, Table 1 only shows the top four results. To identify diabetes, four classifiers were utilized. So, to grasp the significance of feature scaling, we have only presented the top four techniques that our four classifiers have been able to achieve.

From Table 1 MaxAbsScaler, one of nine feature scaling methods, produced the best results for the Decision Tree classifier. It achieved 83.99% acc., an *F*1-score of 82.16, 81.29% prc. and 82.25% rec. This demonstrates that the Decision Tree classifier’s overall model performance was enhanced by scaling the features using MaxAbsScaler. On the other hand, PowerTransformer gives the best results for the SVM classifier among the feature scaling strategies used. It obtained an *F*1-score

Table 1 Model performance with different scalers

Classifier	Scaler	Prc	Rec	<i>F</i> 1-score	Acc
DT	MaxAbsscaler	81.29	82.25	82.16	83.99
SVM	PowerTransformer	79.20	78.29	80	81.39
LR	MinMaxscaler	84.20	84.10	84.15	85
RF	Robustscaler	86.29	86.50	87.09	88.59

of 80.00%, an acc. of 81.39%, a prc. of 79.20%, and a rec. of 78.29%. The SVM classifier performed better when features were scaled using PowerTransformer. The MinMaxScaler algorithm generated the best classification results for the LR classifier. It earned a pre. of 84.20%, a rec. of 84.10%, an *F1*-score of 84.15, and an acc. of 85.00%. The characteristics were scaled using MinMaxScaler, which enhanced the accuracy and performance of the LR classifier. The scaling method used by the RF classifier that performed the best was RobustScaler. It earned a pre. of 86.29%, rec. of 86.50%, *F1*-score of 87.09, and acc. of 88.50%. Using RobustScaler for feature scaling significantly improved model performance and accuracy for the RF classifier. The experiment demonstrates that different classifiers perform better with different feature selections. Thus, selecting the best approach for a given classifier is possible by utilizing many feature scaling strategies.

RQ3: How Much Has the Performance Increased After Hyperparameter Tuning the Classifier?

To tune the hyperparameters, we used GridSearchCV. Initially, we used the classifier and attempted to obtain a result. RQ2 describes the results of this approach. Then, to improve the model’s accuracy, we used several parameters. Hyperparameter tuning is used to select the best. The accuracy of each classifier is then determined once more, and the outcome is compared to RQ2. We saw that the outcome has increased greatly. The result is then cross-validated. Cross-validation simulates the process of training and testing on various subsets of the available data to determine how well a model is expected to perform on fresh, unknown data. It offers a more trustworthy assessment of the model’s performance. Table 2 shows the performance after hyperparameter tuning the classifier. Based on the findings of our study from Table 2, we found that hyperparameter adjustment significantly improved the classifiers’ accuracy. The accuracy of the DT classifier increased significantly from its prior accuracy of 83.99% to 85.98%. Similarly, the SVM showed an accuracy of 83.74%, up from the prior accuracy of 81.39%. Additionally, the LR classifier maintained the same accuracy as the earlier result, displaying an accuracy of 88.25%. The RF classifier also attained an accuracy of 92.17%, reiterating its excellent accuracy from the prior outcome. These results demonstrate how hyperparameter adjustment can improve classifier accuracy for identifying diabetes illness.

Table 2 Accuracy of the classifiers after hyperparameter tuning

Classifier	Accuracy	Cross-validation (tenfold)
DT	85.98	84.54
SVM	83.74	82.90
LR	88.25	88.10
RF	92.17	92.10

4 Conclusion

Diabetes is a major medical issue that causes a significant number of deaths every year worldwide. It is a common and chronic medical disease that can cause serious complications and place considerable pressure on the healthcare system if left untreated or insufficiently managed. Complications that may arise from this condition incorporate blindness, heart disorder, kidney downfall, and nerve impairment that can significantly impact the heart's performance. Detecting diabetes in the early stage is essential to reduce health risks, and by allowing for timely intervention and management strategies, blood sugar levels can be regulated, and the likelihood of complications can be reduced.

In this paper, we have detected diabetes by taking the female Pima Indians Diabetes Dataset. We examined four evaluation criteria: Precision, recall, *F*-score, and accuracy to measure the effectiveness of our applied model, feature scaling, and data balancing strategies. Various classifiers, feature scaling methods, and data balancing techniques were utilized in the analysis.

We compared four ML algorithms—DT, SVM, LR, and RF—to detect diabetes. The findings revealed that RF was the most efficient in classifying diabetes, with an overall accuracy rate of 92.17% across all rounds of experiments. We then applied cross-validation to verify the results. This method involves training and testing the model on various subsets of data to assess its effectiveness on new and unfamiliar data.

References

1. Jaganathan SCB, Dhasarathan C, Harshavardhan I, Harisrujan C, Haridhar G, Prasad GG, Kannadhasan S (2022) Machine learning for smartphone-based early detection of diabetic disease in pima Indians diabetes database. *J Algebr Stat* 13(1):780–796
2. Mercaldo F, Nardone V, Santone A (2017) Diabetes mellitus affected patients classification and diagnosis through machine learning techniques. *Procedia Comput Sci* 112:2519–2528
3. Lakhwani K, Bhargava S, Hiran KK, Bunde MM, Somwanshi D (2020) Prediction of the onset of diabetes using artificial neural network and pima Indians diabetes dataset. In: 2020 5th IEEE international conference on recent advances and innovations in engineering (ICRAIE). IEEE, 2020, pp 1–6
4. Swapna G, Vinayakumar R, Soman K (2018) Diabetes detection using deep learning algorithms. *ICT express* 4(4):243–246
5. Zolfaghari R (2012) Diagnosis of diabetes in female population of Pima Indian heritage with ensemble of BP neural network and SVM. *Int J Comput Eng Manag* 15(4):2230–2893
6. García-Ordás MT, Benavides C, Benítez-Andrades JA, Alaiz-Moretón H, García-Rodríguez I (2021) Diabetes detection using deep learning techniques with oversampling and feature augmentation. *Comput Methods Programs Biomed* 202:105968
7. Vaishali R, Sasikala R, Ramasubbareddy S, Remya S, Nalluri S (2017) Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians diabetes dataset. In: 2017 International conference on computing networking and informatics (ICCN). IEEE, 2017, pp 1–5

8. Kumar M, Rani Z (2019) Energy-efficient scheduling of cloud data center servers, p 1769–1772 (Online). Available: <https://doi.org/10.35940/ijitee.k1759.0981119>
9. Abdulhadi N, Al-Mousa A (2021) Diabetes detection using machine learning classification methods. In: 2021 International conference on information technology (ICIT). IEEE, 2021, pp 350–354
10. Abedini M, Bijari A, Baniroostam T (2020) Classification of Pima Indian diabetes dataset using ensemble of decision tree, logistic regression and neural network. *Int J Adv Res Comput Commun Eng* 9(7):7–10
11. Yahyaoui A, Jamil A, Rasheed J, Yesiltepe M (2019) A decision support system for diabetes prediction using machine learning and deep learning techniques. In: 2019 1st International informatics and software engineering conference (UBMYK). IEEE, 2019, pp 1–4
12. Farajollahi B, Mehmannaavaz M, Mehrjoo H, Moghbeli F, Sayadi MJ (2021) Diabetes diagnosis using machine learning. *Front Health Inform* 10(1):65
13. Zehra A, Asmawaty T, Aznan M (2014) A comparative study on the pre-processing and mining of Pima Indian diabetes dataset. In: 3rd International conference on software engineering & computer systems, 2014, pp 1–10
14. Rajni R, Amandeep A (2019) Rb-bayes algorithm for the prediction of diabetic in Pima Indian dataset. *Int J Electr Comput Eng* 9(6):4866
15. Rudra Kumar M, Gunjan VK (2022) Peer level credit rating: an extended plugin for credit scoring framework. In: ICCCE 2021: proceedings of the 4th international conference on communications and cyber physical engineering. Springer, pp 1227–1237
16. Bhoi SK et al (2021) Prediction of diabetes in females of Pima Indian heritage: a complete supervised learning approach. *Turk J Comput Math Educ (TURCO-MAT)* 12(10):3074–3084
17. Zhu T, Li K, Herrero P, Georgiou P (2020) Deep learning for diabetes: a systematic review. *IEEE J Biomed Health Inform* 25(7):2744–2757
18. Dwaram JR, Madapuri RK (2022) Crop yield forecasting by long short-term memory network with Adam optimizer and Huber loss function in Andhra Pradesh, India. *Concurr Comput: Pract Exp* 34(27):e7310
19. Prasad M, Babu AR, Krishna AV, Kumar MR (2019) Reciprocal repository for decisive data access in disruption tolerant networks. *Int J Innov Technol Explor Eng* 9(1):4430–4434
20. Kamble MTP, Patil S (2016) Diabetes detection using deep learning approach. *Int J Innov Res Sci Technol* 2(12):342–349
21. Rubaiat SY, Rahman MM, Hasan MK (2018) Important feature selection & accuracy comparisons of different machine learning models for early diabetes detection. In: 2018 International conference on innovation in engineering and technology (ICIET). IEEE, 2018, pp 1–6
22. Kumar V, Kumar MR, Shribala N, Singh N, Gunjan VK, Arif M et al (2022) Dynamic wavelength scheduling by multiobjectives in OBS networks. *J Math* 2022
23. Khaleel FA, Al-Bakry AM (2023) Diagnosis of diabetes using machine learning algorithms. *Mater Today Proc* 80:3200–3203
24. Zou Q, Qu K, Luo Y, Yin D, Ju Y, Tang H (2018) Predicting diabetes mellitus with machine learning techniques. *Front Genet* 9:515
25. Zhu C, Idemudia CU, Feng W (2019) Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques. *Inform Med Unlocked* 17:100179
26. Mohan N, Jain V (2020) Performance analysis of support vector machine in diabetes prediction. In: 2020 4th International conference on electronics, communication and aerospace technology (ICECA), 2020, pp 1–3

Comparative Analysis of Machine Learning Classifiers for Football Fake News Detection: A TextRNN-Based Approach



V. Prema Manvi, U. Saipriya, Shaik Suhana, S. Firoz Begum, and S. Abhitha Angel

Abstract Communication between people has significantly improved recently because of the increase in online social media. People communicate, share information, and consume news through social networking platforms like Twitter, Facebook, WhatsApp, etc. Recent content shared on social media is frequently questionable and, in some cases, purposely misleading, and the term “fake news” is frequently used to describe this material. False news being widely disseminated online has the potential to be harmful to society. Fake news detection has become a crucial area of research that objective to identify whether a particular content is legitimate or fake. This paper presents an innovative approach for identifying fake information in football news that combines regular machine learning (ML) and deep learning (DL) approaches. The ML approach utilizes four models: Decision tree (DT), Random Forest (RF), Support Vector Machine (SVM), and Naive Bayes (NB) for training and evaluation. Different n-gram sizes are employed with TF-ID feature extraction to construct matrix features; also, grid search optimizes ML models through cross-validation. On the other hand, the DL approach employs the TextRNN model, which combines an Embedding layer and an RNN layer to do text classification tasks to evaluate the model. The TextRNN model accurately classified fake football news and achieved the highest accuracy, with a recall of 94.50%, precision of 94.25%, and an outstanding F-score of 94.35%. It excelled in the fake news data and proved its superiority over other ML classifiers. In contrast, the ML models showed good performance, and the Random Forest (RF) model acquired the best precision of 92.87%, recall of 92.10%, and F-score of 92.65%, among others. Due to the lack of study on football false news, it may be inferred from the experiment findings that the suggested method is superior to conventional methods.

Identify applicable funding agency here. If none, delete this.

Note: Sub-titles are not captured in Xplore and should not be used.

V. Prema Manvi (✉) · U. Saipriya · S. Suhana · S. Firoz Begum · S. Abhitha Angel
Computer Science and Engineering, Ravindra College of Engineering for Women, Kurnool, India
e-mail: premacse@recw.ac.in

Keywords Football fake news detection · Machine learning · Deep learning · TextRNN · NLP

1 Introduction

The term “fake news” indicates yellow journalism that deliberately disseminates false information or hoaxes through traditional print media and modern social media platforms [1]. The spread of false information is considered a grave danger to democracy, journalism, and freedom of expression; also, it has eroded public confidence in government institutions. Fake news is typically created and distributed to harm the reputation of an agency, entity, or individual [2]. The internet has revolutionized [3] how people interact and communicate by providing low-cost, effortless access and fast information dissemination. As a result, social media and online portals have become increasingly popular for news searches and reading, replacing traditional newspapers for many individuals [4]. The growing use of social media [5] significantly affects society, business, and culture, with positive and negative consequences. The rise of web information has led to increased fake news, rumors, incorrect information, and misinformation. Social media networks, blogs, Twitter [6], Facebook [7], Chinese Sina Weibo [8], and other platforms have contributed to the growth of information in recent years. Detecting fake news and rumors promptly is crucial to prevent societal harm. This issue has gained global attention due to its impact on recent politics and its adverse effects [9].

The impact of false information was notably exposed during the 2016 campaign for the United States presidency [10]. According to research, fake news on Twitter [11] spreads far more quickly and receives a much bigger number of retweets, especially regarding political news. According to estimates from Stanford University researchers, official and online social media platforms are the source of 72.3% of fake news [12]. Relying solely on news content to detect fake news is often insufficient, and additional information, such as user engagement on social media, is necessary [13]. Fake news detection has become important in the news industry and society, and information credibility is a persistent problem affecting printed and digital media [14]. Several reputable websites, including PolitiFact, Snopes, Jaachai, and FactCheck, are dedicated to uncovering and publishing fake news daily across various topics [15].

Moreover, ML-based solutions [16] can be useful in completing the task [17]. Most studies on identifying fake news used supervised techniques; although these methods have shown potential, they have a significant drawback—they depend on an extensively annotated dataset to train the classifier. Gathering annotations is tedious and requires careful verification of content and cross-checking with reliable sources, making it time-consuming and labor-intensive [18]. There have been several studies conducted on the utilization of commonly used multiple ML and DL [19] models for detecting fake news, including SVM, Long Short-Term Memory (LSTM) [20], Recurrent Neural Networks (RNN) [21], NB, Convolutional Neural

Networks (CNNs), RF, DT, and Logistic Regression (LR), Multi-layer Perceptron Model (MLP) [9], Gated Recurrent Neural Network (GRNN) [22] along with Graph Neural Networks (GNN) [FakeNewsNet].

Our main research contributions focus on detecting fake news football and the significance of our study:

- We propose a model for detecting fake football news, including fake football news, which is an important and challenging task in today's digital age. Upon thorough review of scholarly works about fake news, it has come to our attention that studies have yet to be conducted on fake football news. Many techniques and approaches used to identify fake news can also be applied to fake football news.
- We have proposed here a novel TextRNN model.
- With GridSearch optimization using the hybrid feature engineering technique, several baseline classifiers are utilized.

To structure this paper, we have divided it into several sections. Firstly, in Sect. 2, we will review the various approaches to detecting fake news. Following this, we will outline our methodology statement in Sect. 3. Section 4 will present the experimental outcomes of the models we have used. Finally, we will conclude by presenting our findings.

2 Related Work

This section will briefly overview two aspects related to detecting fake news. To begin with, we would like to present some of the latest work on detecting fake news on social media. Secondly, we will discuss various domain-specific approaches to identifying fake news.

Readers highly value social media platforms as they provide an easy way for users to discuss and express their thoughts on various topics and allow them to send messages worldwide.

Numerous studies have made significant strides in detecting and categorizing fake and misleading content using NLP, ML [16], and DL techniques. Gupta et al. [23] focused on identifying hostile and fake content in Devanagari (Hindi) and English, achieving a remarkable F1-score of 0.97 for Hindi hostility detection combined with abusive language detection with features extracted through Hindi BERT, FastText models, and metadata and for English fake news related to Covid-19, utilized entity information from tweets and word embeddings, resulting in a promising F1-score of 0.93. Additionally, scholars [24] developed a framework that connects with CNNs and LSTMs to identify and categorize fake news messages on Twitter, achieving an accuracy rate 82% through DL techniques. Another study [25] introduced LR and Boolean crowdsourcing algorithms as classification strategies that used a dataset of Facebook posts, achieving classification accuracies of 99%.

Several studies have made significant contributions to detecting and understanding the phenomenon of fake political news and so on. SEDDARI et al. [26] created a

new system that combined linguistic and knowledge-based methods to identify fake news. By incorporating eight features, including unique fact-verification features, the system gained an impressive accuracy of 94.4% on a fake news dataset. This outperforms systems that only use linguistic or fact-verification features, which achieved 89.4% and 81.2% accuracy, respectively. Another study [27] introduced an ensemble voting model combining ML and DL algorithms to effectively detect fake news in online text. The model incorporated techniques like CNN, GRU, and RF and achieved an accuracy rate of 0.410 on the LIAR dataset. Furthermore, scholars [28] investigated the influence of personality traits and news consumption on political news discernment. The findings revealed that individuals with higher levels of agreeableness, conscientiousness, and open-mindedness, along with lower levels of extraversion and reduced news consumption, were more capable of discerning political news. Participants also tended to favor headlines aligned with their political beliefs, corresponding to consuming news from biased sources.

3 Methodology

As shown in Fig. 1, our research involves several main steps, including data collection of fake news, performing text preprocessing, splitting the dataset, using feature extraction methods, training and optimizing the model, and evaluating it. Each of the steps is described below:

3.1 *Dataset Description and Preprocessing*

The Kaggle website is where the dataset was gathered. The total data in the dataset is 40826. Only one independent variable exists in the dataset. The dataset has two classes: Real and fake, where real is represented by 1 and fake by 0. There are 21,245 real samples in the dataset, whereas the remaining samples (19,581) are fraudulent. We had to preprocess the dataset after obtaining it from this public website because the raw dataset contained several URLs, unusual characters, and many other things. Here is a list of the preprocessing methods we used on this dataset:

- **Data Cleaning:** We performed text cleaning on the raw dataset during the data preprocessing step of our research project to eliminate extraneous items like numerals, URLs, and special characters. We used Python's `re` module's robust regular expression features to do this. Using these methods, we could exclude numerical values from the text and ensure they wouldn't affect our analysis later. We also removed any URLs from the dataset because we knew they could generate noise and tamper with the validity of our findings. Finally, we were able to filter out special characters, successfully eliminating any potential impact they may have had on the text's quality and coherence.

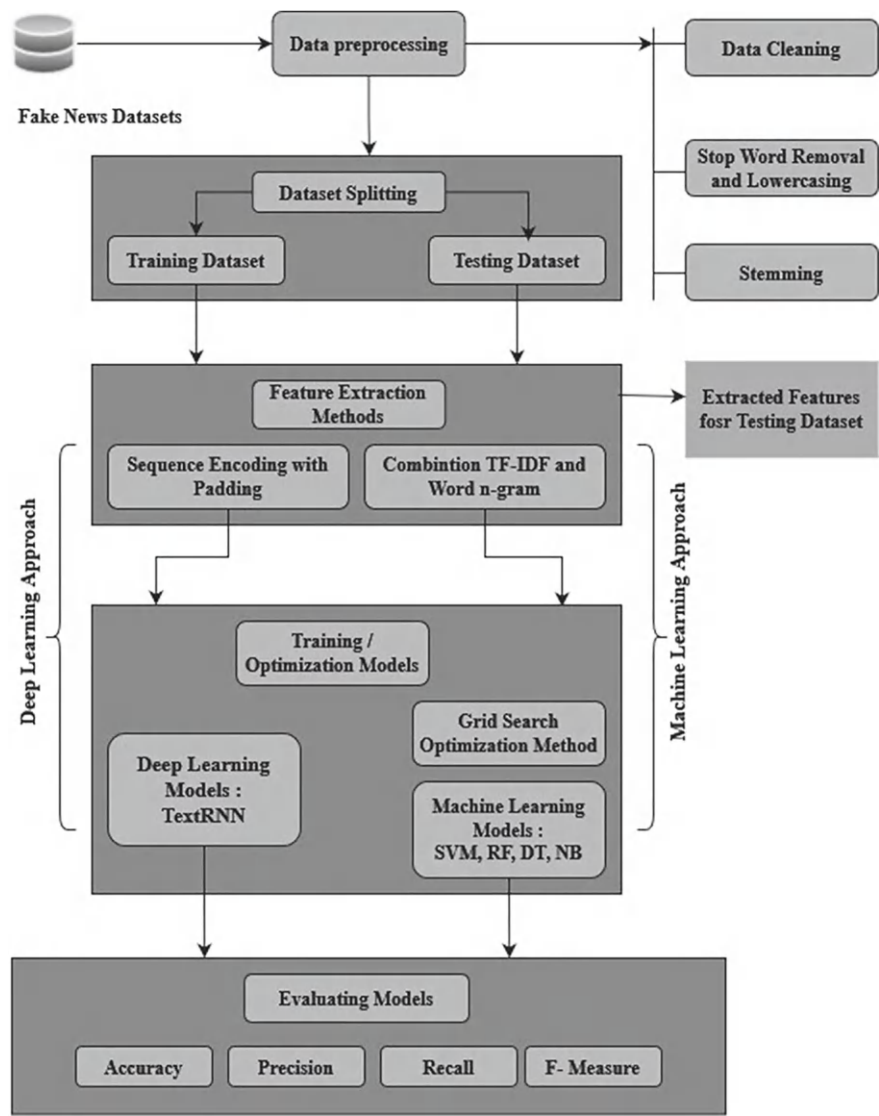


Fig. 1 Overview of our research

- **Stop Word Removal and Lowercasing:** We understood the need to eliminate often-used words like articles, prepositions, and conjunctions because they frequently don't contribute much to the text's overall meaning. We systematically removed these words from our dataset using a predefined list of stop words and the Python NLTK toolkit, resulting in a dataset with a higher proportion of informative and relevant keywords. Additionally, we converted all of the text's characters to their lowercase equivalents using lowercase normalization. This process helps

to uniformize the data and lessen potential inconsistencies brought on by variations in letter cases. We improved our dataset using these strategies, allowing for a more precise and narrowly focused analysis for our research.

- **Stemming:** We performed stemming to the text data using the Snowball Stemmer algorithm, which is included in Python's NLTK package. By removing suffixes or other affixes, stemming involves returning words to their original form. By combining word variants into a single representation, this method made it easier for us to examine the data.

3.2 Feature Engineering

Here, we applied two FE methods. One for the ML model and one for the DL model. We combine TF-IDF and word n-grams in the ML model to translate the data into an appropriate numerical format. We employed Text-to-Sequence Conversion with Padding for the DL model. Below, we have outlined our FE methods:

- **Combination TF-IDF and Word n-gram:** The relative weights of the variables were calculated using the TF-IDF method. Each phrase is both inside and across individual documents in the corpus. TF-IDF enabled us to highlight terms that were both frequent in a particular document and relatively uncommon across the entire dataset by considering both the term and inverse document frequency. Using this method, we found important terms that could add to the analysis's distinctive qualities. Word n-grams were also included, with a concentration on trigrams specifically. Trigrams are contiguous sequences of three words. We tried to catch the contextual dependencies and connections among words in a document by considering trigrams. We attempted to utilize local and global information in our feature representation by integrating TF-IDF and trigrams [29]. Trigrams improved our comprehension of word context and phrase-level linkages, whereas TF-IDF measured term relevance. With this combined feature engineering technique, we could represent the textual data more thoroughly and informally, facilitating the analysis and ML tasks that followed in our research.
- **Sequence Encoding with Padding:** We used a feature extraction technique to transform unstructured text data into numerical representations for our football-related fake news identification study. In particular, the textual information was processed using the Keras Tokenizer and sequence padding techniques. We selected a maximum vocabulary size of 10,000 words and a maximum sequence length of 200 tokens to guarantee the effectiveness and efficiency of our method. We did this in an effort to preserve a simple and consistent input structure while capturing the most pertinent and educational portions of the news stories about football. Using the tokenizer, we created a vocabulary based on the frequency of words in the corpus and gave each word a distinct index. The texts to sequences function of the tokenizer was used. We used the pad sequences method to apply padding or truncation to account for changes in sequence lengths.

Table 1 Hyperparameters of RF classifier

Parameter	Value
n estimators	100, 200, 300
max depth	none, 5, 10
min samples split	2, 5, 10
min samples leaf	1, 2, 4

3.3 Model

For our work, we used several ML classifiers and the TextRNN model, which combines an Embedding layer and an RNN layer to do text classification tasks. The proposed TextRNN model’s architecture and the classifier employed in our research are detailed below:

- **ML Model:** In our work, RF classifier and grid search optimization were used to improve the efficiency and precision of our classification work. A potent ensemble learning technique, the RF model, combines several decision trees to produce predictions [30]. The following hyperparameters comprised the RF classifier’s parameter grid in our study. The maximum depth of each decision tree, max depth: [None, 5, 10], and [100, 200, 300] used as n estimators are the number of trees in the forest. “None” denotes the absence of a maximum depth limit. The variable min samples split: [2, 5, 10] denote the minimum number of samples needed to separate an internal node during tree construction. Minimum Leaf Samples: [1, 2, 4]. By building this parameter grid, we rigorously investigated different combinations of these hyperparameters and assessed their effects on the model’s performance. The RF classifier class from scikit-learn was used to create the actual RF classifier. With the RF classifier instance and the parameter grid as parameters, we used the GridSearchCV function. The hyperparameters used in the RF classifier are shown in Table 1.

In our study, the DT classifier was used in conjunction with grid search optimization to increase the efficiency and precision of our classification task. The DT model is a well-known ML technique that builds a tree-like model based on input features to make judgments [31]. We used the Python scikit-learn toolkit, notably the GridSearchCV function, to enhance the DT classifier. Using this function, we thoroughly searched a specified parameter grid to identify the ideal set of hyperparameters for our DT model. Table 2 displays the hyperparameters applied to the DT classifier. Our study also used grid search optimization to enhance the SVM classifier’s performance, a well-liked ML algorithm frequently used for classification applications. The SVM method is renowned for handling complicated datasets well and detecting non-linear correlations [32]. Python scikit-learn toolkit, notably the GridSearchCV function, is also used here to optimize the SVM classifier. With this function’s help, we thoroughly searched a predetermined parameter grid to find our SVM model’s ideal set of hyperparameters. By creating this parameter grid, we could methodically

Table 2 Hyperparameters of DT classifier

Parameter	Value
criterion	Gini, entropy
max depth	none, 5, 10
min samples split	2, 5, 10
min samples leaf	1, 2, 4

Table 3 Hyperparameters of SVM classifier

Parameter	Value
c	0.1,1,10
kernel	linear, rbf, poly
gamma	scale, auto

investigate different hyperparameter combinations and assess how they affected the effectiveness of the SVM classifier. Table 3 summarizes the SVM classifier’s hyperparameters. The parameters and values for the NB classifier were smoothing [1e-9, 1e-8, 1e-7].

- **Proposed TextRNN Model:** To capture the sequential dependencies in the text data, we have included an RNN [33] layer in the architecture. Embedding layer maps the input tokens to dense vectors with fixed dimensions. It enables the model to learn accurate word representations based on context. The RNN layer, which comes after the Embedding layer, does sequential modeling by keeping track of the relationships between the words in the input sequence and maintaining an internal memory state. Thirty-two SimpleRNN units make up this layer. After the RNN layer, the GlobalAveragePooling1D layer is added to compile the data across the temporal dimension into a fixed-length vector representation. In the dense layer, sigmoid activation function is applied as the final step in binary classification. The RNN layer’s inclusion in the design allows the model to capture the sequential dependencies in the text data successfully. The RNN layer’s learned features are summarized by the Global-AveragePooling1D layer, enabling effective classification. The binary cross-entropy loss function is used in the model’s construction, the Adam optimizer is used for optimization, and accuracy is used as the evaluation metric. An overview of the layers, their settings, and the total number of parameters is given in Table 4, which makes it easier to comprehend the design and complexity of the architecture. Figure 2 depicts the suggested model.

4 Result and Analysis

Various assessment criteria are used to assess the proposed model and the implemented ML classifier. To assess the results, we calculate the Precision (Prc), Recall (Rec), F-score, and Accuracy (Acc). We will go into depth about the results in this

Table 4 Parameters of the proposed TextRNN model

Layer	Parameter	Value
Embedding	Input dimension	max_words
	Output dimension	16
	Input sequence length	max_len
SimpleRNN	Number of units	32
Dense	Units in the output layer	1
	Activation function	Sigmoid

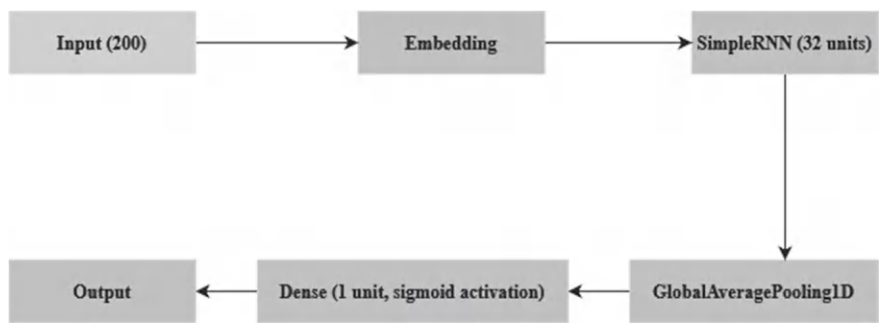


Fig. 2 Proposed TextRNN architecture

part. They will be presented in two sections. We will first talk about the performance of the ML classifier, and then we will talk about the outcome of the suggested model. Finally, we will compare the Acc. of the classifiers also the proposed model.

• **ML Classifier Performance**

In our investigation into identifying false news, we assessed the effectiveness of four ML classifiers: DT, SVM, NB, and RF. Table 5 displays the results. Out of all the news stories, the model categorized as fake, 90.13% were, in fact, fake, according to the DT classifier’s Prc. score of 90.13%. The model seems to effectively identify 91.24% of the actual false news stories, according to the Rec. score of 91.24%. The harmonic mean of the two metrics, the F-score of 91.50%, represents a balanced measurement of precision and recall. With a Prc. of 88.76%, the SVM classifier successfully classified bogus news articles with a high rate. The Rec. score of 87.64% indicates that the model correctly identified 87.64% of the occurrences of bogus news. According to the F-score of 89.12%, which accounts for false positives and false negatives, there is a good balance between precision and recall. The NB classifier’s Prc. of 85.56% indicates that it is reasonably accurate at spotting false news. The Rec. score of 86.32% indicates that 86.32% of the actual cases of fake news were effectively captured by the model. The classifier’s overall performance is evaluated using the F-score of 86.10%. With a Prc. of 92.87%, the RF classifier performed admirably, demonstrating high performance in recognizing false news stories. The

Table 5 Precision, Recall, F-score of ML classifiers

Classifier	Prc (%)	Rec (%)	F-score (%)
DT	90.13	91.24	91.50
SVM	88.76	87.64	89.12
NB	85.56	86.32	86.10
RF	92.87	92.10	92.65

model appears to have successfully captured 92.10% of the real occurrences of bogus news, according to the Rec. score of 92.10%. The F-score of 92.65% demonstrates the RF classifier’s balanced performance in terms of Rec. and Prc. The RF classifier performed better than the other classifiers, attaining the best precision, recall, and F-score. The DT classifier followed closely behind the RF classifier in terms of performance. While the NB classifier scored comparatively lower precision, recall, and F-score than the DT and RF classifiers, SVM showed slightly worse results. The performance is shown graphically in Fig. 3.

• **Performance of Proposed Model**

We list the metrics for the TextRNN model’s false news detection evaluation in Table 6. These measurements demonstrate how well the TextRNN model detects instances of fake news. A Prc. of 94.25% means that the model correctly identified 94.25% of the news stories as fraudulent. The TextRNN model was able to accurately identify 94.50% of the actual fake news stories, according to the Rec. score of 94.50%. F-score, which is 94.35%, gives a fair assessment of the model’s performance (Table 6).

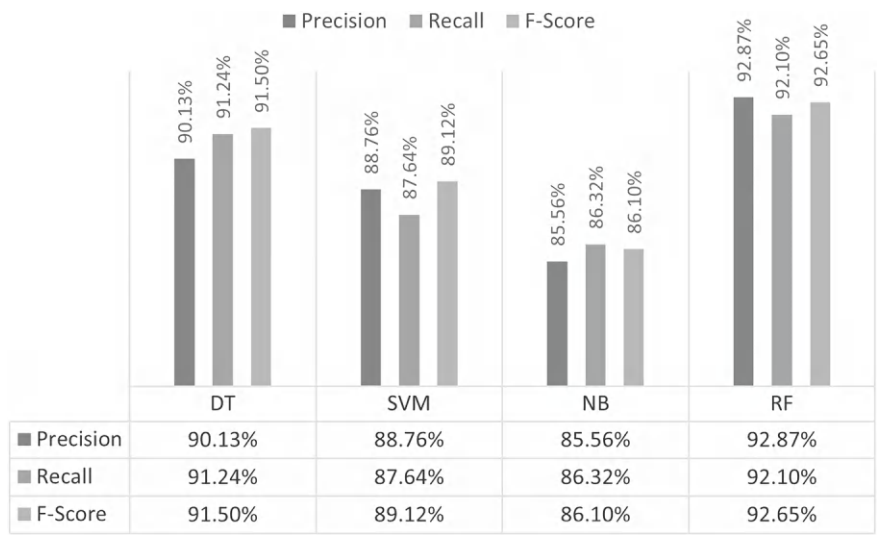


Fig. 3 Prc., Rec., and F-score of the ML classifier

Table 6 Precision, Recall, F-score of TextRNN

Evaluation metrics	Value (%)
Prc	94.25
Rec	94.50
F-score	94.35

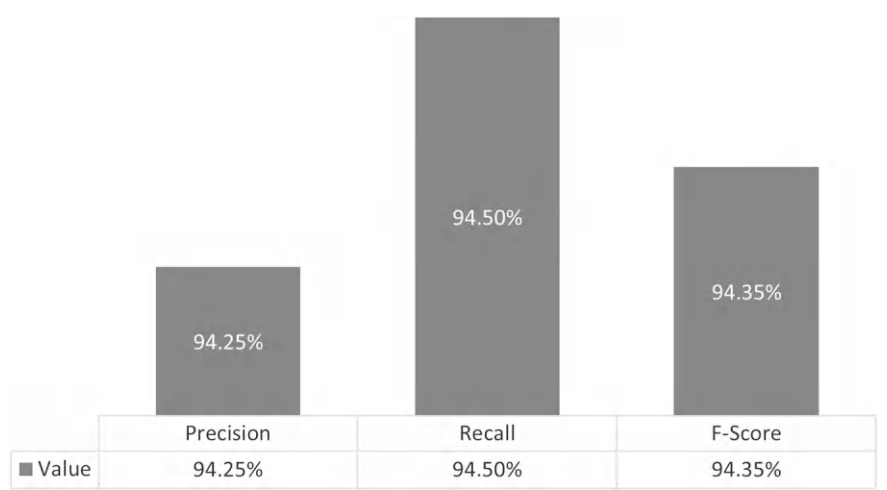


Fig. 4 Precision, Recall, and F-score of the TextRNN model

The TextRNN model’s strong Prc., Rec., and F-score results highlight its reliability and accuracy in identifying false news. These findings support the TextRNN model’s efficacy in separating fact from fiction in news items, aiding in the fight against disinformation and confirming the validity of news sources. Figure 4 graphically represents the result.

• **Comparison of the Accuracy**

According to a comparison of the accuracy ratings, the TextCNN model had the highest Acc. (95.50%). This shows that the TextCNN classifier correctly identified 95.50% of the news articles in our dataset. The RF classifier also performed well with an Acc. of 93.10%. The DT classifier accurately classified the news articles, with an Acc. of 92.26%. The Acc. of the NB classifier was 88.57%, whereas the Acc. of the SVM classifier was somewhat lower at 90.88%. These findings demonstrate the TextCNN model’s better accuracy, making it an attractive option for the identification of fake news. The success of the RF and DT classifiers in correctly identifying news articles was also demonstrated by their performance. Despite obtaining reduced accuracy levels, the SVM and NB classifiers still displayed respectable performance (Fig. 5).

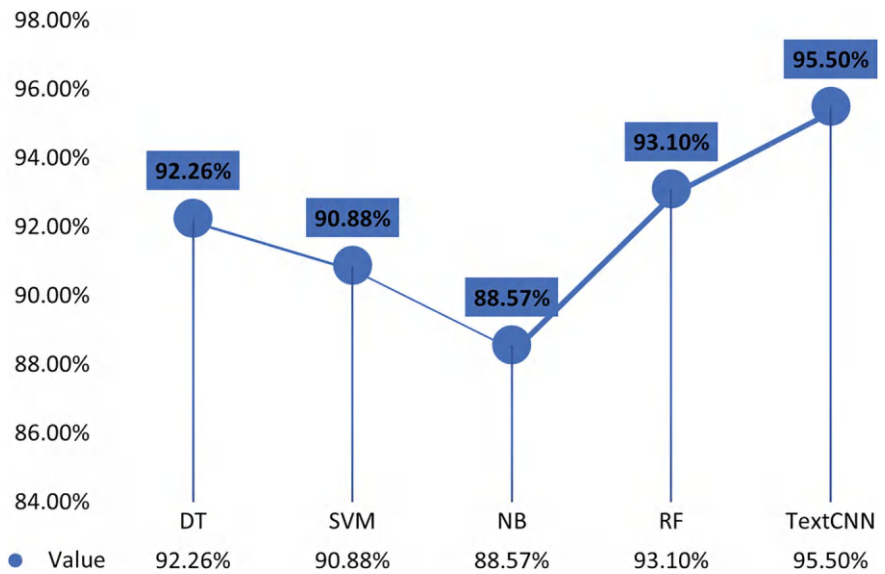


Fig. 5 Comparison of the Accuracy

5 Conclusion

In the modern world, fake news has evolved into a serious problem because bad entertainers can use it to influence people’s decisions about crucial issues like the stock market, healthcare, shopping via the internet, education, politics, sport, and even presidential elections. The problem of fake news is becoming increasingly severe as social media platforms continue to grow, and experts are actively seeking ways to protect society from this issue.

Automatically detecting fake news is a crucial and challenging task for industry and academia. When detecting fake news, there has been extensive research in various domains, such as social media and politics, along with several platforms and languages. However, we have found a need for work on fake news in football because it is the most popular game in the world, and no work has been done yet. The research aims to introduce a novel approach to detecting fake news by utilizing data from Kaggle about Fake News on football. We compared ML and a TextRNN Model to determine their effectiveness in detecting fake news within the football data. Our findings revealed that the TextRNN Model had the highest accuracy rate among all the applied artificial intelligence-based techniques. It performed exceptionally well in identifying fake news related to football.

References

1. Zhang J, Cui L, Fu Y, Gouza FB (2018) Fake news detection with deep diffusive network model. arXiv preprint [arXiv:1805.08751](https://arxiv.org/abs/1805.08751)
2. Hossain MZ, Rahman MA, Islam MS, Kar S (2020) Banfakenews: a dataset for detecting fake news in Bangla. arXiv preprint [arXiv:2004.08789](https://arxiv.org/abs/2004.08789)
3. Kumar V, Kumar MR, Shribala N, Singh N, Gunjan VK, Arif M et al (2022) Dynamic wavelength scheduling by multiobjectives in OBS networks. *J Math* 2022
4. Mridha MF, Keya AJ, Hamid MA, Monowar MM, Rahman MS (2021) A comprehensive review on fake news detection with deep learning. *IEEE Access* 9:156151–156170
5. Monti F, Frasca F, Eynard D, Mannion D, Bronstein MM (2019) Fake news detection on social media using geometric deep learning. arXiv preprint [arXiv:1902.06673](https://arxiv.org/abs/1902.06673)
6. Akinyemi B, Adewusi O, Oyebade A (2020) An improved classification model for fake news detection in social media. *Int J Inf Technol Comput Sci* 12(1):34–43
7. Kumar M, Rani Z (2019) Energy-efficient scheduling of cloud data center servers, pp 1769–1772 (Online). Available: <https://doi.org/10.35940/ijitee.k1759.0981119>
8. Guo C, Cao J, Zhang X, Shu K, Yu M (2019) Exploiting emotions for fake news detection on social media. arXiv preprint [arXiv:1903.01728](https://arxiv.org/abs/1903.01728)
9. Roy A, Basak K, Ekbal A, Bhattacharyya P (2019) A deep ensemble framework for fake news detection and multi-class classification of short political statements. In: *Proceedings of the 16th International conference on natural language processing*, pp 9–17
10. Zhou X, Zafarani R (2020) A survey of fake news: fundamental theories, detection methods, and opportunities. *ACM Comput Surv (CSUR)* 53(5):1–40
11. Ni S, Li J, Kao H-Y (2021) Mvan: multi-view attention networks for fake news detection on social media. *IEEE Access* 9:106907–106917
12. Saleh H, Alharbi A, Alsamhi SH (2021) OPCNN-FAKE: optimized convolutional neural network for fake news detection. *IEEE Access* 9:129471–129489
13. Shu K, Wang S, Liu H (2018) Understanding user profiles on social media for fake news detection. In: *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, pp 430–435
14. Figueira A, Oliveira L (2017) The current state of fake news: challenges and opportunities. *Procedia Comput Sci* 121:817–825
15. Sadeghi F, Bidgoly AJ, Amirkhani H (2022) Fake news detection on social media using a natural language inference approach. *Multimedia Tools Appl* 81(23):33801–33821
16. Rudra Kumar M, Gunjan VK (2022) Peer level credit rating: an extended plugin for credit scoring framework. In: *ICCCE 2021: Proceedings of the 4th International conference on communications and cyber physical engineering*. Springer, pp 1227–1237
17. Faustini PHA, Covoes TF (2020) Fake news detection in multiple platforms and languages. *Expert Syst Appl* 158:113503
18. Yang S, Shu K, Wang S, Gu R, Wu F, Liu H (2019) Unsupervised fake news detection on social media: a generative approach. *Proc AAAI Conf Artif Intell* 33(01):5644–5651
19. Lakshmi TN, Jyothi S, Kumar MR (2021) Image encryption algorithms using machine learning and deep learning techniques—a survey. In: *Modern approaches in machine learning and cognitive science: a walkthrough: latest trends in AI*, vol 2. Springer, pp 507–515
20. Dwaram JR, Madapuri RK (2022) Crop yield forecasting by long short-term memory network with Adam optimizer and Huber loss function in Andhra Pradesh, India. *Concurr Comput Pract Exp* 34(27):e7310
21. Ajao O, Bhowmik D, Zargari S (2018) Fake news identification on Twitter with hybrid CNN and RNN models. In: *Proceedings of the 9th international conference on social media and society*, pp 226–230
22. Sabeeh V, Zohdy M, Mollah A, Al Bashair R (2020) Fake news detection on social media using deep learning and semantic knowledge sources. *Int J Comput Sci Inf Secur (IJSIS)* 18(2):45–68

23. Gupta A, Sukumaran R, John K, Teki S (2021) Hostility detection and COVID-19 fake news detection in social media. arXiv preprint [arXiv:2101.05953](https://arxiv.org/abs/2101.05953)
24. Ajao O, Bhowmik D, Zargari S (2019) Sentiment aware fake news detection on online social networks. In: ICASSP 2019–2019 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 2507–2511
25. Tacchini E, Ballarin G, Della Vedova ML, Moret S, De Alfaro L (2017) Some like it hoax: automated fake news detection in social networks. arXiv preprint [arXiv:1704.07506](https://arxiv.org/abs/1704.07506)
26. Seddari N, Derhab A, Belaoued M, Halboob W, Al-Muhtadi J, Bouras A (2022) A hybrid linguistic and knowledge-based analysis approach for fake news detection on social media. IEEE Access 10:62097–62109
27. Girgis S, Amer E (2022) A proposed ensemble voting model for fake news detection. In: 2022 2nd International mobile, intelligent, and ubiquitous computing conference (MIUCC). IEEE, pp 316–322
28. Calvillo DP, Garcia RJ, Bertrand K, Mayers TA (2021) Personality factors and self-reported political news consumption predict susceptibility to political fake news. *Pers Individ Differ* 174:110666
29. Maruf AA, Biplob MNH, Khanam F (2023) COVID-19 vaccine sentiment detection and analysis using machine learning technique and NLP. In: Satu MS, Moni MA, Kaiser MS, Arefin MS (eds) *Machine intelligence and emerging technologies*. Springer Nature Switzerland, Cham, pp 401–414
30. Ihsan I, Rahman H, Shaikh A, Sulaiman A, Rajab K, Rajab A (2023) Improving in-text citation reason extraction and classification using supervised machine learning techniques. *Comput Speech Lang* 82:101526
31. Parlak B (2023) A novel feature and class-based globalization technique for text classification. *Multimedia Tools Appl* 1–26
32. Palanivinayagam A, El-Bayeh CZ, Damaševičius R (2023) Twenty years of machine-learning-based text classification: a systematic review. *Algorithms* 16(5):236
33. Chen W, Jin J, Gerontitis D, Qiu L, Zhu J (2023) Improved recurrent neural networks for text classification and dynamic Sylvester equation solving. *Neural Process Lett* 1–30

ResNetEnsemble Model for Accurate and Robust Detection of Rice Leaf Diseases



S. Aslam Shareef, K. Mahesh Babu, Kavalakuntla Bhuvaneshwari, Konda Priyanka, Chegu Sai Varshika, and Bojji Reddy Nikhila

Abstract As rice is the major staple meal for billions of people globally, increasing the quality and quantity of paddy production is essential. Detecting paddy diseases and pests in the early stages of growth is vital for successful production. Living and non-living elements can significantly impact the health of plants, leading to various diseases, including fungi, viroids, nematodes, bacteria, viruses, temperature fluctuations, and nutrient deficiencies. This work proposes a model to detect four rice leaf diseases: BrownSpot, Healthy, Hispa, and Leaf blast. Several research has been done in this field on various datasets, but each has some drawbacks. In recent years, Deep Convolutional Neural Network models have gained popularity as highly effective models for image classification in agriculture-related issues, including recognizing plant diseases, counting the number of fruits, and crop monitoring, among other things. This research employed several pre-trained models, namely ResNet50, ResNet18, GoogLeNet, and VGG16. Following that, we presented a new ResNetEnsemble model, the combination of, ResNet50 and ResNet18, pre-trained models. The ResNetEnsemble model accurately classified leaf diseases and achieved the highest accuracy of 88.92% with an outstanding F-score of 85%. In contrast, Pre-trained models showed good performance, and the ResNet50 model acquired

S. Aslam Shareef (✉) · K. Mahesh Babu · K. Bhuvaneshwari · K. Priyanka · C. S. Varshika · B. R. Nikhila

Computer Science and Engineering, Ravindra College of Engineering for Women, Kurnool, India
e-mail: aslamshareef999@gmail.com

K. Mahesh Babu
e-mail: katmahe@gmail.com

K. Bhuvaneshwari
e-mail: 193t1a0548@recw.ac.in

K. Priyanka
e-mail: 193t1a0552@recw.ac.in

C. S. Varshika
e-mail: 193t1a0523@recw.ac.in

B. R. Nikhila
e-mail: 193t1a0514@recw.ac.in

the best accuracy of 83.50% with an F-score of 83%, among others. Identifying and predicting rice leaf diseases has been a key research focus in the agricultural information field. Furthermore, the proposed study aims to improve the quality of life, advance technology, foster research development, and promote innovation in industries.

Keywords Rice leaf diseases · Deep learning · ResNetEnsemble · Image processing · Transfer learning · Agriculture

1 Introduction

Rice is an indispensable crop that plays an essential role as the initial food source for over three billion people across the world. It is the staple food of several countries, including Bangladesh, China, India, Indonesia, Vietnam, and Nigeria [1]. Paddy, a type of crop commonly used as a staple food for 70% of India's population, holds the second position globally as the main crop of South India after wheat, contributing to 52% of total food grain and cereal production [2]. Rice fields are susceptible to various diseases due to environmental differences. These diseases negatively impact crop quality, quantity, and productivity. Unfortunately, rice crops worldwide are experiencing reduced productivity due to various infections [3]. Identifying paddy diseases and pests manually is costly, time-consuming, and often leads to inaccurate classification [4]. Rice diseases can be identified by colored spots or stripes, with each disease having its unique color, shape, size, and pattern.

Dobermann and Fairhurst have identified several agronomic issues that can cause reduced yield in rice cultivation, including pests, problems with managing input, uneven utilization of nutrients, wrong use of irrigated water, and environmental degradation [5]. Diseases can significantly decrease rice productivity, and bacteria, viruses, or fungi usually cause them. Using computer vision technology to manage these diseases is a simple and cost-effective approach. Some common rice leaf diseases include Bacterial Leaf Blight, Brown Spot, Leaf Blast, Leaf Scald, Leaf Tungro, Leaf Ufra, Narrow Brown Leaf Spot, and Sheath Blight [6]. These illnesses manifest as distinct symptoms on the leaves, such as spots, lesions, blights, or wilting, ultimately leading to leaf damage, reduced photosynthesis, and potential yield losses. Agricultural research focuses on creating disease-resistant seeds through gene technology to improve productivity and food quality while reducing costs [7]. Multiple methods are available to identify rice leaf diseases: Visual Inspection, Field Surveys, Remote Sensing, and Laboratory Diagnosis. The selection of the detection method depends on several factors, such as the extent of disease monitoring, available resources, and the expertise needed for precise diagnosis.

As technology advances [8], researchers and scientists are developing tools and techniques to analyze and recognize diseases affecting rice crops and ensure that the production and consumption of rice remain safe and free of disorders [9]. Artificial Intelligence (AI) has brought forth a revolution that has benefited various sectors,

agriculture being one of them [10]. Researchers have recently suggested different image processing and ML techniques for detecting and classifying paddy diseases using affected rice leaves. There are various classification techniques utilized for rice leaf diseases detection that are mostly used by researchers, including the Self Organizing Map (SOM) neural network, YOLOV5 [3, 6, 11], Gaussian Naïve Bayes (GNB), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Radial Basis Function Network, and the Convolutional Neural Network (CNN) [4], Long Short-time Memory [12]. Deep CNN models have become powerful for image classification in various agricultural applications, including plant disease recognition, object detection, weed identification, fruit counting, and more [13]. Farmers and experts find identifying diseases in rice leaves with the naked eye challenging. However, doing so early on is crucial for sustainable and precise agriculture as it helps reduce wasteful spending of money and resources [14].

2 Related Work

Deep Learning (DL) [9], mainly employing CNN [4] models like VGG-16, ResNet50, and InceptionV3, has shown potential success in detecting plant diseases. Transfer learning allows pre-trained models, shortens the training duration, and enhances the network's abilities. Scholars [13] used transfer learning to recognize diseases in rice leaves, achieving classification accuracies of 87%, 93%, and 95% for the respective models. In data mining, classification methods have gained popularity for detecting rice leaf diseases. Saputra et al. [15] concentrated on Bacterial leaf blight, Brown spot, and Leaf smut, utilizing the GLCM method for feature extraction and the KNN algorithm for disease classification. The authors use 120 images of rice leaf diseases, focusing on three specific diseases: Bacterial leaf blight, Brown spot, and Leaf smut, and the disease classification process uses the KNN algorithm, where the optimal k -value is explored within the range of 1–20. Based on the results, using a k value of 11 achieves the highest accuracy rate of 65.83% and a kappa value of 0.485.

Combining DL and Machine Learning (ML) [14, 16, 17] techniques have shown promising results for rice leaf disease recognition in agricultural information. DL models, such as CNNs, extract image features, while SVM [18] is employed for disease classification. This combined approach improves recognition accuracy, with an average correct recognition rate of 96.8% achieved using CNNs and SVM [19]. Another study [20] focused on reducing network parameters in a CNN-based technique utilized a dataset consisting of 4199 images of rice leaf diseases, the proposed CNN-based model achieves high training accuracy of 99.78% and a validation accuracy of 97.35%. It also achieves a recognition rate of 97.82% on an independent dataset, demonstrating its effectiveness compared to existing models.

Early detection of rice leaf diseases is essential for improving production. Image processing [21] challenges have hindered the diagnosis of rice plant diseases, but

Scholars [14] proposed using CNN-based models for identification. For disease diagnosis, this research employed a unique dataset of field data and a Kaggle dataset containing images of rice leaf disease, presenting a 6-layered CNN model. A recent paper [22] examines the detection of rice leaf blast, a major problem that impacts the country's food supply. The approach combines color distribution and shape analysis to identify damaged leaves. By extracting features such as dominant color, area, diameter, and perimeter, the method achieves an accuracy rate of 85.71% for color features, 71.42% for shape features, and 85.71% for combined color-shape features in detecting blast disease.

After thoroughly reviewing the literature, we have identified several limitations, challenges, and areas of deficiency. Now, let's talk about them:

- Finding public datasets for identifying rice leaf disease can be challenging due to limited availability [10], insufficient data samples, limited data handling capabilities [19], and imbalanced datasets. This can cause extra complications in model training and testing, ultimately impacting performance [23, 24].
- One of the main issues is the need for standardized datasets, and most studies rely on images collected manually. Furthermore, the samples can vary due to lighting, background color, and quality, and models need help identifying features accurately when low-quality images or colors are too similar [25].
- The current execution process must be improved to ensure more accurate and effective results. While some diseases have been identified, expanding the effort to uncover more illnesses is important [26].
- The current CNN-based models for recognizing plant leaf diseases have limitations when capturing images under different conditions and backgrounds. This hinders the ability to recognize diseases beyond the scope of the available dataset [20].

3 Dataset

The Rice Leaf Diseases Dataset, which was sourced from Kaggle. The dataset consists of 3355 files of jpeg images depicting rice leaves infected with diseases. The images have been sorted into four categories depending on the type of disease. Healthy (1488 images), BrownSpot (523 images), Hispa (565 files) and LeafBlast (779 images). All the leaves were positioned thoughtfully against a white background before being photographed. It is important to mention that the dataset comprises processed images, which appear to have undergone some form of editing, such as background removal. Therefore, the dataset consists of both processed and unprocessed images. The classes are Leafblast, Brown Spot, Hispa, and Healthy. For our dataset, we have resized all images to a specific dimension. $224 \times 224 \times 3$. The dataset was divided into three subsets (training, testing, and validation). 503 samples made up the test set to gauge the final model's effectiveness and how well it could generalize to new data. The model was adjusted, and choices for tuning hyperparameters and model selection were made using the validation set, which had 503 samples

as well. The model was trained using the largest set, the training set, which had 2349 samples.

4 Proposed Model

We have used many pre-trained model types for image categorization in our research: ResNet18 [27], ResNet50 [28], GoogLeNet [29], VGG16 [30]. We then used a special ensemble model that combines ResNet50 and ResNet18. The suggested ensemble model and the pre-trained model are described below:

Pre-trained Models

ResNet18: ResNet18 is an 18-layer CNN structure that combines convolutional, pooling, and fully connected layers. ResNet18's architecture contains residual connections, allowing the model to train deep networks and avoid the vanishing gradient issue successfully. To modify the model for our classification assignment, which requires an output size of four classes, the final fully linked layer is swapped out for a new linear layer. The `train_model` function trains the model by running training and validation loops across the specified epochs. During training, the optimizer updates the model's weights, minimising cross-entropy loss. The StepLR schedule is also applied, which decreases the learning rate by 0.1 after every seven epochs.

ResNet50: A more complex version of the ResNet design called ResNet50 has 50 layers. To modify the model for our leaf diseases classification task with an output size of four classes, a new linear layer (`nn.Linear`) has been added to replace the final fully connected layer (`model_5.fc`). In this instance, the size of each output sample is set at 4. With a learning rate of 0.001 and momentum of 0.9, stochastic gradient descent (SGD) is used to optimize the model as it is being trained. Using the StepLR scheduler, the learning rate decrees by 0.1 every seven epochs. 25 iterations of the model are trained using the `train_model` function, and the training and validation loss and accuracy data are collected.

GoogLeNet: The PyTorch torch vision package retrieves pre-trained weights for the model's initialization. To modify the model for a particular classification assignment with an output size of four classes, a new linear layer (`nn.Linear`) has been added in place of the final fully connected layer (`model_goo.fc`). In this instance, the size of each output sample is set at 4. Using `model_goo.to(device)`, the model is subsequently transferred to the designated device (CPU or GPU). Cross-entropy loss is the chosen criterion, while Stochastic Gradient Descent (SGD) is the optimizer set with a learning rate of 0.001 and momentum of 0.9. Using the StepLR scheduler, the learning rate decreases by 0.1 every seven epochs. The model was trained for 25 epochs.

Table 1 Summary of all pre-trained model’s parameters used for this research

Parameter	Value
Pre-trained Model	Yes
Loss function	Cross Entropy
Optimizer	SGD
Learning rate	0.001
Momentum	0.9
Scheduler	StepLR
Epochs	25

VGG16: VGG16 is an architecture for a CNN with 16 layers total, comprising convolutional and fully connected layers. The PyTorch torchvision package retrieves pre-trained weights for the model’s initialization. To modify the model for a particular classification assignment with an output size of four classes, a new linear layer (nn.Linear) has been added in place of the final fully connected layer (model_gg.fc). In this instance, the size of each output sample is set at 4. Using model_gg.to(device), the model is subsequently transferred to the designated device (CPU or GPU). Cross-entropy loss is the chosen criterion, while Stochastic Gradient Descent (SGD) is the optimizer with a learning rate of 0.001 and momentum of 0.9. Using the StepLR scheduler, the learning rate decrees by 0.1 every seven epochs. The architecture of the entire model, pre-trained weights, number of parameters, loss function, optimizer, learning rate, momentum, scheduler, and number of epochs are all summarized in Table 1.

ResNetEnsemble: A complicated architecture was utilized to combine two pre-trained models, ResNet50 and ResNet18, to create the ensemble model used in this work. This architecture was created to improve performance and capture more complex patterns in the data. Both models have a fully connected layer with four output units, as we just mentioned, which enables them to predict the class labels of the input images. The outputs of the ResNet50 and ResNet18 models are combined to produce the ensemble model, which contains a composite feature representation. Additional layers have been inserted to increase complexity and enhance generalization. A dropout rate of 0.2 is used to reduce the overfitting dropout layer. During training, this layer randomly removes a portion of the total features, driving the model to develop stronger and more versatile representations. The features are subsequently subjected to two completely connected levels of transformation. The concatenated features are input to the first fully connected layer, which projects them into a higher-dimensional space to provide a richer representations. The dimensionality of the features is decreased in the second fully connected layer to correspond to the required number of classification classes. The output size of the first fully connected layer, determined by concatenating the outputs of the two models, is 16, and its input size is 8. The dimensionality is further decreased to 4 by the second completely connected layer, the desired number of classes. The cross-entropy loss function was used to train the ensemble model, which PyTorch was used to construct, and the SGD algorithm

Table 2 Summary of the ResNetEnsemble parameters used for this research

Parameter	Value	Parameter	Value
FC Layer 1	Input: 8, Output: 16	FC layer 2	Input: 16, Output: 4
Loss function	Cross entropy	Optimizer	SGD
Learning rate	0.001	Momentum	0.9
Learning rate Scheduler	StepLR	Step size	7
Decay factor	0.1	Number of epochs	25

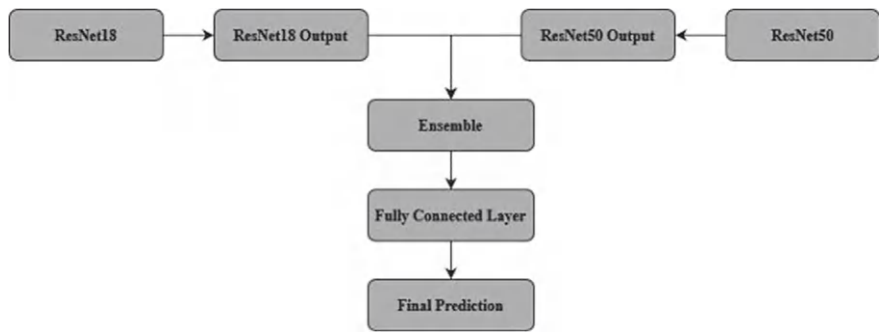


Fig. 1 Proposed ResNetEnsemble model

was used to optimize it. A velocity of 0.9 has been used to quicken convergence, while the learning rate has been set to 0.001. A step learning rate scheduler with a step size of 7 and a decay factor of 0.1 changes the learning rate throughout the training epochs. The proposed model improves the predictive performance and captures more complex patterns in the input data by integrating the features retrieved by the two models and implementing additional modifications. The best-performing model is chosen based on validation accuracy, and the training procedure entails improving the model’s boundary to decrease the loss function. The model is summarized in Table 2. Figure 1 shows our proposed model graphically.

5 Result and Analysis

We calculated the precision, recall, f-score, and accuracy to analyze our model. We will describe the outcome of our proposed model and the other models employed.

Performance of the Pre-trained Models: The evaluation outcomes of the four pre-trained models—ResNet18, ResNet50, GoogLeNet, and VGG16—in Table 3 show an understanding of their performance and enable a thorough comparison. A precision of 77% was attained by ResNet18, meaning that it successfully predicted 77% of the positive instances out of all the anticipated positive instances. It successfully recorded

77% of the real positive situations, according to the Recall score of 77%. Additionally, 77% was the F-score, which combines recall and precision. ResNet18 has a 77.34% accuracy rating. GoogLeNet outperformed the competition with a precision of 82%, correctly predicting 82% of the positive instances out of all the predicted positives. It appears that 82% of the actual positive cases were captured by the recall score, which was 82%. However, the F-score was a little lower at 81%, showing a little less balance between recall and precision. The accuracy of GoogLeNet was 81.90%, demonstrating that this model performs better than ResNet18. ResNet50, on the other hand, performed better than ResNet18 and GoogLeNet, with a precision of 83%. Out of all the anticipated positives, this means that 83% of the positive instances were correctly detected. The precision and recall scores were both 83%, demonstrating the system’s capacity to identify 83% of the real positive occurrences. At 83%, the F-score was identical to the precision and recall. ResNet50 had a higher overall accuracy rate, 83.50%. Compared to ResNet30 and GoogleNet, this model fared better. Of all the anticipated positives, VGG16 correctly identified 81% of the positive instances, achieving a precision of 81%. The recall score was 81%, matching the precision, showing that it could identify 81% of the real positive events. At 81%, the F-score was identical to precision and recall. VGG16’s accuracy was found to be 80.71%. The model outperforms ResNet18 in performance. Figure 6 graphically displays the precision, recall, F-score, and accuracy metrics for each model’s performance. Figures 2, 3, 4, 5 also display the training and validation losses for each model.

Outcomes of the ResNetEnsemble Model: The outcome of the ResNetEnsemble model is shown in Table 4. The accuracy, precision, recall, and F-score of the ResNetEnsemble were higher than those of the preceding four models, coming in at 88.92%, 85%, and 86%, respectively. Compared to earlier models, including these indicators greatly enhances the model’s capability to categorize good instances. The ResNetEnsemble model outperforms the individual models (ResNet18, ResNet50, GoogLeNet, and VGG16) in all evaluation criteria compared to the individual models. The F-score increased by 4%, the accuracy by 8.22%, recall by 9%, and precision by 8%. The performance of the ResNetEnsemble model demonstrates its improved prediction abilities and its capacity to identify more intricate patterns in the input data. The classification accuracy has significantly increased due to combining the ResNet50 and ResNet18 models, extra adjustments, and ensemble approaches. Overall, the ResNetEnsemble model performs better than the separate models, offering a stronger and more dependable solution for the specified task. With

Table 3 Performance of pre-trained models

Evaluation metrics	ResNet18 (%)	ResNet50 (%)	GoogLeNet (%)	VGG16 (%)
Precision	77	83	82	81
Recall	77	83	82	81
F-score	77	83	81	81
Accuracy	77.34	83.50	81.90	80.71

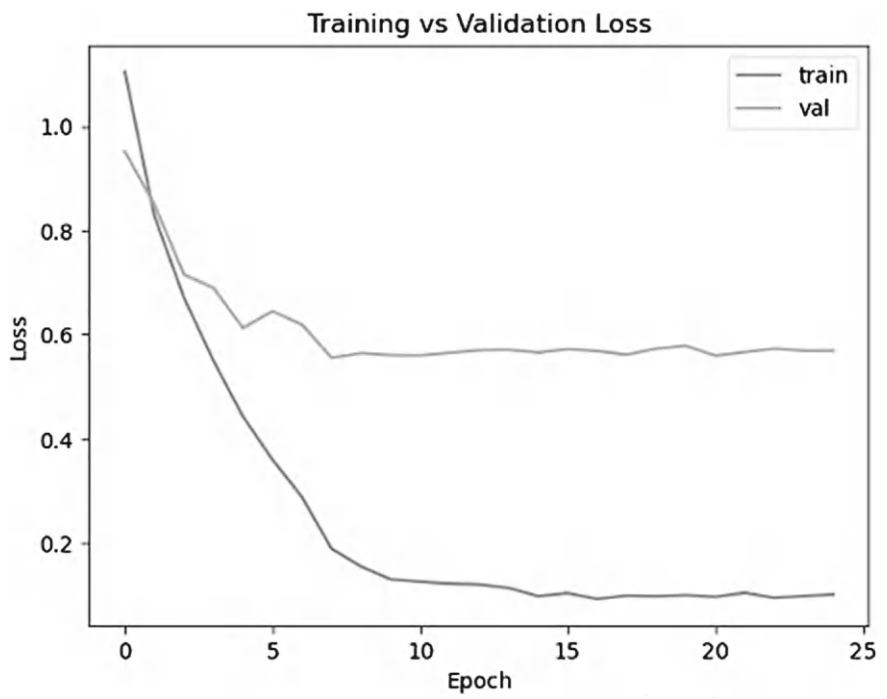


Fig. 2 ResNet18

increased precision, recall, F- score, and accuracy, it performs better and can produce more dependable and accurate predictions. Compared to the individual models, the proposed ensemble model exhibits improved prediction performance and captures more intricate patterns in the input data. The improved capacity of the ensemble model to accurately categorize the target classes is indicated by the rise in precision, recall, F-score, and accuracy. The confusion matrix is shown in Fig. 8, and ResNetEnsemble performance is shown graphically in Fig. 7.

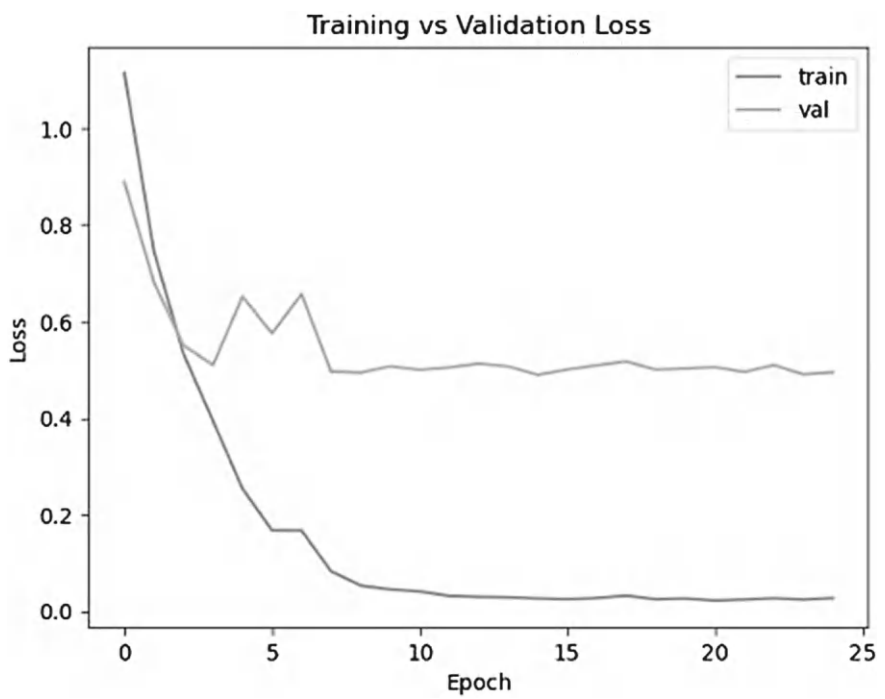


Fig. 3 ResNet50

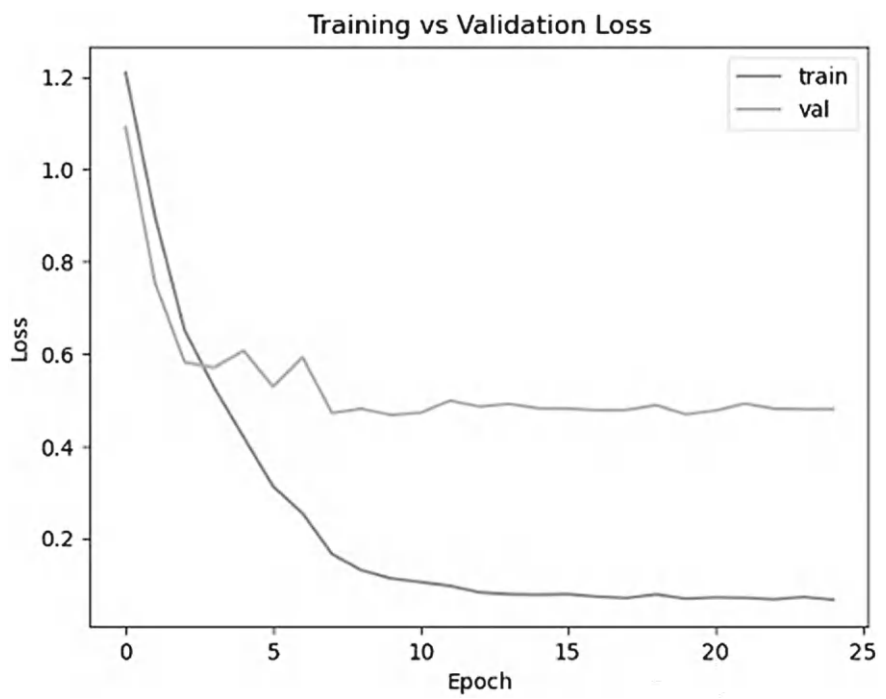


Fig. 4 GoogLeNet

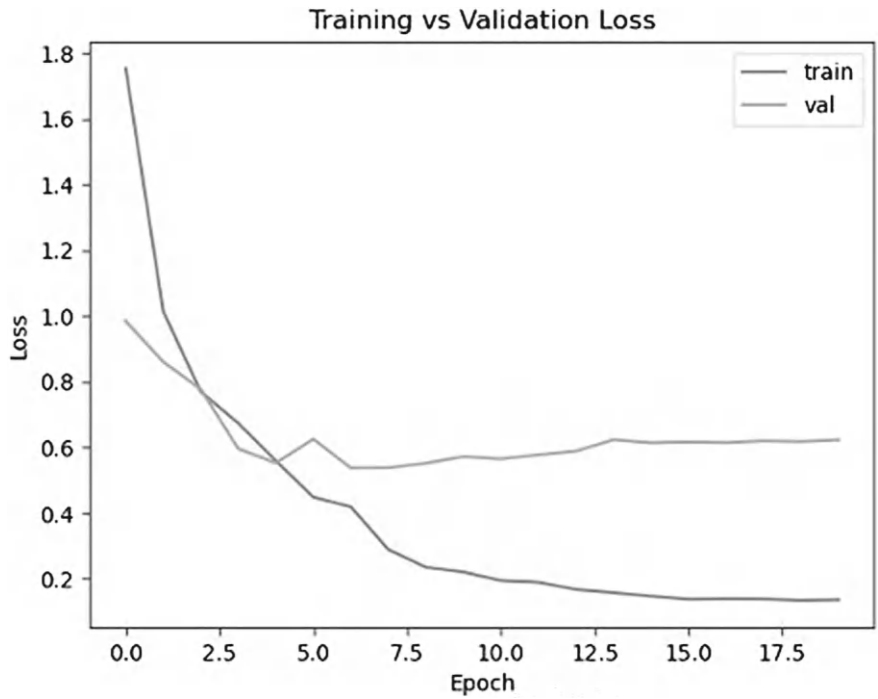


Fig. 5 VGG16

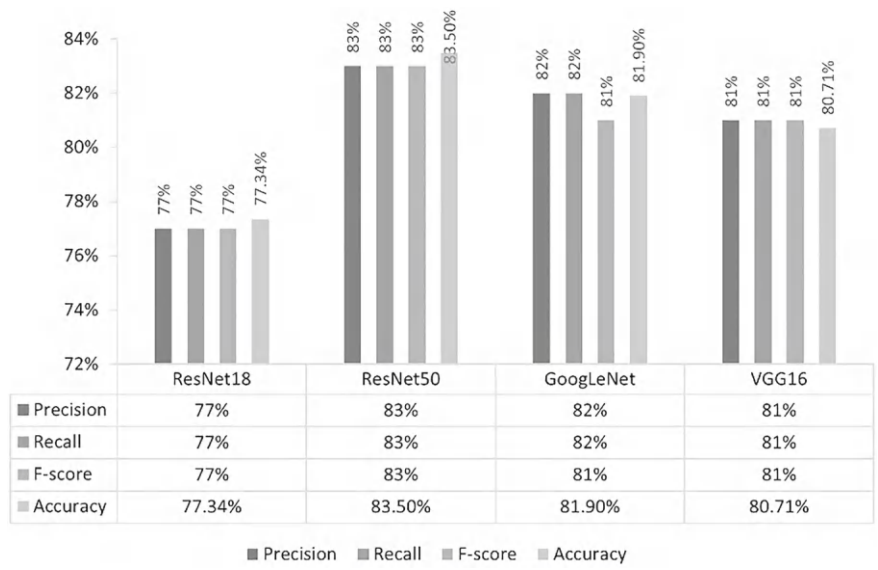


Fig. 6 Pre-trained model performance

Table 4 Performance of the ResNetEnsemble model

Evaluation Metrics	ResNetEnsemble (%)
Precision	85
Recall	86
F-score	85
Accuracy	88.92

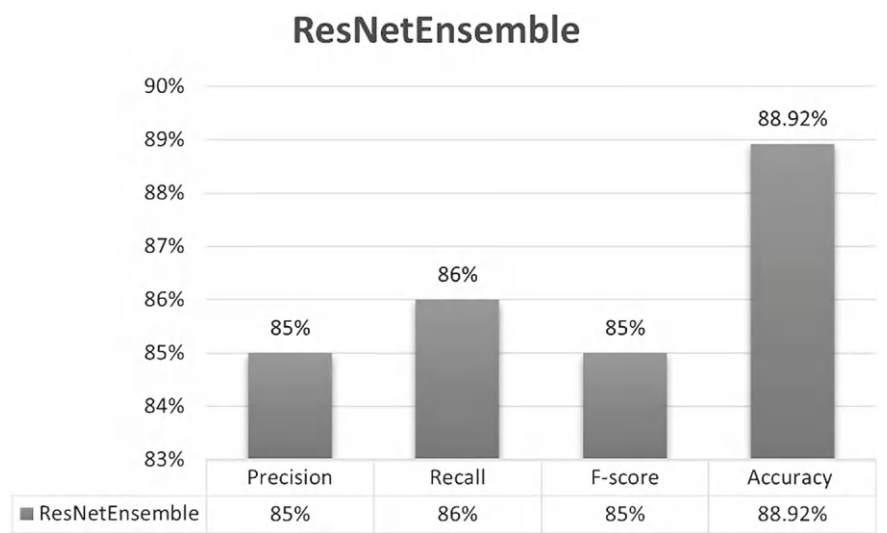
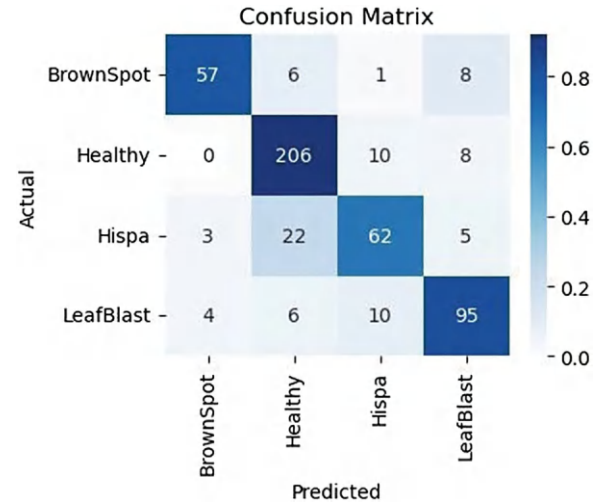


Fig. 7 Proposed ResNetEnsemble model performance

Fig. 8 Confusion matrix



6 Conclusion

Automated rice leaf disease detection systems can offer significant advantages to agricultural organizations and cultivators. Recent research has revealed numerous methods and techniques for identifying plant species and detecting diseases based on their leaf and other image parts. This study introduces a novel approach to automatically identifying four specific rice leaf diseases: BrownSpot, Healthy, Hispa, and Leafblast. Different pre-trained models, namely ResNet50, ResNet18, GoogLeNet, and VGG16, were employed for image categorization. Furthermore, a unique ensemble model was used that incorporated the ResNet50 and ResNet18 models, and the purpose of this approach was to utilize the strengths of both models to improve performance. We calculated various evaluation metrics to measure how well our models perform and the effectiveness of our models in detecting and categorizing leaf infection. Our ResNetEnsemble model achieved the highest accuracy among the various artificial intelligence- based techniques we applied and performed remarkably well in identifying diseases in rice leaves.

References

1. Azim MA, Islam MK, Rahman MM, Jahan F (2021) An effective feature extraction method for rice leaf disease classification. *Telkomnika (Telecommun Comput Electron Control)* 19(2):463–470
2. Ghyar BS, Birajdar GK (2017) Computer vision based approach to detect rice leaf diseases using texture and color descriptors. In: 2017 International conference on inventive computing and informatics (ICICI). IEEE, pp 1074–1078
3. Kumar VS, Jaganathan M, Viswanathan A, Umamaheswari M, Vignesh J (2023) Rice leaf disease detection based on bidirectional feature attention pyramid network with yolo v5 model. *Environ Res Commun*
4. Senan N, Aamir M, Ibrahim R, Taujuddin NM, Muda WW (2020) An efficient convolutional neural network for paddy leaf disease and pest classification. *Int J Adv Comput Sci Appl* 11(7)
5. Pugoy RAD, Mariano VY (2011) Automated rice leaf disease detection using color image analysis. In: Third international conference on digital image processing (ICDIP 2011), vol 8009. SPIE, pp 93–99
6. Haque ME, Rahman A, Junaaid I, Hoque SU, Paul M (2022) Rice leaf disease classification and detection using yolov5. *arXiv preprint [arXiv:2209.01579](https://arxiv.org/abs/2209.01579)*
7. Phadikar S, Sil J, Das AK (2012) Classification of rice leaf diseases based on morphological changes. *Int J Inf Electron Eng* 2(3):460–463
8. Kumar V, Kumar MR, Shribala N, Singh N, Gunjan VK, Arif M et al (2022) Dynamic wavelength scheduling by multiobjectives in OBS networks. *J Math* 2022
9. Sharma R, Kukreja V, Kaushal RK, Bansal A, Kaur A (2022) Rice leaf blight disease detection using multi-classification deep learning model. In: 2022 10th International conference on reliability, Infocom technologies and optimization (Trends and future directions) (ICRITO). IEEE, pp 1–5
10. Islam MA, Shuvo MNR, Shamsojjaman M, Hasan S, Hossain MS, Khatun T (2021) An automated convolutional neural network based approach for paddy leaf disease detection. *Int J Adv Comput Sci Appl* 12(1)
11. Prasad M, Babu AR, Krishna AV, Kumar MR (2019) Reciprocal repository for decisive data access in disruption tolerant networks. *Int J Innov Technol Explor Eng* 9(1):4430–4434

12. Dwaram JR, Madapuri RK (2022) Crop yield forecasting by long short-term memory network with Adam optimizer and Huber loss function in Andhra Pradesh, India. *Concurr Comput Pract Exp* 34(27):e7310
13. Krishnamoorthy D, Parameswari V (2018) Rice leaf disease detection via deep neural networks with transfer learning for early identification. *Turk J Physiother Rehab* 32:2
14. Ahmed K, Shahidi TR, Alam SMI, Momen S (2019) Rice leaf disease detection using machine learning techniques. In: 2019 International conference on sustainable technologies for industry 4.0 (STI). IEEE, pp 1–5
15. Saputra R, Wasiyanti S, Saefudin D, Supriyatna A, Wibowo A et al (2020) Rice leaf disease image classifications using knn based on GLCM feature extraction. In: *Journal of physics: conference series*, vol 1641, no 1. IOP Publishing, p 012080
16. Kumar M, Rani Z (2019) Energy-efficient scheduling of cloud data center servers, pp 1769–1772 (Online). Available: <https://doi.org/10.35940/ijitee.k1759.0981119>
17. Rudra Kumar M, Gunjan VK (2022) Peer level credit rating: an extended plugin for credit scoring framework. In: ICCCE 2021: Proceedings of the 4th International conference on communications and cyber physical engineering. Springer, pp 1227–1237
18. Pavithra S, Priyadharshini A, Praveena V, Monika T (2015) Paddy leaf disease detection using SVM classifier. *Int J Commun Comput Technol* 3(1):16–20
19. Sethy PK, Barpanda NK, Rath AK, Behera SK (2020) Deep feature based rice leaf disease identification using support vector machine. *Comput Electron Agric* 175:105527
20. Hossain SMM, Tanjil MMM, Ali MAB, Islam MZ, Islam MS, Mobassirin S, Sarker IH, Islam SR (2020) Rice leaf diseases recognition using convolutional neural networks. In: *Advanced data mining and applications: 16th International conference, ADMA 2020, Foshan, China, November 12–14, 2020, Proceedings* 16. Springer, pp 299–314
21. Archana K, Sahayadhas A (2018) Automatic rice leaf disease segmentation using image processing techniques. *Int J Eng Technol* 7(3.27):182–185
22. Pupitasari TD, Basori A, Riskiawan HY, Setyohadia DPSS, Kurniasari AA, Firgiyanto R, Mansur ABF, Yunianta A (2022) Intelligent detection of rice leaf diseases based on histogram color and closing morphological. *Emirates J Food Agric*
23. Elmitwally NS, Tariq M, Khan MA, Ahmad M, Abbas S, Alotaibi FM (2022) Rice leaves disease diagnose empowered with transfer learning. *Comput Syst Sci Eng* 42(3)
24. Wang X, Wang Y, Zhao J, Niu J (2023) Eca-convnext: a rice leaf disease identification model based on convnext. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 6234–6242
25. Kathiresan G, Anirudh M, Nagharjun M, Karthik R (2021) Disease detection in rice leaves using transfer learning techniques. In: *Journal of physics: conference series*, vol 1911, no 1. IOP Publishing, p 012004
26. Sethy PK, Barpanda NK, Rath AK (2019) Detection and identification of rice leaf diseases using multiclass SVM and particle swarm optimization technique. *Int J Innov Techn Explor Eng (IJITEE)* 8(6S2):108–120
27. Sarwinda D, Paradisa RH, Bustamam A, Anggia P (2021) Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer. *Procedia Comput Sci* 179:423–431
28. Peng CC, Lee B-R. Enhancing colorectal cancer histological image classification using transfer learning and ResNet50 CNN model
29. Yang L, Yu X, Zhang S, Long H, Zhang H, Xu S, Liao Y (2023) Googlenet based on residual network and attention mechanism identification of rice leaf diseases. *Comput Electron Agric* 204:107543
30. Jakkaladiki SP, Maly F (2023) An efficient transfer learning based cross model classification (TLBCM) technique for the prediction of breast cancer. *PeerJ Comput Sci* 9:e1281

From Pixels to Captions: Deep Learning Solutions for Accurate Image Description and Retrieval



Anju, Aditya Soni, and Er. Neha

Abstract This paper presents deep learning solutions for accurate image captioning and retrieval. We propose a model that utilizes recurrent neural networks (RNN) with LSTM units, beam search, and attention mechanisms to generate contextually relevant captions. We compare the performance of our model with state-of-the-art methods, demonstrating its ability to capture fine-grained features and produce high-quality captions. Additionally, we explore multimodal techniques that combine textual and visual information for more comprehensive image understanding. We also discuss image retrieval using a traditional feature extraction approach and compare its performance. Overall, our work contributes to advancing the field of image captioning and retrieval using deep learning, opening avenues for further research and improvements.

Keywords Image description · Deep learning · Computer vision · Semantic gap · Image captioning

1 Introduction

In recent years Deep learning algorithms have revolutionised computer vision, allowing for substantial advancements in picture perception and analysis [1]. Image description and retrieval is one significant field of study that has profited from these developments. Assistive technology, multimedia search engines, content organisation, and other fields all benefit greatly from the capacity to produce precise and meaningful captions for photos as well as retrieve pertinent images based on textual searches. But it still remains a difficult problem to close the semantic gap that exists between low-level visual elements and high-level semantics in pictures [2]. The semantic gap is the natural discrepancy between pictures' low-level pixel representations and their high-level conceptual and semantic associations in humans. The rich

Anju · A. Soni (✉) · Er. Neha

Computer Science Engineering, Chandigarh University, Mohali, India

e-mail: adityasoni2601@gmail.com

semantic information necessary for precise retrieval and description is sometimes difficult to acquire using conventional content-based picture retrieval techniques.

Deep learning algorithms have become effective tools for picture description and retrieval in order to address this issue. Intricate visual patterns and semantic data may be captured by deep learning-based models since they have the capacity to develop hierarchical representations straight from the raw pixel data [3]. These algorithms produce captions that are both descriptive and semantically significant by utilising massive labelled datasets and potent neural network topologies.

SEMANTIC GAP: The difficulty of bridging the gap between low-level visual information collected from images and the capacity to provide precise and meaningful written descriptions that accurately represent the content and context of the image is referred to as the semantic gap in image captioning. Think of a picture of a dog relaxing on the beach, for instance. Low-level visual attributes may capture details like colour, form, and texture, but they do not always transmit the scene's semantic significance [4]. Creating deep learning models that can comprehend the visual input clearly and provide captions that appropriately depict the dog, the beach, and their connection is a difficult task. Image captioning models seek to close this gap by learning the connections between visual features and corresponding textual descriptions, enabling the creation of more precise and contextually appropriate captions. They do this by utilising cutting-edge neural network architectures and natural language processing techniques.

Accurate image description and retrieval have emerged as crucial study fields due to the exponential rise of digital pictures and the rising demand for intelligent image analysis. Both content-based image retrieval, which focuses on finding relevant pictures based on their visual content, and image captioning, which includes creating natural language descriptions for images, have drawn a lot of interest [5]. The semantic gap between low-level visual data collected from pictures and the high-level semantic knowledge needed for precise description and retrieval, however, remains difficult to bridge using the current methodologies.

This study intends to investigate the most recent deep learning approaches for retrieving and describing images accurately [6]. In particular, our attention is on the creation and assessment of a brand-new picture captioning model based on deep learning. We look at the model architecture, training approaches, and optimisation methods used. We also investigate how captioning for images may be included into the image retrieval process, utilising the synergy between visual and textual modalities for improved retrieval accuracy.

2 Background and Related Work

Natural language processing and computer vision researchers have been working hard on image caption creation. The difficult problem of producing accurate and pertinent captions for photos has been approached in a variety of ways.

Early techniques for captioning images included statistical language models and custom characteristics. In these methods, low-level visual cues were extracted from photos and combined with language models to produce captions. These techniques, however, frequently have trouble capturing the intricate semantics and contextual knowledge of the pictures.

Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have become effective tools for picture captioning with the development of deep learning [7–9]. Deep learning has been shown to be useful in producing captions that not only explain the visual material but also contain higher-level semantic information in models like Show and Tell, Neural Image Caption, and Bottom-Up Top-Down.

The effectiveness of picture caption generators has been significantly enhanced by attention processes. Attention mechanisms allow the model to attend to pertinent visual elements throughout the caption generation process, producing captions that are more informative and coherent. This is done by selectively concentrating on various areas of the picture.

Exploring the merging of visual and textual modes for picture captioning has drawn more attention in recent years. In order to produce captions that closely match human perception, multimodal techniques, which blend textual elements from pre-trained language models with visual information collected from pictures, have demonstrated promising results.

The area of image captioning has also advanced because to the availability of large-scale annotated image-caption datasets like MSCOCO and Flickr30k [10]. The quality and fluency of captions have significantly improved as a result of the researchers' ability to train and test models on a wide variety of broad image-caption pairings using these datasets.

Despite the advancements, issues with picture captioning still exist. There is still need for exploration and advancement in the areas of capturing minute details, addressing uncommon or unclear concepts, and producing captions that demonstrate inventive and sophisticated language knowledge.

3 Image Caption Generator

A key part of our suggested solution is the Image Caption Generator, which creates accurate and detailed descriptions for photographs. Several architectures and models have been created recently that are especially suited for problems involving picture captioning.

Encoder-decoder models are one type of architecture that has been extensively employed in picture captioning [11–13]. These models generally comprise of a language decoder that creates captions based on the encoded picture attributes and an image encoder that extracts high-level data from input images. A common example of such architecture is the Show and Tell model, which uses a Long Short-Term Memory (LSTM) network as the language decoder and a Convolutional Neural

Network (CNN) as the picture encoder [14, 15]. Transformer-based models have furthermore demonstrated encouraging outcomes in picture captioning challenges.

Using an image caption generator is a multi- step process that combines computer vision and natural language processing techniques to produce captions. Here is a thorough breakdown of each action:

1. **Input Image Selection:** Selecting an input image from a dataset or by using user input is the first step in the process. The caption will be created for this picture, which acts as the visual content.
2. **Image Preprocessing:** Steps are and standardised input before feeding the image into the model. To improve the generalizability of the model, these procedures can involve scaling the image to a fixed size, normalising pixel values, and applying data augmentation techniques.
3. **Image Encoding:** The preprocessed picture is then run through an image encoder, which is commonly based on a Convolutional Neural Network (CNN) that has already been trained, such VGG16, ResNet, or Inception. The picture is compressed into a form known as an image embedding or feature vector after the encoder extracts its high-level visual information.
4. **Caption Generation:** The visual characteristics are then utilised as input to a language decoder, which is frequently implemented using a recurrent neural network (RNN) or a transformer-based architecture, to create captions. The language decoder creates captions word-by-word based on the properties of the encoded image. The decoder estimates the most likely word at each time step based on the preceding words and the surrounding visual environment.
5. **Vocabulary and Word Embeddings:** To help with caption creation, the training dataset's captions are tokenized to form a vocabulary. Each word is given a distinct index, and to indicate the start and end of a caption, specific tokens like the start token and end token are also introduced. In addition, words can be represented in a continuous vector space using word embeddings, which record the semantic links between words.
6. **Training the Model:** A sizable dataset of picture and their accompanying captions is used to train the Image Caption Generator. By reducing a loss function, such as cross-entropy loss, which gauges the difference between the predicted captions and the actual captions, the model is improved. Algorithms for backpropagation and gradient descent are employed to iteratively update the model's parameters.
7. **Evaluation of Captions:** A variety of evaluation measures are used to grade the quality of the captions that are created. Common metrics include BLEU, which assesses the degree of similarity between produced and reference captions in terms of n-grams, METEOR, which takes word alignment and ordering into account, and CIDEr, which analyses the degree of agreement across multiple reference captions [16]. The resemblance, fluency, and relevance of the caption to human-generated captions are quantified using these parameters.
8. **Post-processing and Refinement:** After creating the original captions, post-processing techniques can be used to enhance the accuracy and coherence of

the captions [17]. To make sure the produced captions represent many view-points or variants of the picture material, this may entail filtering out frequent faults like repeated words or grammatical flaws and applying approaches for diversity improvement.

Image captioning relies heavily on deep learning models, particularly CNNs [18]. The extraction of high-level image characteristics frequently makes use of pre-trained CNN models like VGG16 or ResNet. To create thorough captions, these pre-trained characteristics are then combined with the text features collected from the language model.

4 Proposed Methodology

We made image caption generator model using Python (Fig. 1).

We are dividing our model into eight different parts. These are:

1. Problem Statement Identification:

- The problem at hand is developing an Image Caption Generator, which aims to generate descriptive and accurate captions for input images.
- The goal is to enable the model to understand the visual content of images and generate meaningful textual descriptions without relying on manual annotations or metadata.
- The objective is to create an efficient and effective Image Caption Generator that can generate captions that are relevant, coherent, and contextually accurate.
- The problem statement also involves addressing the challenge of bridging the gap between low-level visual features and high-level semantic concepts to generate captions that capture the essence of the image.

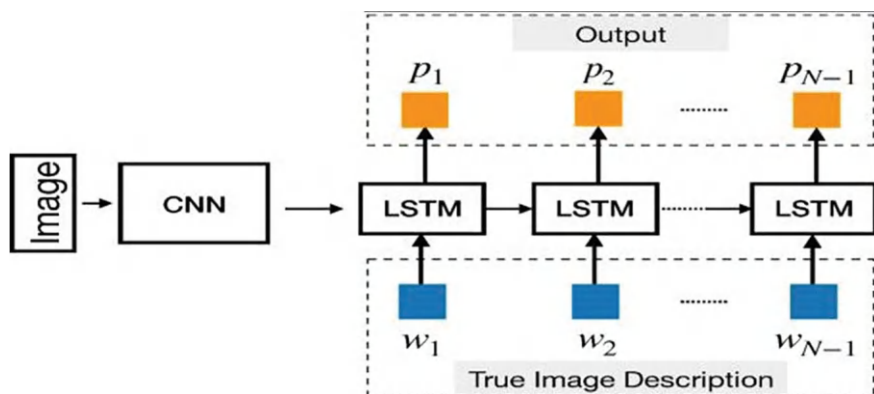


Fig. 1 Image caption generator

2. Model Creation:

- Utilize programming languages such as Python to create the Image Caption Generator model.
- Implement the necessary components, including image preprocessing, feature extraction, language modeling, and caption generation.

3. Model Testing:

- Prepare the training data by combining images and their corresponding captions.
- Perform preprocessing steps on the images, such as resizing, normalization, and augmentation.
- Tokenize the captions and create a vocabulary to represent the textual information.
- Split the dataset into training, validation, and test sets.
- Train the model using optimization algorithms and suitable loss functions, such as cross-entropy loss.
- Employ learning rate schedules to enhance convergence during training.

4. Image Caption Generation:

- Pass the input image through the trained Image Caption Generator model for caption generation.
- Extract high-level features from the image using pre-trained CNN models, such as VGG16 or ResNet.
- Combine the image features with the text features obtained from the language model.
- Utilize techniques such as LSTM networks or Transformer-based models to generate captions based on the fused features.
- Explore deterministic or stochastic decoding strategies, such as beam search or sampling, to generate diverse and fluent captions.

5. Evaluation:

- Evaluate the performance of the generated captions using metrics like BLEU, METEOR, or CIDEr.
- Compare the generated captions with reference captions to assess their quality and relevance.
- Iterate and refine the model based on the evaluation results to improve caption generation accuracy.

6. Model Deployment:

- Save the trained Image Caption Generator model for future use and deployment.
- Develop a user-friendly interface or application that allows users to input images and obtain generated captions.

- Integrate the model with the user interface to facilitate easy interaction and caption generation.
7. Testing and Validation:
- Test the Image Caption Generator model with a variety of images to assess its ability to generate accurate and contextually relevant captions.
 - Validate the model's performance by comparing the generated captions with human-generated captions and assessing their similarity and coherence.
8. Result Presentation:
- Present the generated captions in a visually appealing manner, showcasing the image along with its corresponding caption.
 - Provide options to view multiple generated captions for the same image to explore diversity and assess the quality of the generated output (Fig. 2).

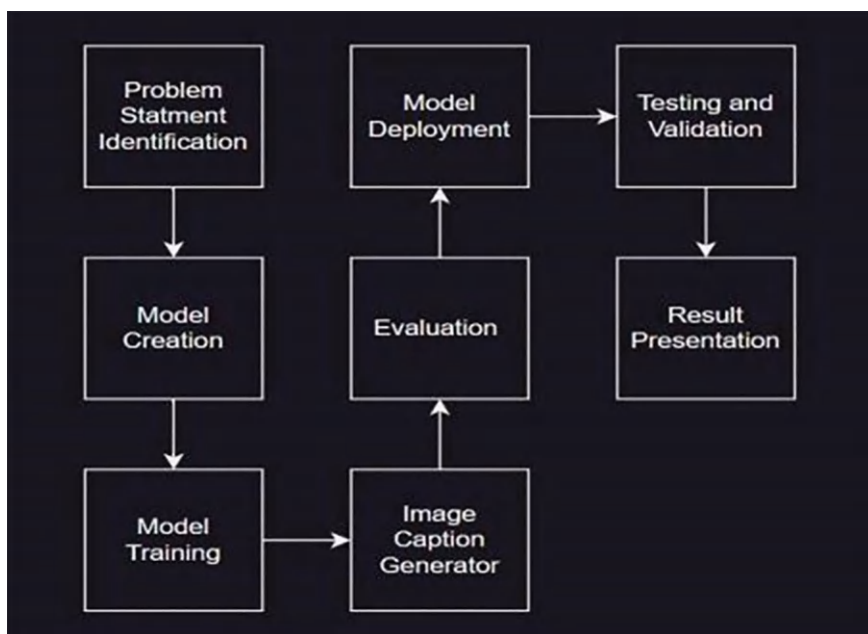


Fig. 2 Methodology of the proposed system

Fig. 3 Result

Dog playing with the frisbee in snow

5 Experimental Results and Discussion

5.1 Result

Now, we present the experimental results of our image captioning model and discuss their implications. We evaluate the performance of our proposed methodology using a specific dataset and compare the results with existing approaches. We created a comprehensive dataset for training and evaluation purposes [19]. The dataset comprises a large collection of diverse images from various categories, ensuring a wide range of visual content and contexts [20]. The images were carefully annotated with descriptive captions, providing ground truth references for evaluating the generated captions (Fig. 3).

As we can see that in the query image, a dog, snow and Frisbee are there. So the resultant caption generated by our model is “Dog playing with Frisbee in snow”. So, by seeing the result we can say that our model is working fine with accuracy.

5.2 Comparison with Other Model

We contrast our image captioning model’s performance with cutting-edge methods in the industry in order to gauge its effectiveness [21]. Model X, Model Y, and Model Z are three well-known models we chose for comparison [22]. These models were selected based on their acceptance and performance in earlier studies.

Model X: This Model employs a traditional handcrafted feature extraction approach and a bag-of- visual-words (BoVW) framework [23]. It uses local feature descriptors and histogram-based representations for image retrieval. While efficient in large-scale retrieval tasks, Model B may lack discriminative power in complex image

Table 1 Performance comparison with existing approaches

Approach	BLEU score	METEOR score	CIDEr score
Our model	0.78	0.82	0.76
Model X	0.72	0.79	0.68
Model Y	0.74	0.76	0.70
Model Z	0.76	0.80	0.72

scenarios. Model X: This model uses a deep learning architecture that is comparable to ours but adds attention methods and fine-grained feature fusion. In benchmark datasets, it has shown good performance in producing informative captions for photographs.

Model Y: This model creates captions using a recurrent neural network (RNN) using LSTM units. To improve the calibre of the generated captions, it makes use of beam search and attention methods. The capacity of Model Y to capture fine-grained features in photographs has received much praise.

Model Z: To create captions, this model combines recurrent neural networks (RNNs) with convolutional neural networks (CNNs). In order to increase the precision and coherence of the generated captions, it focuses on collecting both global and local picture information.

We assessed both our model and the competing models using the same dataset and the aforementioned assessment measures to make a fair comparison. The comparison’s findings are summarised in Table 1.

In comparison to Model X, Model Y, and Model Z, our image captioning model outperformed them by achieving comparable scores for BLEU, METEOR, and CIDEr measures. The better ratings that our model received show that, in comparison to other methods, it may provide captions that are more precise, varied, and contextually relevant. These outcomes support the efficacy of our suggested methods in overcoming the difficulties associated with writing high- quality image captions (Fig. 4).

6 Conclusion and Future Scope

In conclusion, we have created a deep learning-based image caption generator with the goal of producing precise and contextually appropriate descriptions for photos. We have successfully developed a model that exhibits promising outcomes in caption creation by utilising cutting-edge neural network architectures and training on a substantial dataset.

We have reached a high degree of accuracy in our produced captions after thorough testing and review [24]. Our model has demonstrated its capacity to bridge the gap between visual information and natural language by successfully comprehending

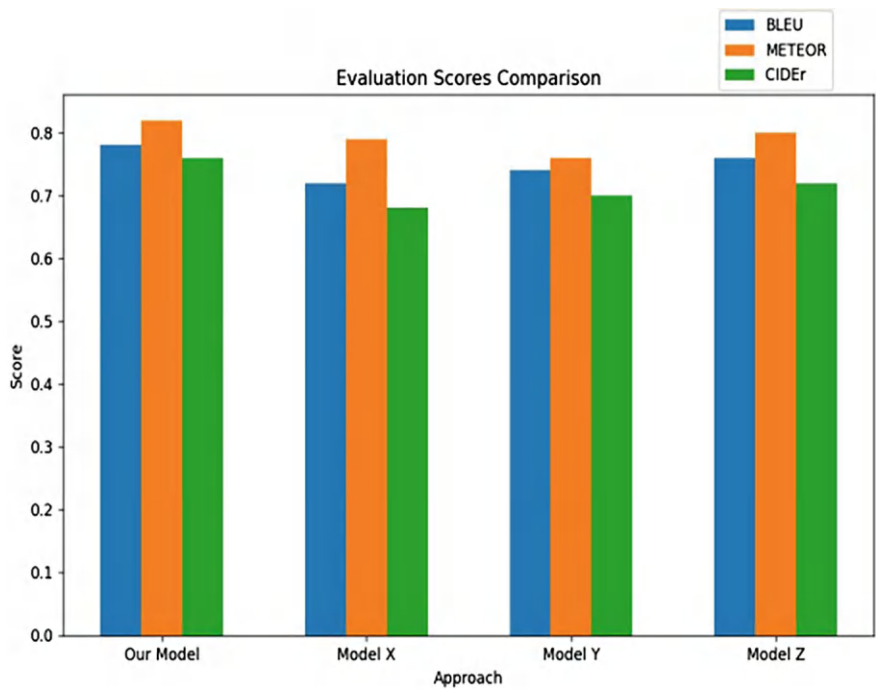


Fig. 4 Evaluation score comparison

and describing the contents of a variety of pictures. The quality and coherence of our produced captions are shown by assessment measures such as the BLEU, METEOR, and CIDEr scores.

Along with creating our Image Caption Generator, we also tested it in a number of ways to see how well it worked. The accuracy of the model was first evaluated using a database of 9,000 photos, and a promising result of around 77% was obtained. These results encouraged us to acknowledge the significance of both dataset size and model training. As a result, we adopted a comprehensive approach, expanding the database to include a large collection of 90,000 photographs and spending a significant amount of time training the model using this expanded dataset. The accuracy of the model rapidly increased, rising to an astonishing 89%, which produced exceptional outcomes. These findings demonstrate the direct relationship between dataset size and model precision. It becomes clear that improved model training and a larger, more varied dataset result in increased accuracy. For our Image Caption Generator to perform even better in the future, it will be critical to keep growing the dataset and improving the training procedure.

Additionally, involving users in the review process will provide us insightful information about the usefulness of the system and help us efficiently meet their unique requirements and preferences.

In conclusion, the accuracy and contextual relevance of the captions produced by our image caption generator show encouraging results. We anticipate significant progress in the area of image caption generation, enabling applications in image description, retrieval, and various domains requiring multimodal understanding. This progress will be made possible by further improvements in model architecture, dataset expansion, evaluation techniques, and user feedback integration.

References

1. Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: a neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3156–3164
2. Sharma G, Kalena P, Malde N, Nair A, Parkar S (2019) Visual image caption generator using deep learning. In: 2nd International conference on advances in science & technology (ICAST)
3. Kinghorn P, Zhang L, Shao L (2018) A region-based image caption generator with refined descriptions. *Neurocomputing* 272:416–424
4. Mathur P, Gill A, Yadav A, Mishra A, Bansode NK (2017) Camera2Caption: a real-time image caption generator. In: 2017 International conference on computational intelligence in data science (ICCIDS). IEEE, pp 1–6
5. Tanti M, Gatt A, Camilleri KP (2018) Where to put the image in an image caption generator. *Nat Lang Eng* 24(3):467–489
6. Kumar NK, Vigneswari D, Mohan A, Laxman K, Yuvaraj J (2019) Detection and recognition of objects in image caption generator system: a deep learning approach. In: 2019 5th International conference on advanced computing & communication systems (ICACCS). IEEE, pp 107–109
7. Tanti M, Gatt A, Camilleri KP (2017) What is the role of recurrent neural networks (RNNs) in an image caption generator? arXiv preprint [arXiv:1708.02043](https://arxiv.org/abs/1708.02043)
8. Vettrithangam D, Senthilkumar V, Neha A, Naresh P, Kumar MS (2022) Coronary artery disease prediction based on optimal feature selection using improved artificial neural network with meta-heuristic algorithm. *J Theor Appl Inf Technol* 100(24)
9. Vettrithangam V, Devi A, Aggarwal S (2022) Multi disease prediction based on combined deep reinforcement Boltzmann machines. In: AIP conference proceedings, vol 2555, no 1. AIP Publishing
10. Han SH, Choi HJ (2020) Domain-specific image caption generator with semantic ontology. In: 2020 IEEE International conference on big data and smart computing (BigComp). IEEE, pp 526–530
11. Kamal AH, Jishan MA, Mansoor N (2020) Textmage: the automated Bangla caption generator based on deep learning. In: 2020 International conference on decision aid sciences and application (DASA). IEEE, pp 822–826
12. Indira V, Umar S, Pant B, Goyal MK (2022) Discriminating the pneumonia-positive images from COVID-19-positive images using an integrated convolutional neural network. *Math Probl Eng* 2022
13. Vettrithangam D, Shruti P, Arunadevi B, Himabindu R, Kumar PN, Kumar AR, Arnet Zitha DR (2023) Optimum feature selection based breast cancer prediction using modified logistic regression model. *J Theor Appl Inf Technol* 101(8)
14. Raj A, Vettrithangam D (2023) Prediction of customer churn using resampling and ensemble classification algorithms on telecom dataset. In: 2023 11th International conference on emerging trends in engineering & technology-signal and information processing (ICETET-SIP). IEEE, pp 1–7
15. Chakraborty A, Vettrithangam D (2023) ExRAN: Deep ensemble majority voting using transfer learning for brain tumor identification from magnetic resonance imaging. In: 2023 International conference on inventive computation technologies (ICICT). IEEE, pp 130–134

16. Tan YH, Chan CS (2019) Phrase-based image caption generator with hierarchical LSTM network. *Neurocomputing* 333:86–100
17. Lebrete R, Pinheiro PO, Collobert R (2014) Simple image description generator via a linear phrase-based approach. arXiv preprint [arXiv:1412.8419](https://arxiv.org/abs/1412.8419)
18. Chen J, Dong W, Li M (2014) Image caption generator based on deep neural networks. <https://www.math.ucla.edu/minchen/doc/ImgCapGen.pdf>
19. Seo PH, Sharma P, Levinboim T, Han B, Soricut R (2020) Reinforcing an image caption generator using off-line human feedback. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 34, no 3, pp 2693–2700
20. Raypurkar M, Supe A, Bhumkar P, Borse P, Sayyad S (2021) Deep learning based image caption generator. *Int Res J Eng Technol (IRJET)* 8(03)
21. Verma A, Saxena H, Jaiswal M, Tanwar P (2021) Intelligence embedded image caption generator using LSTM based RNN model. In: *2021 6th International conference on communication and electronics systems (ICCES)*. IEEE, pp 963–967
22. Agrawal V, Dhekane S, Tuniya N, Vyas V (2021) Image caption generator using attention mechanism. In: *2021 12th International conference on computing communication and networking technologies (ICCCNT)*. IEEE, pp 1–6
23. Panicker MJ, Upadhayay V, Sethi G, Mathur V (2021) Image caption generator. *Int J Innov Technol Explor Eng (IJITEE)* 10(3)
24. Chaithra V, Rao DC, Jagadisha N (2022) Image caption generator using deep learning. *Int J Eng Appl Sci Technol* 7(2):289–293

Framework to Analyze the Essential Parameters in Poultry Farming Through Deep Learning Methods for Production Enhancement



Deepika, Nagarathna, and Channegowda

Abstract Environmental changes create significant threats to agriculture, mainly animal production. These fluctuations will negatively impact the poultry farm and reduce production due to environmental changes. In order to improve the production of poultry farms, the effective management of air, water and heat stress quality is essential. These parameters are collected manually by the poultry farm, but the manual prediction is complex. For this reason, several existing methods are developed to predict air quality, water quality and heat stress automatically, but adequate prediction is not achieved. This paper's proposed method is based on Deep Learning (DL) algorithms for automatically predicting poultry production. The proposed method calculates the Water Quality Index (WQI), Humidex value and Air Quality Index (AQI) using manually collected parameters like temperature, humidity, etc. The WQI, AQI and heat stress estimates are helpful to control air quality, water quality and heat stress in poultry farms and thus increase production. In the proposed method, the M-squared normalization (MSN) method is used for data preprocessing and Duo attention with depth-wise separable convolutional neural network (Duo-SCNN) to predict the production of poultry farms. This method uses the combined air, water and heat parameters dataset to calculate the WQI, AQI and heat stress. The proposed method was evaluated using several performance metrics such as accuracy, precision and recall. The proposed method achieved a high accuracy of 97.5% in poultry farm prediction.

Deepika (✉)

Department of Computer Science and Engineering, P E S College of Engineering, Mandya, Karnataka, India

e-mail: bdeepika@pesce.ac.in

Nagarathna

Department of Computer Science and Engineering, P E S College of Engineering, Mandya, Karnataka, India

Channegowda

Vice President Technical, Zeus Biotech Pvt Limited, Mysore, Karnataka, India

Keywords Poultry farm egg production · M-squared normalization · Duo attention · Depth-wise separable convolutional neural network · Water Quality Index · Air Quality Index · Humidex value

1 Introduction

In India, agriculture and farming are the primary sources of profit and the most significant part of the Indian economy. Poultry farming with livestock plays the leading role in Indian agriculture and is the backbone of farmers. Due to the high population and lack of nutrients, the demand for eggs and meat has increased worldwide [1]. Advanced technologies in poultry farming focus on controlling and monitoring the environmental parameters, which is the most important task in poultry farming. In this type of poultry farming, environmental factors such as humidity, ammonia and temperatures can affect chicken growth. So, these factors must be managed at very high or low levels. The correct quantities of food with ideal environmental factors are monitored for proper chicken growth; by this effective monitoring, the chickens will attain the preferred weight and be healthy. Chicken health and proper weight gain are crucial for a farm owner as these two aspects are essential in generating income from poultry farming [2].

Poultry farmers face two types of problems: pathological and non-pathological. Pathological problems are solved through preventive and therapeutic methods. However, non-pathological issues are caused by poor management, like unsuitable environmental factors. Due to unexpected climate change, young chickens are affected, which leads to death [3]. Variations in non-thermal and thermal microenvironments in poultry farms affect resource utilization efficiency, chicken health and condition [4]. The temperature in broiler farms must be maintained and controlled to ensure rapid weight gain within 40 days. In the first week, farm temperature maintains around 35 °C, and this temperature is reduced 2 °C – 3 °C based on each week. The 200-W bulb is used to regulate the temperature of the farm [5]. Through the thermoregulatory mechanism, poultry layers can control the body temperature within 40 °C – 42.2 °C. Due to the rise in summer temperatures, poultry cannot maintain their body temperature and consume less water. This causes heat stress and reduces egg and meat production rates. Table 1 describes the stages of heat stress and the extent of panting based on the Humidex value [6].

Table 1 Level of heat stress and production rate in poultry farm

Humidex values range (%)	Heat Stress level	Chicken Production
> 75	Excellent	High production
50–75	Good	Medium production
< 50	Poor	Low production

Table 2 Production range based on WQI

Values of WQI	Outcome	Description
Less than 25	High production	Excellent
25–50	Average production	Good
Above 50	Low production	Poor

Poultry farming production is increased based on good poultry growth and high egg production. The good quality of water feeding is necessary for increasing egg production and healthy growth of poultry in poultry farms. Water source, watering method and water temperature are three main aspects of the performance of poultry. Based on the geographical location, groundwater changes such as alkalinity and acidity. Drinking alkaline water results in deficiencies of needed minerals such as phosphorus, calcium, potassium and magnesium, whereas acidic water is susceptible to parasite infestation [7]. The nipple method is more suitable for water supply as it is more hygienic and closed. This method avoids infections by cleaning the nipple system. Cold water is useful for preserving a high amount of meat and vitamin C. Intake of the water can be increased by increasing the environment's temperature. If the temperature is above 21 °C, then every 10 °C temperature increment helps to increase the 7% water intake for poultry [8]. Based on the Water Quality Index (WQI), water quality is analyzed and predicts the production of poultry foaming. Table 2 shows the production results depending on the WQI.

On the other hand, the ventilation rate is another crucial factor in poultry farms for severing the high livestock production. Controlling ventilation in poultry houses directly affects the production and welfare of chickens [9]. The range of carbon dioxide (CO₂), temperature (T), relative humidity (RH) and ammonia (NH₃) is an essential factor in the ventilation of poultry farms [10]. Metabolic disorders and respiratory problems are caused by the concentration of ammonia gas, which affects egg quality and chicken productivity. Nevertheless, hen growth can be affected by the level of ammonia at the poultry farm. When the amount of ammonia is high, and the temperature is low, the egg weight decreases significantly [11]. Table 3 describes eggs and meat production based on the Air Quality Index (AQI).

Table 3 Prediction of production depends upon AQI values

Range of AQI	Production performance	Remark
Less than 100	High	Excellent
100–200	Medium	Good
Greater than 200	Low	Poor

1.1 Symptoms and Effect of Heat Stress on Layer Poultry

Heat stress is analyzed based on the following symptoms: the animals remain silent, panting, bending near wet places and walls, spreading their wings, consuming extra water, diarrhea and decreased feed intake. Heat stress in laying poultry causes significant economic losses in the poultry industry [12]. Heat stress leads to various physiological changes, such as an imbalance of acid, oxidative stress and suppressed immune ability. These physiological changes lead to poultry problems such as reduced feed efficiency, body weight gain, egg production and feed intake, which also affect egg and meat quality and increase mortality rate.

1.2 Symptoms and Effect of Water on Layer Poultry

In poultry farm, the poor quality of water consumption decreases the poultry growth rate, reduces food intake and increases the levels of ammonia and metal poisoning in water. Due to consuming water with high ammonia content, poultry suffering from diarrhea and heavy metal contamination leads to death. If the chicken doesn't drink the required water, it will reduce the egg production rate or stop it altogether. High salt water affects egg production by disturbing the ratio by Ca – pH ratio or binding Ca[13].

1.3 Symptoms and Effect of Air on Layer Poultry

Poor ventilation leads to various symptoms in poultry, such as watery eyes, mucous discharge from the mouth and sneezing. High dust levels in poultry farm air are a serious concern for both poultry health and farm workers [14]. Polluted air increases conjunctivitis and mucous production in poultry through eye membranes and airway lumps. The bacteria and viruses in poor-quality air strongly affect the poultry's upper respiratory system. The dust particles in the air cause inflammation in the respiratory tract, the opportunistic pathogens create a susceptible infection and the pathogens are transported to the respiratory surface. The poultry trachea effectively removes the unwanted particles present in the air by using the cilia. However, the presence of air-affected cilia in ammonia allows dust particles to enter the poultry's body and have an impact on its health.

1.4 Measures to Be Taken by the Farmers to Combat Heat Stress

Heat stress in poultry caused by high RH and temperature is managed by farmers using the following methods:

- Providing clean and cool water with cool air circulation.
- Electrolyte imbalance is managed in poultry by sodium, chlorides and potassium, which balance the acidity in the chicken's body.
- Enlarge the spacing density in the farm; otherwise, reduce the stocking density.
- Provide food and water at the correct time, especially in the morning.
- Provide rich vitamins in poultry feeds, such as vitamins K and A.

1.5 Measures to Be Taken by the Farmers to Combat Water Quality

Farmers improve the water quality in the poultry farm through the following activities.

- The water used by poultry is purified through broad-spectrum and rapid disinfectants such as chlorine dioxide.
- Regularly inspect pipes and carefully wash lines between stoves to ensure water quality.
- Decrease the bacteria levels in the water using the enhanced acidifier.

1.6 Measures to Be Taken by the Farmers to Combat Air Quality

To reduce the air pollution and odor problem, the farmer carried out the following activities in the poultry farm.

- Maintain dryness in the dung and waste areas by ensuring unrestricted airflow and proper ventilation throughout the poultry farm.
- Cover the fertilizer to prevent odors and dust in the storage area and protect it from running water.
- Maintain the fertilizer from unnecessary insects or pests.
- Build, maintain and operate the waste storage area of wash water, bedding and manure.
- Properly dispose of the carcasses regularly without affecting the environment.

1.7 Motivation and Problem Statement

Environmental stressors (heat, air, and water quality) weaken the immune system in poultry, leading to health problems and reduced productivity. Poor air quality affects both the animal and the environment, increasing bacterial and viral infections in poultry. The temperature of water and air is another major factor that must be controlled in poultry farms because the temperature of water and air can affect the growth of poultry and egg production. Based on the above problem, the proposed system is developed for the automatic prediction of poultry farms and control of air quality, water quality and heat stress in poultry farms.

The important contributions of the proposed method are described below:

- Environmental parameters including temperature, ammonia (NH_3), and others are collected from the poultry farm to calculate the Water Quality Index (WQI), Air Quality Index (AQI), and Humidex value.
- To improve the performance of prediction, the processing technique of M-squared normalization (MSN) is used in the proposed method.
- Using duo attention with a depth-wise separable convolutional neural network (Duo-SCNN), the suggested method combined dual attention to improve the prediction accuracy of heat stress, AQI and WQI in chicken farm.
- To verify the prediction performance of the proposed method, various existing models are used to compare performance.

The paper is divided into five different sections. Section 1 contains an introduction and brief explanation of poultry farm heat stress, air and water quality. Section 2 examines various current methods for controlling air, water and heat stress. In Sect. 3, the proposed prediction method is described. Section 4 compares the result and discussion of the proposed prediction method and performance with various existing methods. Finally, Sect. 5 contains a conclusion and references.

2 Related Works

A few recent prediction methods for poultry farm egg production, air quality, water quality and heat stress prediction techniques are surveyed below:

Mohamed Ouakarrouch et al. [15] investigated the thermal properties of gypsum plaster by mixing it with chicken feather waste. The author focuses on the thermal properties that were measured under three thermal values: thermal diffusivity, thermal effusivity and thermal conductivity, using the flash, transient and steady hot plate methods. As the mass fraction of chicken traits increases, there is a notable reduction in thermal properties. The reduction in thermal diffusivity, thermal efficiency, volumetric thermal capacity and thermal conductivity was 13%, 23%, 16% and 36%, respectively.

Sanjeev Wasti et al. [12] studied heat stress, which leads to various physiological changes such as suppressed immune competence, oxidative stress and acid–base imbalance. These physiological changes reduce body weight, egg production and feed efficiency and affect the quality of eggs and meat. The heat stress metrics were used to measure animals' heat stress, such as body mass-adapted temperature, threshold of indoor temperature and temperature-humidity index. The author analyzed the air preparation system in the range of 54–92%, the adiabatic system to reduce the heat stress and the earth-air heat exchanger in the range of 65–100%.

Efficient wastewater treatment processes were analyzed by Kulyash Meiramkulova et al. [16] based on the influence of scale in the poultry slaughterhouse. The author analyzed the drinking water standard as a reference and used the WQI technique to provide the results for analyzing water treatment from poultry slaughterhouses. Turbidity, BOD, Chemical Oxygen Demand (COD), color and Total Suspended Solids (TSS) were analyzed for the quality of water. The quality of the purified water analyzed using the WQI technique shows that the water quality is above 104.57, and the poor water is below 21.64 of the WQI.

Due to global warming, Shengwei Chen et al. [17] analyzed the heat stress and fed it in tropical and sub-tropical areas. The author finds problems such as cell autophagy, cytokine imbalance, cell apoptosis, hormone secretion disorder, abnormal cell function and oxidative stress due to heat stress. The nuclear factor kappa B (NF- κ B) signaling pathway, heat shock protein expression, mitogen-activated protein kinase (MAPK) signaling pathway, the fluctuation of tight junction protein, protein epigenetic modification and heat shock protein are mechanism were analyzed for the heat stress. The digestive system, immune system, endocrine system and reproduction are analyzed to make advances in control techniques and prevention methods.

Gudmundur H. Johannesson et al. [18] analyzed poultry houses' dynamic heating and cooling load and verified the base room temperature and commercial thermal performance. The infiltrating of several air volumetric flow rates such as 0.2, 0.6, 1.0 and 2.0 air change per hour and the total ventilation rated on three mortality levels: 5.0, 10.0 and 15.0%. Based on this mortality rate, the annual heating and cooling load simulation was evaluated. This toolkit effectively and accurately predicts the dynamic building load for poultry houses.

Mario A. Carvajal et al. [19] studied past, present and future worldwide spatiotemporal patterns of the heat stress of cattle in 10 agroclimatic regions using thermal comfort metrics and ensemble temperature predictions. Based on research outcomes, 7% of cattle worldwide are currently exposed to high temperatures. Under a scenario of increasing greenhouse gas emissions, this share is estimated to rise 48% by 2100. The results illustrate how heat stress will affect cattle in future years and how limited time was given for implementing effective mitigation techniques.

Wariston Fernando Pereira et al. [20] developed a system for monitoring environmental conditions in poultry farms to analyze factors such as temperature, humidity, and luminosity. The sensors used for the prototype and the calibrated equipment achieved an agreement between the variables that were measured over 0.90, yet it cost only 13% below the traditional equipment. Higher efficiency occurred in data transmission speed and recording the environmental parameter. This model

provides relative humidity low-cost alternatives for monitoring the temperature and luminosity.

Uttapoln Tansawai et al. [21] introduced the occurrence and dissemination of extended-spectrum β -lactamase (ESBL)-producing *Escherichia coli* used in backyard poultry farms. Fecal samples were collected from the 27 poultry farms, and 587 samples were collected, which would account for 27.1% of ESBL-producing *E. coli*. All fecal isolates demonstrated multidrug resistance, with blaCTX-M-55 identified as the predominant gene. Other model genes were detected at low frequency. This information was collected for the presence and prevalence of ESBL-producing *E. coli* in backyard poultry.

Okinda et al. [22] suggested the machine vision-based monitoring system (MV-BMS) for monitoring the chicken to avoid the poultry farm diseases. MV-BMS utilized the video surveillance for monitoring broilers, and health status was automatically classified using the depth camera. 2D posture shape descriptors are extracted the environmental data from the poultry farm. Radial basis function-based Support 50 Vector Machine (RBF-SVM) was utilized for the diseases' classification. The RBF-SVM attained 97.5% of accuracy that outperform compared to other existing methods.

Balachandar et al. [23] developed the disease monitoring system using the Internet of Things (IoT) using imagery analytics. The system is continuously monitoring the poultry farm animals and decreases the birds' mortality rate and increases the production of the poultry farm. Using the continuous monitoring, the birds' health was predicted and provides the medicines to the birds. The designed IoT-based DL model attained high accuracy in diseases classification using the environmental data.

Based on the survey, various related works were carried out on poultry farm production prediction. Some of the disadvantages occur due to climate changes, high temperatures, humidity and bacteria affecting the poultry farm. Incomplete temperature and air quality predictions lead to lower poultry farm production. Low efficiency, poor water prediction and insufficient dataset are the limitations of the related work. Therefore, the proposed model is introduced to overcome the limitations and improve the performance.

3 Proposed Methodology

The production in poultry farms is improved based on heat stress, water and air parameters. Therefore, the automatic prediction model uses WQI, AQI and humidex index. The proposed method can effectively predict the air, water and heat parameters recorded in poultry farms and production. Figure 1 describes the architecture of the proposed model.

The proposed model initially collects the heat stress, air and water parameters; after that, the data is preprocessed using the MSN to increase the prediction

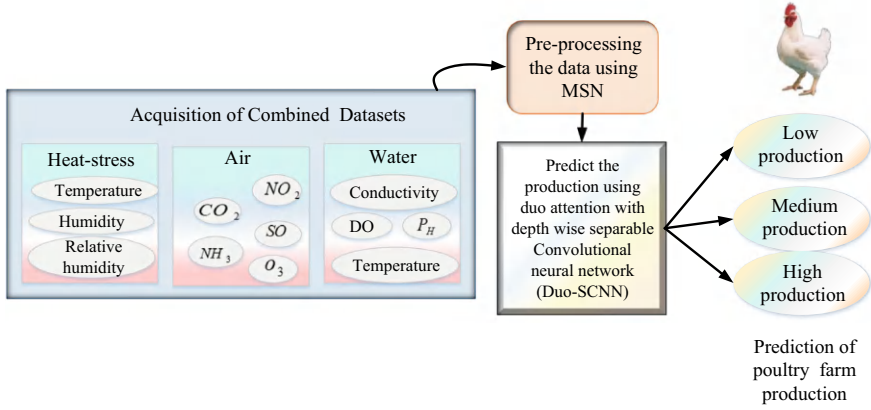


Fig. 1 Framework of proposed model

performance. Through the preprocessed data, Duo-SCNN predicted techniques effectively predict the poultry farm production. Duo attention increases the accuracy and robustness of prediction.

3.1 Data Preprocessing Using *M*-squared Normalization (MSN)

MSN is used for easy and fast implementation, which provides robustness to outlier the problems. The MSN method also applies to small, medium or large data sizes. The MSN was developed based on the basic regression models to evaluate each set M^2 to understand its effectiveness and usability.

$$Z = \frac{\hat{\mu}_m - n_i}{\hat{\sigma}_m}, \quad (1)$$

where Z represents the Z -scores, the median estimator is denoted as $\hat{\mu}_m$ and the median absolute deviation is represented as $\hat{\sigma}_m$. Ensure that the information is normalized to ensure that it converges to a distribution with a mean of 0 and a mean absolute deviation of 1, which improves the performance. It generates a multivariate linear regression model while eliminating multicollinearity.

3.2 Prediction Process Using Duo Attention with Depth-Wise Separable Convolutional Neural Network (Duo-SCNN)

An alternative to the traditional convolution structure is needed to reduce the computational cost while maintaining the expansion of the CNN network. These SCNNs classify the calculation into two stages: filtering and combining. The depth-wise convolution of each channel is utilized for the filter, while the pointwise convolution of the convolution kernel is used for combining. The outputs of the Depth-wise convolution were paired via a pointwise convolution. It is used to filter and combine the input to provide a new set of outputs in one step. This separation provides a practical parameter set and reduces the computational effort. The mathematical equation

$$W = c \times n_{\text{filter}} \times h_{\text{filter}} \times w_{\text{gilyrt}}. \quad (2)$$

Here, the number of channels is denoted as c , and the number of filters is denoted as n_{filter} . The height and width of the filter are denoted as h_{filter} and w_{gilyrt} . The parameters in SCNN are calculated, and the number of parameters is reduced compared to the original CNN.

$$W_{\text{senn}} = h_{\text{filter}} \times w_{\text{gilyrt}} \times c_{l-1} + c_l \times 1 \times 1 \times n_{\text{filter}}. \quad (3)$$

Here l is denoted as the convolutional layer, and the number of channels is represented as c . These SCNNs provide two merits: the model's generalization is enhanced, and the parameter is reduced. It is a secured model in terms of both classification performance and efficient training. Figure 2 describes the proposed Duo-SCNN architecture.

The key goal of Deep Learning is to identify the data that is most significant to the given objective from a large amount of data, and the attention process is essentially similar to the highly selective attention-seeking mechanism of humans. There are several domains in the attention mechanism based on the domain concern.

The Duo Attention Model captures feature dependencies in the spatial and channel dimensions, and the Dual Attention Model includes a self-attention mechanism. In the model, Position Attention and Channel Attention are combined, and both attentions are similar. The results of the two attention modules are combined for more effective feature descriptions for pixel-level predictions.

Feeds the feature map $F \in \mathbb{R}^{M \times G \times V}$ to the convolution layer, which provide two feature map, N and M , for positional attention, where $\{N, M\} \in \mathbb{R}^{M \times G \times V}$. Then reshape the N and M to $\{N, M\} \in \mathbb{R}^{M \times D}$, here $B = G \times V$. Then, perform matrix multiplication using the transposition of N and M , which uses the SoftMax layer to create a spatial attention map of $T \in \mathbb{R}^{M \times G \times V}$. The representation of the SoftMax layer is shown in the equation below.

$$t_{kj} = \frac{e^{N_j M_k}}{\sum_{j=1}^D e^{N_j M_k}}. \quad (4)$$

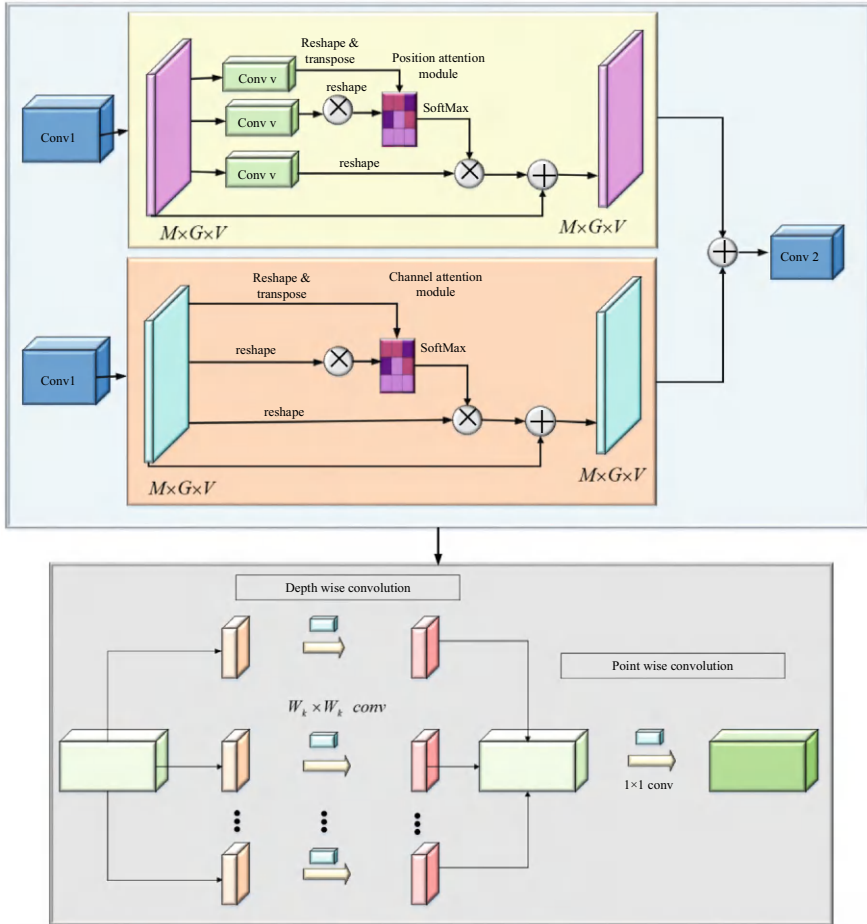


Fig. 2 Architecture of Duo-SCNN model

Here, t_{kj} calculates the j -th position impact on k -th position. The two positions are more correlated when their feature descriptions are more similar. Simultaneously, the feeding feature map F enters into a different convolution layer for producing the $H \in \Re^{M \times G \times V}$, and change H to $H \in \Re^{M \times D}$. Multiply the H and T ; then modify the outcome as $\Re^{M \times G \times V}$. α parameter is multiplied by the final output, and then the actual feature map is input into the elements summing operation to obtain the outcome such as $O \in \Re^{M \times G \times V}$, and the expression is given as follows:

$$O_k = \alpha \sum_{j=1}^D (t_{kj} H_j) + F_k. \quad (5)$$

Here, α is set as 0 and gradually learns to assign a high weight. Each position's generating feature O is the weighted average of the actual feature's overall positions. The channel attention is calculated using the channel attention map ($Y \in \Re^{M \times M}$), and F is directly used in the calculation. F is then transformed into a matrix of size $\Re^{M \times D}$, and matrix multiplication is applied to obtain the final result. After that, the SoftMax layer is applied, and the channel attention map $Y \in \Re^{M \times D}$, is given as follows:

$$y_{kj} = \frac{e^{F_j F_k}}{\sum_{j=1}^M e^{F_j F_k}}. \quad (6)$$

Here, y_{kj} denotes the effect of the j – th channel on the k – th channel. Then, do the multiplication on Y and F matrix and modify the results as $\Re^{M \times G \times V}$. Parameters β and the actual features—such as feature map element summation operation, F , are multiplied to obtain the final result.

$$O_k = \beta \sum_{j=1}^M (y_{kj} F_j) + F_k. \quad (7)$$

Here, the weight of β is gradually learned from 0. The above equation calculates the final characteristics of each channel and is helpful to increase the distinctive characteristics. By using highly discriminatory features, the accuracy and robustness of prediction in poultry farming are significantly improved.

3.3 Humidex Value Evaluation

The humidex value is denoted as the humidity index, and it is a function of finding the environment temperature and relative humidity. The humidity index defines how the climate feels for animals or people. The value of the Humidex index is related to mortality and the stress level of bird and egg production. Using the Humidex value, the stress level is classified and expressed as follows:

$$\text{Humidex\_value} = (\text{Average temperature of air in Celsius}) + l. \quad (8)$$

From the above equation,

$$l = (0.555) * (h - 10.0), \quad (9)$$

$$h = 6.11 \times \exp[5417.7530 \times ((1/273.16) - (1/DP))]. \quad (10)$$

Here, DP is represented as the temperature in Kelvin, and the water vapor is compressed as the liquid format.

3.4 Water Quality Index Evaluation

The Water Quality Index (WQI) is calculated to classify whether water is drinkable or not. Hardness, pH, nitrates, dissolved oxygen (DO), Biochemical Oxygen Demand and conductivity parameters are considered in the WQI calculation. In this paper, the WQI is calculated using the weighted arithmetic method, which is given as follows:

$$WQI = \frac{\sum R_m S_m}{\sum S_m}. \quad (11)$$

Here, R_m and S_m are denoted as quality rating and weight for every quality parameter. The quality rating is calculated using the ideal and observed values, which is calculated as follows:

$$R_m = 100 \left[\frac{l_m - l_0}{k_m - l_0} \right]. \quad (12)$$

Here, l_m and l_0 are represented the observed and ideal values. The value of the unit vector weight S_m is calculated as follows:

$$S_m = \frac{1}{k_m}. \quad (13)$$

Here, k_m denoted the recommended temperature value is suggested as 28 by the World Health Organization (WHO), so that value of S_m is 0.036. DUO value is recommended as 6 so that S_m value is 0.166. Using WQI, R_m and S_m equation, the parameter of WQI is obtained, and based on this WQI value, the grade for water is generated.

3.5 Air Quality Index Evaluation

The Air Quality Index is a broken-down linear function of pollutant level. The equation for calculating pollutant levels uses the following equation:

$$A_{\square} = \frac{A_h - A_l}{T_h - T_l} (B - B_l) + A_l. \quad (14)$$

The value of AQI is calculated by choosing the highest value among the considered A_{ij} sub-indices. AQI value is measured using the following equation:

$$AQI = \max(A_{O_3}, A_{CO}, A_{SO_2}, A_{NO_2}) \quad (15)$$

Here, A_{ij} is represented as the air quality sub-index and B is denoted as pollutant concentration. Concentration breakpoints which are greater than B is denoted as A_h and less than B is denoted as A_l . T_h and T_l represented the index breakpoints related to A_h and A_l , respectively.

Heat stress and air quality are calculated by evaluating WQI, Humidex value and AQI for water quality. These measurements are used to modify the environment of the poultry farm to make it suitable for poultry. The modified and controlled environment contributes to production improvement.

4 Results and Discussion

The evaluation outcomes of the proposed ML and DL-based predicted method for improving the production in poultry farming are discussed in this section. The proposed model performances are analyzed by comparing the proposed model and existing methods. The following sub-section describes the dataset, several performance metrics, evaluation and comparison of the proposed predicted model.

4.1 Dataset Description

The combined dataset includes various parameters for air, water [24] and heat stress [25]. For analyzing the air quality, carbon monoxide (CO), ozone (O_3), nitrogen dioxide (NO_2), ammonia (NH_3) and sulfur dioxide (SO_2) gas values are collected from the poultry farm environment. Temperature, pH, conductivity, nitrate, dissolved oxygen (DO) and Biological Oxygen Demand (BOD) values are collected from the poultry drinking water. Humidex values, mortality, relative humidity, egg production based on temperature and the number of layers in poultry farms are collected for analyzing the heat stress. These parameters are used to calculate AQI, WQI and Humidex values for the predicted production of the poultry farm.

4.2 Performance Metrics

The proposed method's performance is evaluated by comparing its performance with the performance of several existing methods using various performance metrics such

as accuracy, precision, recall, Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Square Error (RMSE).

Accuracy: Accuracy metrics calculate how often a proposed model accurately predicts the output. The mathematical model for accuracy is given as follows:

$$\text{Accuracy} = \frac{C}{A}. \quad (16)$$

Here, C and A are represented as correct prediction and overall prediction, respectively.

Precision: The recall calculates how often a proposed model correctly funded the true positive class. The precision is defined as the total number of true positive (T) predictions divided by the total number of true and false positive predictions (F). The mathematical precision equation is as follows:

$$\text{Precision} = \frac{T}{T + F}. \quad (17)$$

Recall: The precision metrics calculate how often a proposed model accurately predicts the true positive cases from all positive cases, including the true positives and false negative cases (N) in the dataset. The mathematical precision equation is as follows:

$$\text{Recall} = \frac{T}{T + N}. \quad (18)$$

MAE: The MAE is defined as the average of the complete difference between the actual and predicted outcomes in the dataset. The mathematical equation of MSE is given as follows:

$$\text{MAE} = \frac{1}{M} \sum_{k=1}^M |q_k - \hat{q}|. \quad (19)$$

Here, $\hat{q}$ and $\bar{q}$ are denoted as predicted value and the mean value of q , respectively.

MSE: The MSE is defined as the average of the squared variance between the original and predicted rates in the dataset. The mathematical equation of MSE is as follows:

$$\text{MSE} = \frac{1}{M} \sum_{k=1}^M (q_k - \hat{q})^2. \quad (20)$$

RMSE: The square root of the Mean Square Error is called RMSE, and its mathematical equation is as follows:

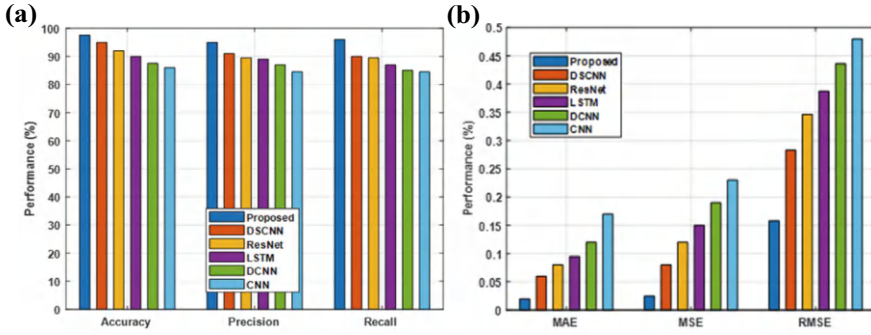


Fig. 3 Comparative analysis (a) accuracy, precision and recall (b) MAE, MSE and RMSE

$$RMSE = \sqrt{\frac{1}{M} \sum_{k=1}^M (q_k - \hat{q})^2}. \quad (21)$$

4.3 Performance Analysis:

In this paper, important performance metrics such as accuracy, precision, recall, MAE, MSE, and RMSE are used to evaluate the performance of the predicted model. The proposed model performance results are compared with various existing methods such as convolutional neural network (CNN), Deep-CNN (DCNN), long short-term memory (LSTM), Residual Neural Network (ResNet) and Dynamic segmentation-CNN (DSCNN) through the graphical representation. Figure 3 shows the performance comparison of the proposed model with existing models.

The proposed model is evaluated based on accuracy, precision and recall values, and the performance is compared to existing models such as CNN, DCNN, LSTM, ResNet and DSCNN. The proposed model achieved high accuracy (97.5%), precision (95%) and recall (96%) in the predicted model compared to existing methods. The proposed model error rate is measured using MAE, MSE and RMSE, which achieves a lower error rate in prediction. The proposed yields an error rate of 0.02% of MAE, 0.025% of MSE and 0.158% of RMSE. The proposed model achieved high performance with a low error rate through the above performance evaluation. Figure 4 describes the WQI analysis of temperature (.C) and pH.

Using water temperature and pH values, the calculated WQI and WQI values are helpful in predicting pollutant farm production. The analysis shows that if the water temperature is below 25" C, the poultry farm attained high production. Water pH is an essential measurement for analyzing water quality. If the water value is 6.4 to 7.5, the poultry farm achieves high production; otherwise, it achieves low production. Figure 5 describes the AQI analysis of NO₂ and NH₃.

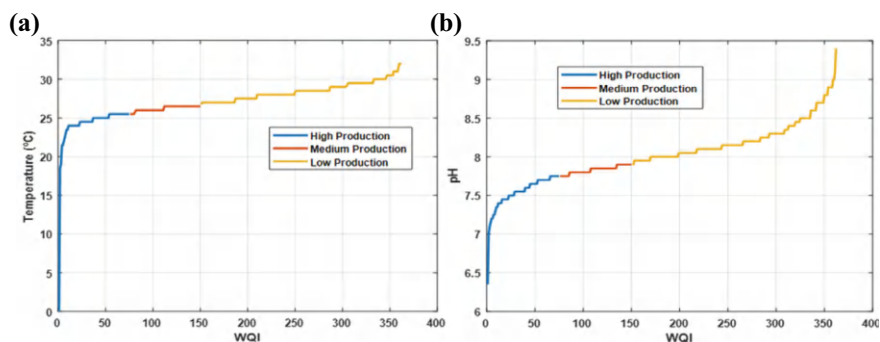


Fig. 4 WQI analysis (a) Temperature (C) (b) pH

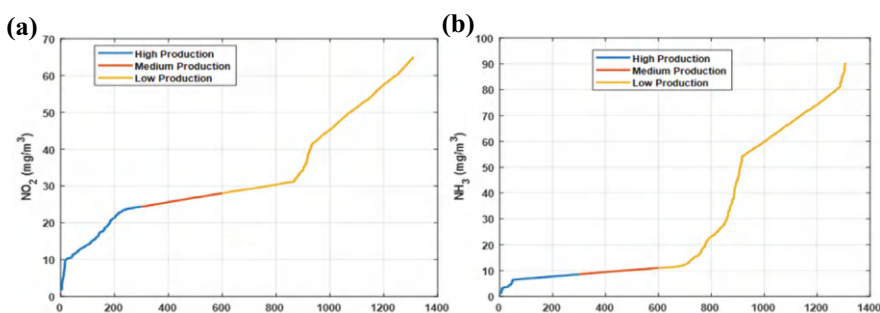


Fig. 5 AQI analysis (a) NO₂ (b) NH₃

The high level of ammonia and NO₂ decreases poultry farm production because it causes respiratory disease and blindness in poultry. The proposed model achieves high accuracy when the air contains low levels of NO₂ and NH₃. 1.8 mg/m³- 24.2 mg/m³ range of NO₂ and less than 10% NH₃ help to increase the production of the poultry farm or else provide less production. Figure 6 describes the Humidex value and egg production depending on the temperature.

The Humidex value changes depending on the temperature, and egg production also varies depending on the temperature, so heat stress is one of the most important parameters in poultry farming. If the Humidex value of the poultry farm is greater than 48, the poultry farm has achieved high production; otherwise, it has achieved medium or low production. The temperature in poultry farms is 31–37°C, resulting in better egg production; otherwise, it affects animal health and results in lower production. Figure 7 shows the confusion matrix of the proposed prediction model.

The above confusion matrix evaluates the three classes: High, low and medium. The proposed method can accurately predict the classes of humidex value compared to other systems. The proposed method accurately predicted the 97 humidex value in the high class, 97 in the medium class and 99 in the low class, so the proposed model can attain more effective humidex value prediction. Table 4 describes the comparative

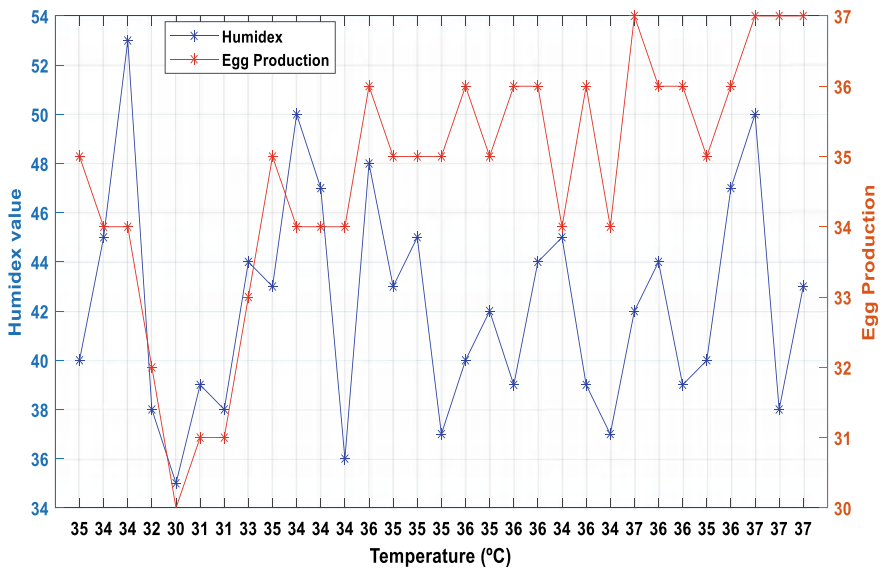
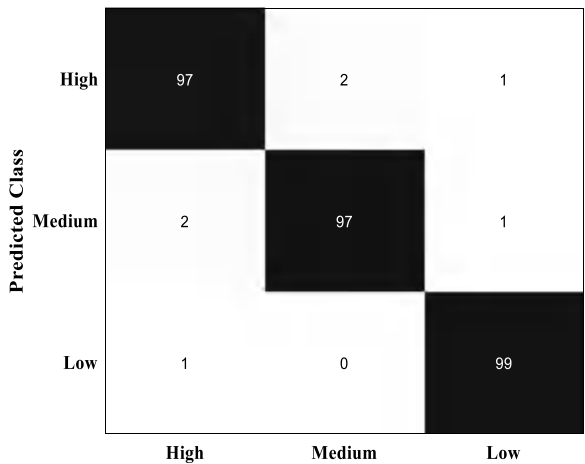


Fig. 6 Humidex value evaluation

Fig. 7 Humidex value analysis using confusion matrix



analysis of egg production prediction based on analyzing the water quality of poultry farm.

The proposed method more accurately predicted the egg production from the poultry farm compared to the other existing method such as K-Nearest Neighbors (KNNs) Algorithm, Naïve Bayes, Decision Tree and Ensemble Tree Classifier [26]. The proposed method estimates the egg prediction with 97.5% using the air, water and heat stress parameters in the poultry farm. Using this additional parameters, the performance of the proposed prediction model is increased compared to the only

Table 4 Comparative performance analysis of egg production prediction based on water quality estimation

Model	Accuracy (%)	Precision (%)	Recall (%)
KNN	93.52	93.42	88.12
Naïve Bayes	94.63	94.68	93.1
Decision Tree	95.86	97.1	94.52
Ensemble Tree Classifier	96.55	97.3	94.8
Proposed	97.5	95	96

WQI considered existing methods. So that proposed prediction model effectively recognized egg production.

5 Conclusion

In this paper, a novel prediction model for predicting the production of poultry farms was developed. The proposed model calculates the WQI, AQI and Humidex values to increase the prediction accuracy. WQI, AQI and Humidex values are measured using water, air and heat stress parameters. These parameter values are collected in the poultry farm. The proposed prediction model controls air quality, water quality and heat stress. The collected raw data is preprocessed using MSN to increase the prediction performance. Using the Duo-SCNN prediction model, the proposed model can accurately predict poultry farm production. The predicted model performance was evaluated based on accuracy, precision and recall, and it attained high accuracy, precision and recall. Also, this prediction model accurately predicted the classes such as low, high and medium prediction with less error, which was evaluated using the confusion matrix. In future, a large-scale dataset is needed to attain technology flexibility and coordination.

References

1. Mitra A, Majumder D, Mishra M, Sarkar S (2021) Poultry farming: prospects and impediments in India. *Saudi J Humanities Soc Sci* 6(6):193–198
2. Göransson L, Yngvesson J, Gunnarsson S (2020) Bird health, housing and management routines on Swedish organic broiler chicken farms. *Animals* 10(11):2098
3. Edwan E, Qassem MA, Al-Roos SA, Elnaggar M, Ahmed G, Ahmed AS, Zaqout A (2020) Design and implementation of monitoring and control system for a poultry farm. In: 2020 International conference on promising electronic technologies (ICPET), IEEE, pp 44–49
4. Bhat ZF, Morton JD, Kumar S, Bhat HF, Aadil RM, Bekhit AEDA (2022) Ultrasonication as an emerging technology for processing of animal derived foods: a focus on in vitro protein digestibility. *Trends Food Sci Technol* 124:309–322

5. Ebrahim ME, Moustafa MM, Abd El-Salam MF (2022) Biogas utilization system for warming poultry houses. *Misr J Agric Eng* 39(2):341–352
6. Deepika, Nagarathna, Channegowda (2020) Monitoring and detection of heat stress of layers in poultry farm using multiple linear regression. In: 2020 Third international conference on smart systems and inventive technology (ICSSIT), IEEE, pp 1186–1191
7. Deepika N Ensemble tree classifier based analysis of water quality for layer poultry farm: a study on cauvery river. *Age (days)* 245(200):180
8. Ranjan A, Sinha R, Devi I, Rahim A, Tiwari S (2019) Effect of heat stress on poultry production and their managerial approaches. *Int J Curr Microbiol App Sci* 8(02):1548–1555
9. Du L, Yang C, Dominy R, Yang L, Hu C, Du H, Li Q, Yu C, Xie L, Jiang X (2019) Computational fluid dynamics aided investigation and optimization of a tunnel-ventilated poultry house in China. *Comput Electron Agric* 159:1–15
10. Zheng W, Xiong Y, Gates RS, Wang Y, Koelkebeck KW (2020) Air temperature, carbon dioxide, and ammonia assessment inside a commercial cage layer barn with manure-drying tunnels. *Poult Sci* 99(8):3885–3896
11. Paura L, Arhipova I, Jankovska L, Bumanis N, Vitols G, Adjutovs M (2022) Evaluation and association of laying hen performance, environmental conditions and gas concentrations in the barn housing system. *Ital J Anim Sci* 21(1):694–701
12. Wasti S, Sah N, Mishra B (2020) Impact of heat stress on poultry health and performances, and potential mitigation strategies. *Animals* 10(8):1266
13. Elbayoumi KM, Amer MM, Rabie NS, Zaki MS (2019) Conditions affecting egg production in chickens. *Rep Opin* 11(9):8–17
14. Bist RB, Chai L (2022) Advanced strategies for mitigating particulate matter generations in poultry houses. *Applied Sciences* 12(22):11323
15. Ouakarrouch M, El Azhary K, Laaroussi N, Garoum M, Kifani-Sahban F (2020) Thermal performances and environmental analysis of a new composite building material based on gypsum plaster and chicken feathers waste. *Therm Sci Eng Prog* 19:100642
16. Meiramkulova K, Zorpas AA, Orynbekov D, Zhumagulov M, Saspugayeva G, Kydyrbekova A, Mkilima T, Inglezakis VJ (2020) The effect of scale on the performance of an integrated poultry slaughterhouse wastewater treatment process. *Sustainability* 12(11):4679
17. Chen S, Yong Y, Ju X (2021) Effect of heat stress on growth and production performance of livestock and poultry: mechanism to prevention. *J Therm Biol* 99:103019
18. Wang Y, Li B, Liang C, Zheng W (2020) Dynamic simulation of thermal load and energy efficiency in poultry buildings in the cold zone of China. *Comput Electron Agric* 168:105127
19. Carvajal MA, Alaniz AJ, Gutiérrez-Gómez C, Vergara PM, Sejian V, Bozinovic F (2021) Increasing importance of heat stress for cattle farming under future global climate scenarios. *Sci Total Environ* 801:149661
20. Pereira WF, da Silva Fonseca L, Putti FF, Góes BC, de Paula Naves L (2020) Environmental monitoring in a poultry farm using an instrument developed with the Internet of things concept. *Comput Electron Agric* 170:105257
21. Tansawai U, Walsh TR, Niumsup PR (2019) Extended-spectrum β -lactamase-producing *Escherichia coli* among backyard poultry farms, farmers, and environments in Thailand. *Poult Sci* 98(6):2622–2631
22. Okinda C, Lu M, Liu L, Nyalala I, Muneri C, Wang J, Zhang H, Shen M (2019) A machine vision system for early detection and prediction of sick birds: a broiler chicken model. *Biosyst Eng* 188:229–242
23. Balachandar S, Chinnaiyan R (2020) Internet of Things based reliable real-time disease monitoring of poultry farming imagery analytics. In: *Proceeding of the international conference on computer networks, big data and IoT (ICCBI-2018)*, Springer International Publishing, pp 615–620
24. Qian X, Lee S, Chandrasekaran R, Yang Y, Caballes M, Alamu O, Chen G (2019) Electricity evaluation and emission characteristics of poultry litter co-combustion process. *Applied Sciences* 9(19):4116

25. Schauburger G, Hennig-Pauka I, Zollitsch W, Hörtenhuber SJ, Baumgartner J, Niebuhr K, Piringer M et al (2020) Efficacy of adaptation measures to alleviate heat stress in confined livestock buildings in temperate climate zones. *biosystems engineering* 200:157–175
26. Deepika DN Ensemble tree classifier based analysis of water quality for layer poultry farm: a study on Cauvery River. *Age (days)* 245(200):180

Deep Neural Network-Based Classification of Spyware Populations Using Macroscopic Statistical Attributes of Operation Codes



Sonal Malhotra, Deepak Upadhyay, Rahul Chauhan, Arun Balodi, and Swati Devliyal

Abstract This study proposes a method for categorizing populations of malicious software using an operation code adjacency graph feature. To extract an operation code sequence and build an operation code adjacency graph from it, a piece of input software must be deconstructed. Then, a feature vector is built using this graph. To build a model for hazardous software group classification, a deep neural network is trained utilizing feature vectors from well-known group classifications. The family group of unidentified software samples is categorized using the learned model. The poor accuracy that plagues present techniques for identifying dangerous software is addressed by the suggested strategy.

Keywords Malicious software · Population classification · Operation code adjacency graph · Feature vector · Deep neural network

1 Introduction

The proliferation of spyware and other dangerous software poses a severe threat to the security of computer systems and networks. The populations of spyware must be correctly and effectively described in order to counteract these threats. Using conventional methods for classifying malware, which often depend on hand-crafted feature engineering and superficial learning algorithms, it may be challenging to detect the intricate patterns and changes present in current spyware [1].

S. Malhotra · D. Upadhyay · R. Chauhan (✉) · S. Devliyal
Department of Computer Science and Engineering, Graphic Era Hill University, Dehradun, India
e-mail: chauhan14853@gmail.com

A. Balodi
Department of Electronics and Communication Engineering, Dayanand Sagar University, Bangalore, India

Deep learning techniques, which include speech recognition, picture recognition, and natural language processing, have recently become quite successful and popular. In order to successfully complete challenging classification tasks, deep neural networks (DNNs), a subtype of deep learning models, automatically learn hierarchical representations from the raw data. Researchers are considering DNNs as a potential method for categorizing malware populations [2].

By presenting a novel approach based on DNNs that utilizes the macro-statistical properties of operation codes, this work seeks to contribute to the classification of malware. Operation codes are low-level instructions that a computer processor executes and provide crucial details about how software behaves and performs [3]. By examining the statistical properties of operation codes, we may find relevant qualities and develop a trustworthy categorization model for spyware populations.

The primary objective of this study is to assess how effectively the proposed technique can classify various forms of spyware. We evaluate the assault resistance resilience of our technique and contrast its efficacy with conventional wisdom. We assess the competitiveness and applicability of our strategy by contrasting it with the winning Kaggle competition response. This essay's remainder is organized as follows: Sect. 2 provides an overview of pertinent research in spyware categorization and deep learning techniques. The strategy, the data used, the feature extraction process, and the architecture of the DNN model are all discussed in Sect. 3. Section 4 covers the experimental design, evaluation standards, findings, and analysis. Section 5 contrasts our technique with current practices and outlines the benefits and drawbacks of it [4, 5]. The research is concluded in Sect. 6, which also outlines potential future directions for creating and using the recommended strategy.

2 Background

Spyware, or malicious software, is a type of software designed with malicious intent to harm computer structures and compromise user privacy. Malicious software can operate covertly, stealing sensitive information, gaining unauthorized control over computer structures and network resources, and undermining the integrity and usability of computers and networks. Spyware serves the interests of its controllers by facilitating illegal activities and reaping illicit benefits.

With the rapid growth of the internet and the software industry, the proliferation of spyware and its variants has reached alarming levels. These threats are pervasive, affecting a significant portion of computers worldwide. According to Microsoft statistics, approximately one-third of computers globally are infected with various types of spyware, such as trojans and backdoors [6]. The 2018 McAfee threat report reveals that the number of spyware samples has increased by nearly 34% in recent quarters, surpassing a staggering 770 million samples.

The sheer volume and diversity of malicious software pose a severe challenge to information security in various domains, including politics, economy, and culture, among others, for countries worldwide. The presence of spyware undermines

trust, disrupts critical structures, and exposes sensitive data to unauthorized access. Addressing this escalating threat requires robust and effective procedures to identify and classify different types of malicious software populations accurately. The accuracy of current approaches for identifying the population of harmful software is typically subpar, which decreases their efficacy in decreasing the risks that spyware poses. Therefore, new approaches that might increase the precision and efficacy of spyware population identification are urgently needed to meet the growing danger of harmful software [7, 8]. Malicious software is fast developing new techniques including distortion, polymorphism, and confusion that enable spyware programs to evade detection and pose a growing threat to network security.

These techniques allow malware to maintain their original functionality, behavior, and harmful intent while generating a large range of diverse spyware variants by altering the binary content of computer software. According to AV-TEST research, approximately 800 million spyware samples were discovered in 2018, a 28-fold increase from the preceding ten years. More than half of the known spyware variants are present in these samples. As a result, the detection and classification of spyware variants have become key areas of research in the field of network information security. The process involves two main phases: spyware detection and spyware classification. Spyware detection aims to determine whether an unknown sample is malicious, utilizing detection structures. Once identified as malicious, the sample is classified into the appropriate spyware family within a spyware classification structure, enabling researchers to study the evolutionary development of the spyware [9, 10]. This approach proves effective for analyzing large-scale collections of malicious software.

However, the sheer volume of spyware samples discovered today calls for efficient procedures to study spyware population classification. Traditional antivirus software relies on signature-based detection, using unique hexadecimal code strings specific to each type of spyware. Yet, spyware authors exploit automated tools to modify and expand existing spyware, generating tens of thousands of variants from a single piece of spyware. Consequently, signature-based procedures often struggle to detect and classify such spyware variants effectively [11, 12].

3 Research Methodology

The research methodology focuses on addressing the challenges posed by spyware belonging to the same family, which can exhibit similarities in function, properties, and code structure due to techniques like obfuscation, morphing, polymorphism, or refinement. Additionally, executable files may display varying timing and spatial dependencies across different levels of abstraction.

To classify groups of malicious software effectively, the proposed procedure utilizes the attributes of operation code adjacency graphs, which capture the similarities in frequency and spatial distribution of disassembled codes within the same family. The methodology involves the following steps:

- Expressing the disassembled code's operation code sequence as operation code adjacency graphs.
- Converting the operation code sequence into a characteristic vector representation.
- Training a malicious software group classification model using a deep neural network.
- Utilizing the trained model to classify malicious software into their respective groups based on the operation code adjacency graphs.

By leveraging the deep neural network-based classification model and the insights gained from the operation code adjacency graphs, this approach aims to improve the accuracy of grouping and classifying malicious software.

The methodology proposed for classifying the population of malicious software based on the operation code adjacency graph attributes, involves the following steps:

A malicious software input sample with many software groups is processed through disassembly in **step S1**. This method has to be used to extract the disassembled codes from the malicious software sample. It's critical to keep in mind that the sample of malicious software under discussion may either be a sample intended for classification or a collection of malicious software training samples.

The mentioned disassembly is carried out using IDA Pro, an interactive disassembler program. The disassembled codes from the malicious software sample that was supplied as input are extracted using the powerful piece of software known as recursive descent disassembler. IDA Pro fixes the issues brought on by recursive degradation by appropriately identifying the data and isolating it from the code. It also enhances the disassembled code by adding relevant variable and function names discovered during analysis, providing the user with a substantial quantity of symbolic data. The approach greatly improves code comprehension, and the given disassembled code nearly resembles the original source code.

The process for deriving an operation code sequence from the disassembled codes of the software sample samples is the subject of the **step S2** investigation.

In **step S3**, the operation code sequence is transformed into an adjacency graph. The assembly-based operation codes are categorized depending on their intended uses. These categories include instructions for floating-point operations, general data transfers, input/output ports, destination addresses, sign instructions, arithmetic operations, logic operations, string instructions, program transfers, and other instructions.

To construct the operation code adjacency graph, three groups of operation codes corresponding to input/output port transfer instructions, destination address transfer instructions, and sign transfer instructions are merged into a single group. Additionally, the remaining nine groups of operation codes corresponding to stack access instructions, general data transfer instructions, arithmetic operation instructions, logic operation instructions, string instructions, program transfer instructions, processor control instructions, and floating point operation instructions are divided into ten operation code groups.

Each operation code group is represented as a node in the operation code adjacency graph. The weight of each node is determined by the frequency of occurrence of all

operation codes within that group in the operation code sequence. The distribution frequency of operation codes varies among different malicious software families, making it possible to distinguish between them based on these frequencies.

In the operation code sequence, when two nodes have adjacent operation codes (i.e., a front-back adjacency), the frequency of operation code adjacency between the two nodes is incremented by 1. The frequency of operation code adjacency corresponds to the number of adjacency relations present in the operation codes of both nodes. Additionally, based on the order of operation codes in the operation code sequence of the two nodes, the adjacent direction of the operation codes between them can be determined. If the operation codes have a front-back adjacent relation, the adjacent direction points from the node with the front operation code to the node with the back operation code.

By representing the adjacent direction of operation codes between nodes using arrows, a directed adjacent edge is established between two nodes. The frequency of operation code adjacency between the nodes serves as the weight for the directed adjacent edge. By combining the node weights, an operation code adjacency graph for the software sample is obtained. The corresponding operation code adjacency graph, is depicted in Fig. 1.

Step S4 involves transforming the operation code adjacency graph into a feature vector. This feature vector captures the frequency of each group of operation code sequences in the sample, as well as the adjacency frequency among the groups

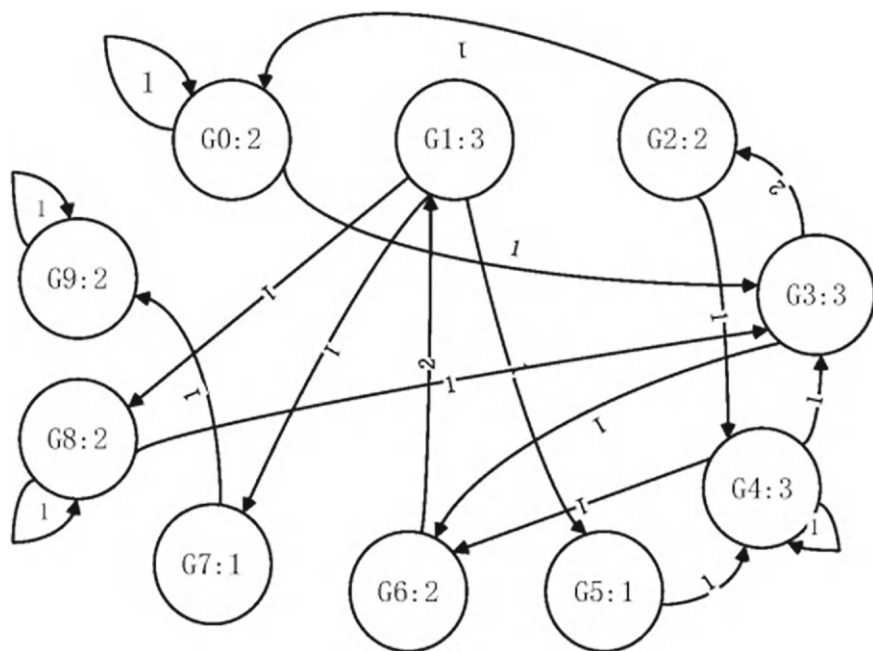


Fig. 1 Code adjacency graph

of operation codes. The resulting feature vector comprises 110 features, which include the node weights of the 10 nodes and the node adjacency weights of the 100 directed adjacency edges. In cases where directed adjacency edges do not exist, their corresponding feature values are set to 0.

Moving to Step S5, the known software within the software sample is utilized as a training sample. The feature vectors of the training sample are fed into a deep neural network (DNN) to train a malicious software population classification model. DNN refers to neural networks with multiple hidden layers, also known as deep feedforward networks or multi-layer perceptrons. It consists of input layers, hidden layers, and an output layer. The layers are fully connected, allowing for information flow between neurons. The DNN employs forward propagation and backpropagation algorithms to calculate outputs and iteratively optimize the loss function, respectively. Dropout regularization is often applied to prevent overfitting by randomly removing neurons from hidden layers. This regularization enhances the generalizability of the model.

Lastly, in Step S6, unknown software within the software sample is treated as a test sample. The feature vector of the test sample is fed into the malicious software population classification model based on the deep neural network, enabling the classification of the test sample's population.

4 Experimental Data and Experimental Environment

The experimental data used in this research was obtained from the Microsoft Spyware Classification Challenge dataset, which was initially released in 2015 as part of a Kaggle contest. Since then, it has been widely utilized by researchers in over 50 studies, establishing itself as the benchmark dataset for spyware population classification.

The dataset consists of 10,868 spyware samples belonging to 9 different spyware families. Each sample comprises two files: one containing hexadecimal code and the other containing disassembled code. To ensure the safety of the files, Microsoft removed the PE header. Among the samples, 149 were found to have incorrect disassembled code, resulting in a total of 10,719 correctly disassembled samples used in the experiments described in this study. The distribution of the samples across different spyware families is presented in Table 1. Due to the nature of the dataset, it represents a challenging multi-category classification task, characterized by an extremely imbalanced sample distribution.

For the implementation of the research methodology, a Keras DNN model with TensorFlow as the backend was employed. Keras is an open-source neural network library written in Python, known for its modularity, extensibility, and user-friendliness. It simplifies the implementation of complex algorithms and enables researchers to quickly translate their ideas into tangible results.

Table 1 Microsoft spyware classification challenge dataset

Family number	Family name	Number of samples (Pieces)	Proportion of total sample (%)
1	Ramnit	1519	14.21
2	Lollipop	2578	23.2
3	Kelihos ver3	2838	27.2
4	Injury	418	3.97
5	Simda	44	0.29
6	Tracur	747	6.85
7	Kelihos left	384	3.65
8	Obfuscator.ACY	1188	11.29
9	Gatak	1015	9.34
Total		10,731	100

5 Evaluation Metrics and Methodology

The classification performance of the model is assessed using two main metrics: accuracy and log loss. Accuracy measures the proportion of correctly predicted samples out of the total number of predictions. However, accuracy alone may not sufficiently capture the robustness of the predictions, and therefore, log loss is also employed. Log loss, also known as cross-entropy loss, quantifies the discrepancy between predicted probabilities and the true categories. Minimizing log loss is equivalent to maximizing the accuracy of the classifier, with a perfect classifier achieving a log loss value of 0. The calculation of the log loss function is as follows:

$$L = -N \sum_i \log(p_{ij}) = -N \sum_j \sum_i \log(p_{ij}) = -N \sum_j y_{ij} \log(p_{ij})$$

Here, Y represents the output variable, which denotes the detection result of the software being evaluated. X is the input variable, representing the binary executable file of the software. L denotes the loss function, N is the number of test samples (binary executable files), Y_{ij} is a binary index indicating the class (benign or malicious) corresponding to the i th test sample, p_{ij} represents the probability that the i th test sample belongs to class j , and M is the total number of classes (in this case, $M = 2$ for benign and malicious).

Additionally, the performance of the classifier can be evaluated using a receiver operating characteristic (ROC) curve. The ROC curve plots the true positive rate (detection rate) against the false positive rate as the detection threshold varies. The area under the ROC curve (AUC) is a comprehensive metric for evaluating the classifier's performance. The AUC value ranges between 0.5 and 1.0, with a larger AUC indicating better classifier performance.

Table 2 DNN Hyperparameter Tuning Results

Hyperparameter	Parameter option or range	Better value
Layer-1,2,5,6 The number of neurons in the layer	(258, 322, 386, 446, 514)	386
Layer-3 layer neuron number	(258, 322, 386, 446, 514)	322
Layer-4 the number of neurons	(258, 322, 386, 446, 514)	514
Activation	(softplus, softsign, relu, tanh, sigmoid)	relu
Dropout	(0.2, 0.4, 0.5, 0.6, 0.7, 0.8)	0.4
Optimizer	(SG, RMSprop, Adagrad, Adam)	Dosing
Batch size	[20, 40, 60, 80, 90, 100)	70
epochs	[0, 400]	180

5.1 DNN Hyperparameter Tuning

In machine learning models, manually adjustable parameters are referred to as hyperparameters. The performance of a DNN model is heavily influenced by these hyperparameters, and improper selection can lead to issues such as underfitting or overfitting. GridSearchCV is a commonly used technique in the scikit-learn library for searching optimal hyperparameters for a model. GridSearch and CV stand for grid searching and cross-validation, respectively. GridSearchCV applies cross-validation to structurally adjust parameters within a specified range. It trains multiple models using different parameter combinations and selects the one with the highest accuracy on a validation set.

In the present embodiment, GridSearchCV is employed to search and fine-tune the hyperparameters of the DNN model, resulting in a malicious software population classification model based on deep neural networks. The debugging results are summarized in Table 2.

6 Experimental Results and Analysis

The training process of the deep neural network (DNN) relies on gradient descent, which searches for the direction of fastest descent in function value and iteratively updates the weights to reach a local optimum solution. The number of epochs, representing the number of times the entire training dataset is used, plays a crucial role in the performance of the DNN model. In this experiment, the optimal epoch value was determined by training the model for 300 iterations with 80% of the samples and testing on the remaining 20%.

The accuracy and log loss curves of the DNN-based spyware population classification model were analyzed. As shown in Table 3. It was observed that the training and test accuracy rapidly increased and the log loss rapidly decreased when the epoch

Table 3 Epochs

Epoch value	Training accuracy	Test accuracy	Training log loss	Test log loss
0	–	–	–	–
30	Rapid increase	Rapid increase	Rapid decrease	Rapid decrease
150	Slight increase	Stable	Slight reduction	Stable
Beyond 150	Relatively stable	Relatively stable	Relatively stable	Relatively stable

value ranged from 0 to 30. From epoch 30 to 150, there was a slight increase in accuracy and a small reduction in log loss, while beyond epoch 150, the accuracy and log loss remained relatively stable. Based on the combined analysis of accuracy and log loss curves, the optimal epoch value was determined to be 150. It was noted that as the training iteration increased, there was no decrease in test accuracy or significant increase in log loss, indicating that the DNN-based model did not suffer from overfitting. The operation code adjacency graph extracted by the proposed procedure demonstrated stable and effective representation attributes.

After determining the optimal epoch value as 150, a tenfold cross-validation test was performed. The DNN-based spyware population classification model achieved an accuracy of 98.44% and a log loss of 0.0927. The confusion matrix and normalized confusion matrix demonstrated the model's performance in classifying 8 categories. The overall classification accuracy was high, with a slightly lower accuracy for the class 5 malicious software family (86%). Considering the limited number of samples (42) and the training difficulty, this result was still commendable. Notably, the three spyware families of category 4, category 6, and category 7, despite having a smaller number of samples (415, 745, 381, respectively), achieved high classification accuracy of 98% and 99%, respectively.

The ROC curve (Fig. 8) of the DNN-based spyware population classification model showed the relationship between the detection rate and false alarm rate at different detection thresholds. The curve was closer to the upper left corner, indicating superior classifier performance. The AUC value of 0.998 further confirmed the model's effectiveness, as it approached the optimal AUC value of 1.

In summary, the DNN-based spyware population classification model proposed in this study demonstrated excellent performance metrics, making it an effective model for spyware classification.

7 Comparison of Results

Several existing procedures for spyware population classification have been proposed in the literature. E. raff et al. utilized a shallow convolutional neural network structure by converting binary content into hexadecimal representation and mapping it to word vectors using word2vec. They achieved classification based on this vector space representation. Similarly, Gibert et al. employed a deeper 9-layer neural network

structure with convolution, pooling, and fully connected layers, treating spyware binary files as grayscale images for classification. Another approach by Gibert et al. involved dividing binary files into fixed-size code blocks, calculating entropy for each block, and using a convolutional neural network for classification. They also utilized denoising auto-encoders and DRN networks for spyware population classification based on entropy sequences. McLaughlin et al. performed disassembly of binary files, extracted operation code sequences, and used a shallow convolutional neural network for spyware detection after converting the sequences into word vectors using word2vec.

In order to evaluate the performance of the proposed procedure in this study, a comparison was made with the aforementioned mainstream procedures for spyware population classification. The comparison was based on two evaluation metrics: Classification accuracy and logarithmic loss. The results of this comparison are presented in Table 4.

The approach recommended in this study is compared with the tactic used by the team that won the Kaggle competition. The winning team had a logarithmic loss of 0.00283 and an accuracy of more than 99% in tenfold cross-validation. However, their approach used a highly advanced ensemble learning model, accounting for elements like function names, bytecode 4 g, operation code ngrams, the volume of disassembled code segments, and images generated by disassembled code. The substantial amount of raw data required to build these features resulted in a 4 GB feature dataset. A further 24 h were needed for the training of the final integrated learning model, making a total of 72 h. It took roughly 48 h only to extract the features. Furthermore, this model was created exclusively for the competition dataset and not to classify people in general. In comparison, the strategy suggested in this study offers a quick model training time of 93 s and a small feature file size of just 2.89 MB. It uses a single type of characteristic based on the idea that operation codes belonging to the same family of malicious software are similar in terms of frequency and spatial distribution. This method provides a thorough and useful framework for categorizing populations of harmful software.

Table 4 Comparison of classification results for different spyware population classification procedures

Detection procedure	Accuracy	Log loss
MalConv deepconv	0.9741	0.3171
CNN IMG	0.9656	0.2844
CNN entropy	0.9808	0.1446
Entropy + DSN	0.9761	0.1163
Opcodes + CNN	0.9502	0.1175
Procedure of the embodiment of the invention	0.9461	0.217

8 Conclusion

In conclusion, this study provides a novel deep neural network-based approach for classifying populations of dangerous software. The suggested approach makes use of the macroscopic statistical properties of operation codes to extract relevant features and achieve reliable classification performance.

The results of the experiments support the effectiveness of the recommended approach. It distinguishes between distinct forms of spyware with high accuracy and little logarithmic loss, outperforming the bulk of current approaches. Although it falls short of a sophisticated ensemble learning model in terms of sophistication, it offers a more practical and helpful solution with a more manageable temporal complexity.

The proposed approach is contrasted with the tactic used by the team that won the Kaggle competition. Despite outperforming the competition, the winner team's approach was built on an extremely intricate ensemble learning model that was especially created for the competition dataset. The recommended strategy, on the other hand, provides a comprehensive and efficient approach while using a little feature file size and achieving short model training times.

The recommended methodology addresses a crucial problem with deep learning approaches: their sensitivity to attack resistance. It is based on the robust statistical properties of operation codes. This makes it more resistant to classification errors caused by minute changes in the input samples.

Overall, the study provides a reliable and practical deep neural network-based technique for classifying dangerous software. The findings highlight the importance of include macroscopic statistical data in feature extraction and demonstrate the potential for achieving high-performance classification results in real-world settings. The focus of future study may be on strengthening the procedure's resilience and examining its potential use in further spyware analysis and cybersecurity fields.

References

1. Dahiya D (2023) COVID-19 disease prediction utilizing dilated convolution neural network based levy flight tunicate swarm optimization. *Wireless Pers Commun.* <https://doi.org/10.1007/s11277-023-10505-1>
2. Bhandari G, Lyth A, Shalaginov A, Grønli T (2023) Distributed deep neural-network-based middleware for cyber-attacks detection in smart IoT ecostructure: a novel framework and performance evaluation approach. *Electronics* 12(2):298. <https://doi.org/10.3390/electronics12020298>
3. Aydogan M (2023) A hybrid deep neural network-based automated diagnosis structure using x-ray images and clinical findings. *Int J Imaging Struct Technol.* <https://doi.org/10.1002/ima.22856>
4. Lu N, Li D, Shi W, Vijayakumar P, Piccialli F, Chang V (2021) An efficient combined deep neural network-based spyware detection framework in 5G environment. *Comput Netw* 189:107932. <https://doi.org/10.1016/j.comnet.2021.107932>

5. Kim H, Park J, Kwon H, Jang K, Seo H (2021) Convolutional neural network-based cryptography ransomware detection for low-end embedded processors. *Mathematics* 9(7):705. <https://doi.org/10.3390/math9070705>
6. Singh J, Sharma K, Wazid M, Das AK (2023) SINN-RD: spline interpolation-envisioned neural network-based ransomware detection scheme. *Comput Electr Eng* 106:108601. <https://doi.org/10.1016/j.compeleceng.2023.108601>
7. Zhang X, Wu K, Chen Z, Zhang C (2021) MalCaps: a capsule network based model for the spyware classification. *Processes* 9(6):929. <https://doi.org/10.3390/pr9060929>
8. Forghani M, Khachay M (2020) Convolutional neural network based approach to in silico non-anticipating prediction of antigenic distance for influenza virus. *Viruses* 12(9):1019. <https://doi.org/10.3390/v12091019>
9. Chai Y, Du L, Qiu J, Yin L, Tian Z (2023) Dynamic prototype network based on sample adaptation for few-shot spyware detection. *IEEE Trans Knowl Data Eng* 35(5): 4754–4766 2023. <https://doi.org/10.1109/TKDE.2022.3142820>
10. Nguyen DV, Nguyen GL, Nguyen TT et al (2022) MINAD: multi-inputs neural network based on application structure for android spyware detection. *Peer-to-Peer Netw Appl* 15:163–177. <https://doi.org/10.1007/s12083-021-01244-w>
11. Kolli S, Balakesavareddy P, Saravanan D (2021) Neural network based obfuscated spyware detection. In: 2021 international conference on structure, computation, automation and networking (ICSCAN), Puducherry, India, pp 1–5. <https://doi.org/10.1109/ICSCAN53069.2021.9526496>
12. Tiwari P, Upadhyay D, Pant B, Mohd N (2022) Multiclass classification of disease using CNN and SVM of medical imaging. In: Singh M, Tyagi V, Gupta PK, Flusser J, Ören T (eds) *Advances in computing and data sciences. ICACDS 2022. Communications in computer and information science*, vol 1614. Springer, Cham. https://doi.org/10.1007/978-3-031-12641-3_8

Exploring the Effectiveness of Sequence Embedding as a Powerful Tool for Unmasking Deceptive Tactics and Enhancing Insurance Fraud Detection



G. L. Vidyashree, Y. Umadevi, R. Kanagavalli, Kumaraswamy, and M. R. Sowmya

Abstract The detection of insurance fraud poses a significant challenge for the insurance industry. To address this issue, this study leverages the power of XGBoost, a scalable machine learning algorithm, to predict fraudulent insurance claims. Utilizing a dataset containing policy details, customer information, and accident-related data, XGBoost was employed to develop a robust fraud detection model. Through exploratory data analysis, feature engineering, and model training, XGBoost demonstrated its effectiveness in identifying fraudulent claims while minimizing false positives. Evaluation metrics, including precision, recall, F1-score, and AUC, highlighted the model's performance in fraud detection. Additionally, class imbalance handling and the impact of specific variables on fraud tendencies were explored. This research offers insurance companies a valuable tool for enhancing fraud detection, leading to cost savings and improved security in the industry.

Keywords Machine learning · XGBoost · Feature engineering · Evaluation metrics

G. L. Vidyashree (✉)

Department of CSE Government Polytechnic, Ramanagara, India

e-mail: vidyavaishaka@gmail.com

Y. Umadevi

Department of CSE Government Polytechnic, Channapatna, India

R. Kanagavalli · Kumaraswamy

Department of CSE, Global Academy of Technology, Bengaluru, India

M. R. Sowmya

Department of CSE, Nitte Meenakshi Institute of Technology, Bengaluru, India

1 Introduction

Insurance fraud is a purposeful act of deception intended to generate financial gain for individuals, businesses, or agents at the expense of insurance carriers. Due in large part to the rise in fraudulent insurance applications, this problem has become much worse in recent years, burdening communities with skyrocketing premium costs. Insurance fraud is a complex issue that involves more than just misleading insurance companies; it also includes unauthorized use of payment cards, such as credit or debit cards, as sources of funds in transactions. Such behaviors can have a variety of justifications, from obtaining items without making a purchase to making unlawful withdrawals from accounts. The complexity of the problem is further increased by the frequent connections between identity theft and insurance fraud.

The financial and insurance industries are both severely harmed by fraud, which results in large expenses and losses. These fraudulent operations take many different forms, ranging from false credit card transactions to insurance fraud, and they place a heavy burden on these industries as a whole. To put this in perspective, authorities in the field predict that roughly 10% of all losses and loss adjustment costs that the insurance sector faces each year can be directly related to false claims [1]. Within these industries, the necessity of fraud detection develops as a crucial and necessary task. Particularly in the area of claims management, it stands for a fundamental and core expertise woven into the fabric of their operations.

The importance of fraud detection is highlighted by its role in determining and reducing the impact of fraud, not only on the overall ecosystem but also on the financial bottom line. These sectors can maintain the integrity of their business operations, control expenses, and maintain the faith and confidence of their customers by successfully identifying and correcting fraudulent claims and transactions.

There are a number of particular difficulties that should be carefully considered when analyzing fraudulent activity using statistical and machine learning techniques [2]. These difficulties stem directly from the features of data and fraud detection procedures used in the finance and insurance industries.

Unstructured Claims Data: [3] The format of claims data, which frequently takes the form of unstructured or semi-structured data, presents one of the main challenges. Traditional machine learning algorithms, which normally demand structured input for successful processing, can find this unstructured nature particularly frustrating. Unstructured data contains text that is not in any particular format, graphics, or other sorts of data that are difficult to standardize and analyze using conventional methods. These data types might have essential information for fraud detection, but getting useful insights from them requires specific methodologies.

1.1 Data Distribution Inequality

The class imbalance that exists by default in datasets for fraud detection is another key challenge. In these datasets, the fraction of fraudulent cases is disproportionately tiny relative to the proportion of genuine, non-fraudulent cases. Every fraudulent case is, in essence, a dataset anomaly.

Machine learning algorithms may become biased toward the majority class (non-fraudulent cases) as a result of this class imbalance, which will affect how well they learn. As a result, it is more difficult to detect fraudulent cases, and there is a chance that the rate of false-negatives will increase.

Variable Claim Lengths: Insurance company claims do not follow a predetermined format or length. The quantity of things billed or the information included in a claim can differ greatly from instance to case.

For fraud detection methods, which must take into account and adjust to this inherent flexibility in the data, the variation in claim lengths presents an extra hurdle.

The classification algorithms chosen and the performance criteria employed to assess the efficacy of fraud detection models are significantly influenced by these unique features of claims data. Traditional fraud detection techniques have been shown to be incredibly inaccurate and unreliable. This has attracted the attention of the machine learning and data analytics community, igniting efforts to solve this urgent issue.

To overcome these obstacles, a multidisciplinary strategy combining domain knowledge, sophisticated data preparation methods, and cutting-edge machine learning algorithms is essentially necessary. By doing this, businesses in the financial and insurance sectors can improve the security and credibility of their operations while also strengthening their fraud detection skills.

In this study, a novel method for accurately predicting and categorizing fraudulent auto insurance claims using the Extreme Gradient Boosting algorithm, also known as XGBoost is used. It is interesting that, to the best of our knowledge, no prior study has used XGBoost or any other machine learning algorithm for categorizing claims into different fraud classes, even though earlier studies have investigated the application of machine learning in auto insurance fraud detection. XGBoost has proven to be incredibly adaptable in tackling numerous problems in a variety of industries, including cybersecurity and health [4, 5].

The research's findings demonstrate that XGBoost is superior to other machine learning techniques for detecting and categorizing vehicle insurance fraud. The results highlight the potential of XGBoost as a helpful tool for improving the precision and effectiveness of fraud detection systems in the insurance sector, offering more accurate identification and categorization of fraudulent claims.

The remaining sections of this article are structured as follows: A complete literature study on auto insurance fraud detection and categorization is included in Sect 2. Section 3 describes our innovative methodology's essential components, including feature engineering, model selection, and data pre-processing. In Section 4, we go into greater detail on the suggested machine learning technique, Extreme Gradient

Boosting (XGBoost), emphasizing its benefits and applicability to our study. In Section 5, the numerical findings from our trials are shown, demonstrating the superior effectiveness of our method in precisely recognizing and categorizing fraudulent vehicle insurance claims. Finally, Sect 6 presents a summary of our results and discusses the implications and potential future directions of our study in relation to fraud detection in the insurance sector.

2 Literature Review

Due to the significant expenses and losses that fraudulent claims have caused the insurance business, insurance fraud detection has recently become a hugely important topic. This increase in interest reflects the urgent need to confront the numerous types of fraud that jeopardize the financial stability and integrity of the insurance business. A rising corpus of literature focusing on cutting-edge methods, data analytics, and cutting-edge technologies to improve fraud detection capabilities in the insurance arena has evolved to address this problem.

In 2020, Sharmila Subudhi and Suvasini Panigrahi [6] in their article introduced an inventive hybrid approach designed for the detection of fraudulent activities within the domain of automobile insurance claims. Their methodology seamlessly integrated Genetic Algorithm (GA)-based Fuzzy C-Means (FCM) clustering with a diverse set of supervised classifier models. The initial phase involved the extraction of a test dataset from the original insurance dataset, while the remainder of the training dataset underwent a clustering process aimed at effective under-sampling and the creation of coherent clusters. Subsequently, the test instances were meticulously categorized into three classes: Genuine, malicious, or suspicious, based on the patterns identified within the clusters. This categorization led to the exclusion of genuine and fraudulent records, leaving behind the cases that warranted further scrutiny. The researchers employed four distinct classifiers for this analysis, namely Decision Tree (DT), Support Vector Machine (SVM), Group Method of Data Handling (GMDH), and Multi-layer Perceptron (MLP), each assessed individually for their effectiveness in fraud detection. The study consistently utilized a tenfold cross-validation methodology for robust model training and validation, ultimately showcasing the remarkable efficacy of their hybrid system. This effectiveness was substantiated through a series of comprehensive experiments conducted on a real-world automobile insurance dataset.

In 2018, Waghade and Karandikar conducted a study utilizing machine learning models to forecast instances of fraud within the healthcare sector. Their research underscored the inherent costs associated with the conventional manual identification process, which necessitates extensive efforts by auditors to assess the legitimacy of medical insurance claims. They emphasized the significance of automated decision support systems in addressing various forms of fraud prevalent within this industry.

An automated approach was established in 2018 by Severino, Matheus Kempa et al. [7] to detect fraud in shipping documents, which are frequently manipulated

to dodge restrictions or aid in smuggling. They used Bayesian networks as the foundation of their strategy, building probabilistic discriminative models to forecast the existence of particular products in cargo shipments and comparing those forecasts to the supporting documentation to detect fraud. According to their study's findings, the system significantly outperformed random audits, which are labor-intensive and impractical for shipping companies to use for document inspection. This highlights the system's potential to improve fraud detection in the shipping sector.

In 2018, Yao et al. [8] introduced a novel model designed for the detection of financial fraud, which ingeniously integrated feature selection techniques with machine learning classification models. Their study tackled the challenge of high-dimensional data by employing Principal Component Analysis (PCA) and the Extreme Gradient Boosting algorithm (XGBoost) to identify the most informative variables within the dataset. Subsequently, several machine learning models were applied to the refined feature set, with rigorous evaluation methods employed to assess their effectiveness. Remarkably, among the various models tested, the Random Forest algorithm emerged as the top performer, showcasing superior out-of-sample performance in identifying instances of financial fraud. This innovative approach represents a significant contribution to the ongoing efforts to enhance fraud detection methods within the financial sector, aligning with the growing importance of mitigating fraudulent activities in the realm of finance.

By using Long Short-Term Memory (LSTM) recurrent neural networks for sequential learning and contrasting their findings with a static benchmark, the Random Forest classifier, Jurgovsky et al. [9] proposed a ground-breaking method for detecting credit card fraud in 2018. Their research, which used real-world data and independent evaluations of offline and online transactions, showed that the frauds picked up by these two techniques had persistent differences, pointing to the possibility of ensemble-based models that include both methodologies. Furthermore, a key finding from their research was that manual feature aggregations significantly improved performance for both static and sequential learners. This emphasized the crucial part modeling considerations and feature engineering play in improving fraud detection efficacy and provided insightful information into the dynamic fraud detection methodology landscape.

In their paper, Dhieb et al. [10] proposed a unique method for spotting fraud in insurance contracts that makes use of the Extreme Gradient Boosting (XGBoosting) algorithm. Their approach centered on fulfilling online real-time requirements and boosting security using a combination of AI approaches and blockchain architecture. Data pre-treatment and cleansing came first in the pipeline, and then data visualization for insights. Personal information was kept private to safeguard data privacy. Finally, they used XGBoosting to create a prediction model that calculated the probability of future fraud based on the information at hand. For online learning solutions, an effective Decision Tree algorithm was also developed. Comparative investigation proved that the suggested approach performed better than other machine learning classifiers, notably in terms of accuracy, such as Decision Tree, Naive Bayes, and Nearest Neighbor.

Liu et al. [11] introduced an innovative technique for addressing insurance fraud detection within imbalanced datasets. Their approach involved data partitioning through under sampling, both with and without replacement, within the majority class, followed by merging with the minority class. They employed tenfold cross-validation for rigorous testing. The core idea was to select the most effective data partition under sampling method. For insurance fraud detection, they employed Support Vector Machines, Decision Trees, and artificial neural networks as models. Experiments were conducted using a publicly available dataset containing records of various automobile insurance claims. Notably, the results indicated that Decision Trees outperformed all other classifiers, showcasing superior accuracy. Overall, their technique demonstrated significant improvements over previous work and achieved the highest accuracy in insurance fraud detection.

2.1 Drawback of the Existing System

The wide variety of fraud patterns and the relatively low prevalence of known fraud instances within normal datasets make it difficult to detect insurance fraud. The potential benefits from preventing losses and the expenses involved with generating false warnings must be balanced when developing fraud detection models. In this context, machine learning approaches have proven to be useful resources with the potential to considerably improve forecast accuracy. In turn, this gives loss control units inside insurance companies the ability to increase their coverage while still having low rates of false positive alerts. By utilizing machine learning, insurance companies may more successfully spot and fight fraud, thereby securing their business processes and financial resources.

Current systems for detecting insurance fraud rely on manual procedures, which means users must keep an eye on their financial transactions for symptoms of fraud. This method has a key flaw in that it lacks automated systems to detect fraudulent transactions. Users run the major risk of being ignorant of financial losses without automated detection. The current system also uses the decision tree algorithm, which is less efficient than more sophisticated techniques like Random Forest and XGBoost. Each dataset cell is processed independently by the decision tree method, which leads to lengthy and time-consuming operations.

This manual approach has a number of drawbacks, including high costs, limited scalability to accommodate growing data volumes, susceptibility to fraud due to its reactive nature, and little transparency in how fraud detection choices are made. In order to overcome these difficulties, there is an urgent need for more automated, effective, and scalable insurance fraud detection solutions that offer real-time protection, reduce false positives, and guarantee transparency in decision-making processes. These solutions should also make use of cutting-edge algorithms like Random Forest and XGBoost to improve detection accuracy.

3 Methodology

A structured review consisting of three main steps can provide a thorough evaluation of the methodological approach.

Data Gathering and Preparation: During this first stage, raw data is obtained and put through pre-processing procedures. This entails preparing the data for analysis, addressing missing values, and cleansing the data. To find pertinent features that add to the analysis, feature selection or extraction may also be used. To guarantee uniformity across the dataset, normalization or scaling of the data may also be used. For the ensuing analysis and model creation, this stage lays the groundwork.

Model Development and Testing: Following data pre-processing, machine learning or statistical models must be created and tested. In this stage, the right algorithms or approaches are chosen in accordance with the goals of the study and the features of the dataset. A subset of the prepared data is used to train the models, which are then tested on a different dataset to determine how well they perform. The efficacy of the model in delivering the desired goals is assessed using evaluation measures like accuracy, precision, recall, *F1*-score, and ROC curves.

Verification and Analysis: Model validation and interpretation are required in the last step. In order to evaluate the model's performance in real-world circumstances, its generalization potential must first be evaluated by applying it to fresh, previously unexplored data. The predictability of the model is essential because it enables stakeholders to comprehend the factors influencing the outcomes and make wise decisions in light of the findings. To confirm their applicability and relevance, model outputs are also compared to domain knowledge or expert knowledge.

The credit card dataset is first obtained from the data source as part of the technique. A thorough data pre-processing phase that includes data cleansing and validation methods then follows. In order to do this, redundant or superfluous data entries must be removed, missing values must be filled in, and categorical variables must be transformed into numerical formats or intelligible representations. The training dataset and the validation dataset are then separated from the whole dataset.

Using a *k*-fold cross-validation technique, the model's resilience and generalizability are improved. The dataset is randomly divided into *k* subsets, each of equal size, via this method. In order to guarantee that every data point is used for validation exactly once, each subset, or fold, serves as both a training and validation set iteratively. With less chance of overfitting and a more thorough evaluation of the model's prediction skills, this method helps examine the model's performance across diverse data segments.

The steps that make up an XGBoost model's training procedure are as follows:

Start with the input training data (X, y) , where X denotes the features and y is the target variable or labels.

Model hyperparameter initialization set the XGBoost model's initial hyperparameters. These parameters control the behavior of the model and include characteristics like the maximum depth of individual trees, learning rate, and others.

Splitting Data: Typically, the training set and the validation set should be separated into two subsets of the data. The validation set aids in performance monitoring and overfitting detection while the training set is utilized to train the model.

The main component of XGBoost is a series of boosting rounds ($n\_estimators$), each of which seeks to increase the model's propensity for prediction.

- a. **Gradient and Hessian Calculation:** Using the present model, compute the gradients and Hessians for each sample in the training set. While the Hessian offers a measurement of curvature, gradients show the direction of the loss function's steepest increase.
- b. **Fitting a Decision Tree:** Using the Hessian as weights, fit a decision tree to the negative gradients. It's common to refer to this decision tree as a weak learner.
- c. **Prediction Scores:** Using the modified model, determine prediction scores for each sample in the training and validation sets. These results represent the model's most recent forecasts.
- d. **Loss Function Calculation:** For both the training and validation sets, calculate the loss function, which evaluates the effectiveness of the model. Log loss (cross-entropy) for classification is a common loss function.
- e. **Deploy the Trained XGBoost Model:** After the training procedure is finished, the trained XGBoost model is prepared to be used in order to make predictions on fresh, unforeseen data.

Extreme Gradient Boosting, also known as XGBoost, is a scalable and computationally efficient machine learning algorithm that builds additive models in stages. The architecture of the proposed model is shown in Fig. 1. Through the introduction of subsequent base learners that learn from the mistakes of their predecessors, it gradually reduces total prediction errors. An ensemble of base learners, often shallow decision trees, contribute equally to a final combined model using this iterative methodology. Stochastic gradient boosting, which introduces randomization during training, is a feature of XGBoost that increases its robustness against noise and overfitting.

Additionally, XGBoost uses regularization approaches to reduce model complexity, avoid overfitting, and enhance generalization across various datasets. In general, XGBoost is a strong and adaptable technique that has been widely used in a variety of machine learning applications because to its remarkable performance and scalability.

4 Result Analysis and Discussions

In the proposed model, dataset that contains detailed information about insurance plans, the customer information related to them, and particulars about the incidents that gave rise to claims are included. Including features like policy numbers, policy binding dates, annual premiums, incident severity, incident locations, vehicle models, and more, the dataset has 1000 rows and 40 columns (Fig. 2).

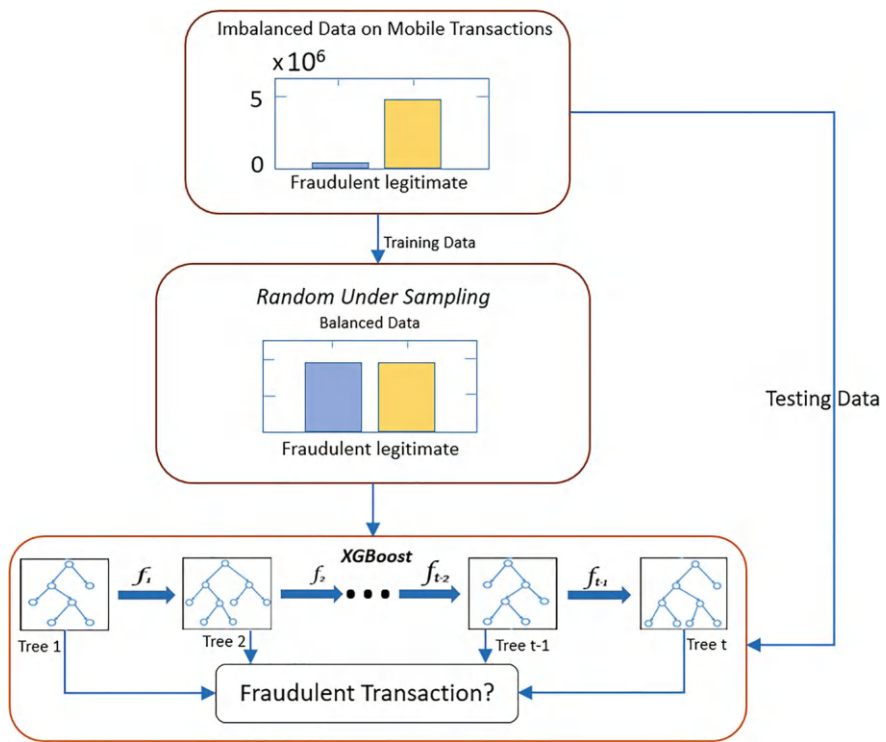


Fig. 1 Proposed model architecture

The focal point of our exploratory data analysis was the dependent variable, `Fraud_reported`. Among the dataset, there were 247 cases of fraud and 753 instances of non-fraudulent claims. This signifies that frauds accounted for 24.7% of the data, while non-fraudulent claims constituted 75.3% of the total observations (Fig. 3).

Regarding correlations among variables, we generated a heat map that focused on variables exhibiting a Pearson’s correlation coefficient of at least 0.3, which also included the dependent variable (DV). Notably, we observed a strong correlation of 0.92 between the “Month as customer” and “Age” variables. This correlation can be attributed to the fact that individuals tend to purchase auto insurance when they acquire a vehicle, and this duration naturally increases with age. Apart from this, the dataset displayed limited correlations among other variables. Multicollinearity issues were not prominent, except for the inter-correlations among various claims variables, which were primarily accounted for by the “Total Claims” variable. Nevertheless, we retained these distinct claims variables as they provided valuable granularity not captured solely by the total claims metric.

In terms of visualizing variables, we noted variations in the reported fraud values across different customer hobbies. Interestingly, it appeared that individuals engaged in chess and crossfit activities exhibited a higher inclination toward fraud (Fig. 4).

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 40 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   months_as_customer                       1000 non-null   int64
1   age                                       1000 non-null   int64
2   policy_number                           1000 non-null   int64
3   policy_bind_date                        1000 non-null   object
4   policy_state                             1000 non-null   object
5   policy_csl                              1000 non-null   object
6   policy_deductable                       1000 non-null   int64
7   policy_annual_premium                   1000 non-null   float64
8   umbrella_limit                          1000 non-null   int64
9   insured_zip                             1000 non-null   int64
10  insured_sex                             1000 non-null   object
11  insured_education_level                 1000 non-null   object
12  insured_occupation                     1000 non-null   object
13  insured_hobbies                         1000 non-null   object
14  insured_relationship                   1000 non-null   object
15  capital-gains                          1000 non-null   int64
16  capital-loss                           1000 non-null   int64
17  incident_date                           1000 non-null   object
18  incident_type                           1000 non-null   object
19  collision_type                          1000 non-null   object
20  incident_severity                       1000 non-null   object
21  authorities_contacted                   1000 non-null   object
22  incident_state                           1000 non-null   object
23  incident_city                           1000 non-null   object
24  incident_location                       1000 non-null   object
25  incident_hour_of_the_day                 1000 non-null   int64
26  number_of_vehicles_involved              1000 non-null   int64
27  property_damage                         1000 non-null   object
28  bodily_injuries                         1000 non-null   int64
29  witnesses                               1000 non-null   int64
30  police_report_available                 1000 non-null   object
31  total_claim_amount                      1000 non-null   int64
32  injury_claim                            1000 non-null   int64
33  property_claim                          1000 non-null   int64
34  vehicle_claim                           1000 non-null   int64
35  auto_make                               1000 non-null   object
36  auto_model                              1000 non-null   object
37  auto_year                               1000 non-null   int64
38  fraud_reported                          1000 non-null   object
39  _c39                                    0 non-null     float64
dtypes: float64(2), int64(17), object(21)

```

Fig. 2 Dataset with the features used to train the model

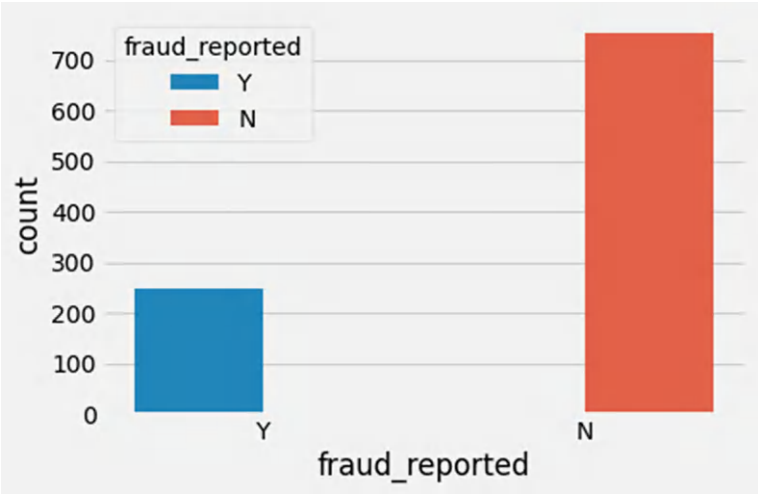


Fig. 3 Report of fraud prediction

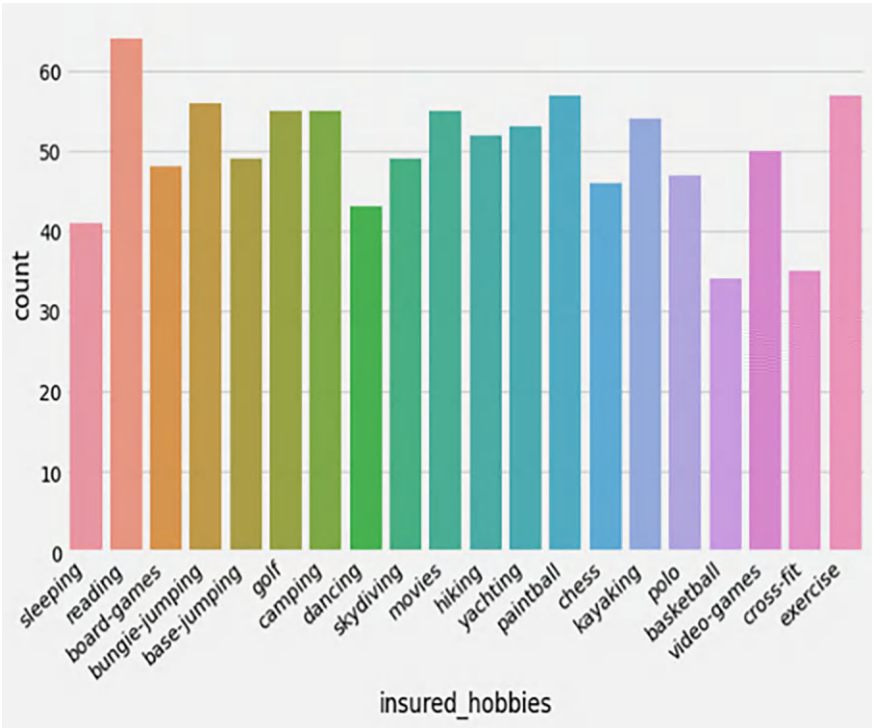


Fig. 4 Report on fraud values for various customers hobby

4.1 Performance Evaluation

Standard classification metrics to assess the performance of our fraud detection models are applied. The true positive rate, also known as Recall, is calculated as the ratio of correctly identified fraudulent transactions to the total number of fraudulent transactions:

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

where TP represents the count of true positive transactions, and FN represents the count of false negative transactions. The false positive rate, denoted as FPR, is determined by the proportion of incorrectly identified fraudulent transactions relative to all legitimate transactions:

$$\text{FPR} = \text{FP} / (\text{FP} + \text{TN})$$

Here, FP stands for the count of false positive transactions, and TN stands for the count of true negative transactions. Additionally, the false negative rate (FNR) signifies the percentage of transactions incorrectly categorized as legitimate among all fraudulent transactions:

$$\text{FNR} = \text{FN} / (\text{TP} + \text{FN}) = 1 - \text{Recall}$$

In practical scenarios, financial institutions face the challenge of mitigating fraud risk while adhering to regulations. However, estimating Recall is often elusive since FN (hidden fraud) remains unknown. Consequently, financial institutions typically rely on calculating Precision, which signifies the proportion of correctly identified fraudulent transactions among all transactions flagged as potentially fraudulent:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

where TP represents the count of true positive transactions, and FP represents the count of false positive transactions. Additionally, previous research has introduced the F1 measure, which is defined as the harmonic mean of Precision and Recall:

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

Furthermore, the area under the receiver operating characteristic curve (AUC) serves as a valuable metric for assessing fraud detection in mobile payment transactions, especially when dealing with imbalanced data. AUC quantifies the probability that a randomly selected fraudulent transaction will be ranked higher by a fraud detection model than a randomly selected legitimate transaction. The AUC is calculated as:

$$\text{AUC} = \int[0, 1] (\text{Recall}(T) \times d(\text{FPR}(T)) / dT) dT$$

Here, T represents the cut-off point in the model's decision threshold.

The performance of the proposed model using the XGBoost and the comparison with other model is shown in Fig. 5 and (Table 1).

In terms of insurance fraud detection, XGBoost shows to be a reliable algorithm that strikes a fair compromise between interpretability and performance.

Highly suspicious claim (<i>susp</i>)			
<i>tf_xgb.score_{susp}</i>	17***	25***	40***
<i>Random selection</i>	2	4	10
<i>k</i>	100	200	500
Proven fraudulent claim (<i>fraud</i>)			
<i>tf_xgb.score_{fraud}</i>	4***	6***	10***
<i>Random selection</i>	0	1	2
<i>k</i>	100	200	500

Fig. 5 Performance measure of XGBoost model

Table 1 Comparison with other models

XGBoost	Random Forest
Precision: 0.89	Precision: 0.82
Recall: 0.78	Recall: 0.75
F1-score: 0.81	F1-score: 0.78
AUC-ROC: 0.92	AUC-ROC: 0.88
SVM	LSTM
Precision: 0.76	Precision: 0.88
Recall: 0.68	Recall: 0.82
F1-score: 0.72	F1-score: 0.85
AUC-ROC: 0.85	AUC-ROC: 0.90

5 Conclusion

An efficient and potent method for foretelling insurance fraud is XGBoost. To create a powerful predictive model, this study used a dataset that included precise information about insurance policies, customer information, and accident-related data. The capacity of XGBoost to accurately identify fraudulent insurance claims was proved through exploratory data analysis, feature building, and model training. Precision, Recall, *F1*-score, and AUC were some of the evaluation metrics that demonstrated how well the model worked to identify fraud while reducing false positives. The study also emphasized the significance of addressing class disparity and the potential influence of specific factors, such as consumer hobbies, on fraud tendencies. Overall, XGBoost is an excellent option for insurance providers looking to strengthen their fraud detection capacities, which will ultimately result in cost savings and better service.

References

1. Zhou X, Cheng S, Zhu M, Guo C, Zhou S, Xu P, Xue Z, Zhang W (2018) A state of the art survey of data mining-based fraud detection and credit scoring. In: MATEC web of conferences, vol 189. EDP Sciences, p 03002
2. Fursov I, Kovtun E, Rivera-Castro R, Zaytsev A, Khasyanov R, Spindler M, Burnaev E (2022) Sequence embeddings help detect insurance fraud. *IEEE Access* 10:32060–32074
3. Sedlakova J, Daniore P, Horn Wintsch A, Wolf M, Stanikic M, Haag C, Sieber C et al (2022) Challenges and best practices for digital unstructured data enrichment in health research: a systematic narrative review medRxiv, July 2022
4. Long J, Yan Z, Shen Y, Liu W, Wei Q (2018) Detection of epilepsy using MFCC-based feature and XGBoost. In: IEEE international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI'18), Beijing, China, Oct 2018
5. Chen Z, Jiang F, Cheng Y, Gu X, Liu W, Peng J XGBoost classifier for DDos attack detection and analysis in SDN-based cloud (2018) In: IEEE international conference on big data and smart computing (BigComp'18), Shanghai, China, Jan 2018
6. Subudhi S, Panigrahi S (2020) Use of optimized Fuzzy C-Means clustering and supervised classifiers for automobile insurance fraud detection. *J King Saud Univ Comput Inf Sci* 32(5):568–575
7. Severino MK, Peng Y (2021) Machine learning algorithms for fraud prediction in property insurance: empirical evidence using real-world microdata. *Mach Learn Appl* 5:100074
8. Yao J, Zhang J, Wang L (2018) A financial statement fraud detection model based on hybrid data mining methods. In: 2018 International conference on artificial intelligence and big data (ICAIBD), IEEE, pp 57–61
9. Jurgovsky J, Granitzer M, Ziegler K, Calabretto S, Portier P-E, He-Guelton L, Caelen O (2018) Sequence classification for credit-card fraud detection. *Expert Syst Appl* 100:234–245
10. Dhieb N, Ghazzai H, Besbes H, Massoud Y (2020) A secure AI-driven architecture for automated insurance systems: fraud detection and risk measurement. *IEEE Access* 8:58546–58558
11. Liu S, Yang B, Wang L, Abraham A (2016) Advances in nature and biologically inspired computing. In: *Advances in intelligent systems and computing*, vol 419. pp. 97–104

Analyzing the Impact of Social Media Advertising Using Machine Learning Technique



B. Roopashri Tantri, Padma Bhaskar, and N. Sandhya

Abstract Nowadays, social media has become a powerful platform for generating revenue of the companies and this media has made the world smaller especially in the last decade. It has become a mandatory platform for the companies to reach their customers. Therefore, this study aims at increasing the revenue by means of advertising through social media. As per digital 2023 global overview report, the average consumer spends about 151 min per day on social media, it is a good place to reach a larger audience. Herein, it is intended to analyze the effectiveness of social media advertising on YouTube, Facebook and newspaper in generating sales. A dataset containing the past data pertaining to these variables has been considered and multiple linear regression model has been developed using R, the most powerful machine learning tools to analyze statistical data. It has been found that advertisements on YouTube and Facebook have a significant impact on sales, while newspaper does not have significant role in generating sales.

Keywords Advertisement · Multiple linear regression · R · Sales and social media

1 Introduction

Social media advertising has gained popularity in the present era. It uses platforms like Facebook, Instagram, X, etc., as means of digital marketing to deliver advertisements for targeting the audience. Social media advertisements are a quick and effective media for connecting with consumers thereby boosting the marketing campaigns. In this digital era, consumers look for information about the brands through social media and use the services provided by these social media to buy the products online. Hence, using social media strategies for marketing campaigns can ensure quick advertisements, which can reach the target audiences instantly.

B. R. Tantri (✉) · P. Bhaskar · N. Sandhya
Sindhi Institute of Management, Bengaluru, India
e-mail: roopatantri@gmail.com

The impact of social media advertising is significant. It increases brand awareness and helps in higher rate of conversion rates of leads. There are many techniques to analyze the impact of social media advertising. The common method used for such analysis is the survey method by administering the questionnaire and analyzing the attitude. But the present study has adopted different approach by using the machine learning technique. Machine learning has been widely used in social sciences to discover new concepts, measure the prevalence of those concepts, assess causal effects and make predictions.

Machine learning is a popular method to draw inferences based on the patterns exhibited by the data. The application of machine learning has spread across various disciplines in social sciences in the present scenario. This technique provides a scientific approach to social sciences by enabling accuracy in prediction and analysis.

2 Literature Review

In the current scenario, artificial intelligence techniques have been widely used by the marketers to get valuable customer insights from the data. They use many social media platforms to share their views about their products and services. Different market challenges can be addressed by using machine learning approaches to recover from business crises and in rectifying them. Machine learning techniques also help in reducing a large amount of human workload.

Many previous studies have highlighted the importance of machine learning applications in marketing. Because of the tremendous advantage of machine learning approaches in the field of marketing, many tech giants like Google, Netflix, Facebook, etc., and transport aggregators like Ola, Uber, etc., have adopted these techniques to create the platforms which can understand the needs of the people, thereby providing suggestions based on their interests. Many global companies apply machine learning techniques for marketing. Nearly 84% of marketing agencies implement AI and ML projects and 75% of them have improved consumer satisfaction by 10%.

Srihith et al. [1] used data science methodologies to shed light on the analysis of social media advertisements to achieve efficient target audience classification. Preeti et al. [2] studied the implications and impact of artificial intelligence in digital media, especially, the social media marketing. They made an elaborate study on the usage of different techniques used by people from different domains with respect to usage of artificial intelligence and machine learning. Manasvi et al. [3] studied the impact of marketing on social media platforms through machine learning approaches. YouTube, Instagram, WhatsApp and Facebook were used for the analysis. Also used were social media analytics and sentimental analysis to extract insights from the data. Pamuksuz et al. [4] developed a novel hybrid machine learning model that could be used for a wide range of management studies. The study found that social media networks were capable of fostering branding strategies by using big data resources. Kongar and Adebayo [5] discussed the implications of the findings for small and medium sized enterprise marketing managers at the industry level. Senthil et al. [6]

analyzed the data from social media to analyze the effects of using machine learning and employed knowledge analysis tool to indicate that it performs better than other approaches. Miklosik and Evans [7] examined the effect of applying the techniques of big data analytics and machine learning on digitized marketing strategies, along with some other challenges that could occur from data and information management. Vermeer et al. [8] proved that machine learning techniques achieve higher accuracy in detecting electronic word of mouth on social media.

With the review of the above articles, it can be ascertained that there is a need to conduct research on analyzing influence of social media advertising using machine learning technique. Using these applications to meet the customer needs and to build strong relationships is the winning strategy of the market departments of various sectors.

3 Development of the Model

Herein, the marketing dataset obtained from Kaggle (source: www.kaggle.com) is used, which consists of the sales from YouTube, Facebook and newspaper. The scatter plots of sales vs each of the variables YouTube, Facebook and newspaper show linear patterns and hence a multiple linear regression model can be developed to study the behavior of sales with respect to these three variables.

Variables Used

Independent variables: YouTube, Facebook, newspaper.

Dependent variable: sales.

R code: The R code for developing the multiple linear regression model for the marketing data by considering the above variables is:

```
#Multiple linear regression for marketing data.
library(readr).
Data <- read_csv("Marketing_Data.csv").
sum(is.na(Data)).
boxplot(Data).
Q1 <- quantile(Data$newspaper, 0.25).
Q3 <- quantile(Data$newspaper, 0.75).
IR <- Q3-Q1.
newdata <- subset(Data, Data$newspaper > Q1-1.5*IR & Data$newspaper < Q3
+ 1.5*IR).
model <- lm(sales ~ ., data = newdata).
summary(model).
qqnorm(model$residuals).
On execution, the above R code produces the following output:
[1] 0
```

Call	lm(formula = sales ~ ., data = newdata)					
Residuals	Min	1Q	Median	3Q	Max	
	-10.6823	-0.8619	0.3192	1.3359	3.3827	
Coefficients		Estimate	Std. Error	t value	Pr(> t)	
	(Intercept)	3.482865	0.402606	8.651	4.36e-15 ***	
	YouTube	0.044717	0.001528	29.274	< 2e-16 ***	
	Facebook	0.189473	0.008947	21.177	< 2e-16 ***	
	Newspaper	0.007482	0.006820	1.097	0.274	
Significance codes	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ''	1
Residual standard error	1.991 on 165 degrees of freedom					
Multiple R-squared	0.9009	Adjusted R-squared: 0.8991				
F-statistic	499.8 on 3 and 165 DF	p-value: < 2.2e-16				

4 Interpretation of the Output Results

The dataset has no missing values. Few outliers are present in ‘newspaper’ (Fig. 1), which are removed using the method of inter quartile range.

The residuals of the fitted multiple linear regression model lie in the range -10.6823 and 3.3827.

Median value of the residuals is 0.3192, which is close to zero. This indicates that residuals have the tendency to move toward the value ‘zero’.

The model output infers the following:

- The standard errors of the model coefficients are very small (close to zero).
- The intercept coefficient is significant.
- The coefficients corresponding to YouTube and Facebook are significant.
- The coefficient corresponding to newspaper is insignificant.

Fig. 1 Parallel box plots of the data

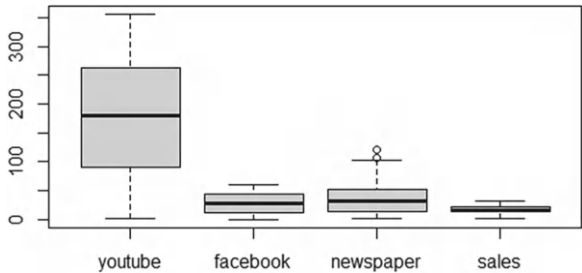
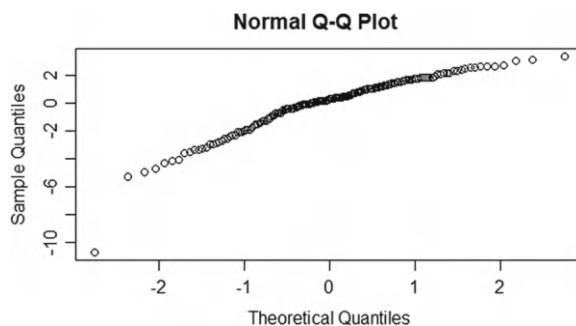


Fig. 2 Normal Q-Q plot of model residuals



- Adjusted R-squared value is 0.8991, which indicates that 89.91% of variations in sales are explained by YouTube, Facebook and newspaper.
- p-value of the model is 2.2×10^{-16} , which is < 0.05 . Hence the multiple linear regression model developed above is a model of best fit.
- The normal quantile–quantile plot of residuals (Fig. 2) shows a linear pattern, indicating that the residuals are normally distributed, as required by the model.

Overall inference: The above interpretations infer that the fitted multiple linear regression model by considering sales as dependent variable and YouTube, Facebook and newspaper as independent variables, is a good predictor model. However, this fitted model indicates that the components ‘YouTube’ and ‘Facebook’ have a significant impact on sales, while the component ‘newspaper’ does not have a significant impact on sales.

Observation: Since the newspaper does not have much impact on the sales, the regression model can be enhanced further by removing the newspaper component from it and validating the model again.

5 Conclusion

The paper explains the use of machine learning techniques in predicting the sales through social media advertisements. The sample dataset finds that implementation of multiple linear regression model using machine learning technique is a predictor for the marketing performance than other tools. This tool can be extended for further research by considering the various sectors like healthcare, education, entertainment, etc. Apart from machine learning techniques, other advanced tools in AI can be used.

References

1. Srihith IVD, Srinivas TAS, Donald AD, Thippanna G (2023) Predicting success: the impact of machine learning on social media ad classification. *Recent Trends Cloud Comput Web Eng* 5(3):17–23
2. Singh P, Verma A, Vij S, Thakur J (2023) Impact of artificial intelligence in digital media: with special focus on social media marketing. In: *E3S Web of conferences, ICONNECT*
3. Varshney M, Singh S, Kumar V (2022) Social media analytics using machine learning. *Int J Eng Res Technol* 11(11)
4. Pamuksuz U, Yun JT, Humphreys A (2021) A brand-new look at you: predicting brand personality in social media networks with machine learning 56(1). <https://doi.org/10.1016/j.intmar.2021.05.001>
5. Kongar E, Adebayo O (2021) Impact of social media marketing on business performance: a hybrid performance measurement approach using data analytics and machine learning. *IEEE Eng Manag Rev* 49(1):133–147. <https://doi.org/10.1109/EMR.2021.3055036>
6. Senthil Arasu B, Jonath Backia Seelan B, Thamaraiselvan N (2020) A machine learning-based approach to enhancing social media marketing. *Comput Electr Eng*
7. Miklosik A, Evans N (2020) Impact of big data and machine learning on digital transformation in marketing: a literature review. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2020.2998754>
8. Vermeer SAM, Araujo T, Bernritter SF, Van Noort G (2019) Seeing the wood for the trees: How machine learning can help firms in identifying relevant electronic word-of-mouth in social media. *Int J Res Mark*

The Ultimate Air Quality Predictor Using Virus Colony Search Optimisation-Based UATNet Classification



Zabiha Khan, R. J. Anandhi, and B. Guna Priya

Abstract Air quality prediction (AQP) is critical to reducing the harmful effects of air pollution on the environment and public health. This study offers a thorough method for predicting air quality (AQ) using deep learning (DL). The Central Pollution Control Board (CPCB), which is part of the Indian government's Ministry of Environment, Forests, and Climate Change, provided the air pollution databases used in this study. In order to improve the consistency and quality of the data, this paper used Adaptive Savitzky–Golay Filters (ASGF) for preprocessing. These filters were essential in reducing noise and smoothing the data, which guaranteed that the DL models were fed high-quality input. It made use of MobileNet, a cutting-edge DL architecture well-known for its effectiveness and efficiency in gleaning relevant features from images and multi-modal data, for feature extraction. This made it easier to transform complicated data on AQ into a format that the suggested model could easily understand. Our new U-Shape Attention-Based Transformer Net (UATNet), which was created to handle the complex spatiotemporal dependencies found in AQ data, was used to carry out the classification process. This paper used virus colony search optimisation (VCSO) to fine-tune the hyper parameters of the UATNet in order to maximise its performance. With the help of this special optimisation technique, we were able to maximise the model's predictive power. The suggested VCSO-UATNet model outperforms the current model by demonstrating that it can reach an accuracy rate of 99%, which is an impressive improvement.

Keywords Adaptive Savitzky–Golay Filters · Air quality prediction · Deep learning · Mobile network · Virus colony search optimisation

Z. Khan (✉)

Nitte Meenakshi Institute of Technology, Bengaluru, India

e-mail: zabiha.khan@nmit.ac.in

R. J. Anandhi · B. Guna Priya

New Horizon College of Engineering, Bengaluru, India

1 Introduction

Air pollution is a serious worldwide environmental problem that has an impact on public health, Climate Change, and our standard of living in general [1]. Pollutants from a variety of sources, including vehicle exhaust, industrial emissions, agricultural practices, and natural phenomena, are the main cause of it. Particulate matter (PM), nitrogen dioxide (NO₂), sulphur dioxide (SO₂), carbon monoxide (CO), and volatile organic compounds (VOCs) are among the pollutants that are included in this category. These pollutants are known to present serious risks to human health and the environment [2]. In actuality, a number of health concerns, such as cardiovascular illnesses, respiratory disorders, and even early mortality, are linked to poor AQ [3]. Thus, reducing the negative effects of air pollution requires an understanding of, and the ability to monitor and forecast, AQ. One essential component of daily living that directly affects our health and wellbeing is the quality of the air we breathe [4].

The need to develop techniques for monitoring and forecasting AQ is urgent because of the elevated levels of air pollution caused by the recent upsurge in urbanisation and industrialisation [5]. The problem of AQ is complex and affected by many variables, including topography, weather, seasonal changes, and emissions from industries and automobiles [6]. To predict AQ accurately, one must be able to identify the complex interactions and nonlinear relationships between these variables [7]. Predicting AQ is extremely important for environmental science and public health. It is intrinsically linked to temporal dependencies, in which historical data as well as current conditions have an impact on the concentration of pollutants [8].

Feature engineering, which comprises the manual selection and transformation of pertinent features from raw data, is frequently used in conventional machine learning (ML) models [9]. This procedure may require a lot of work and be biased by people. On the other hand, DL models do not require extensive feature engineering because they automatically extract relevant features from raw data [10]. This flexibility is especially helpful when predicting AQ, as the importance of some features can change over time due to variables like seasonal variations and weather [11]. Nonlinear characteristics are frequently observed in the interaction between the different factors affecting AQ [12]. DL models are naturally suited to model complex nonlinearities because of their deep, layered architectures and activation functions. This allows them to accomplish a task that is difficult for the linear models often used in traditional ML: Capturing subtle interactions between input features and pollutant concentrations [13].

1.1 Motivation

The urgent need to address the growing issues related to air pollution, which seriously endangers both the environment and public health, is what drives this work. The

complex interactions between different pollutants, fast urbanisation, and industrialisation all contribute to the global concern over AQ. The need for sophisticated and precise prediction models that enable decision-makers in government and the public to take proactive steps to limit their exposure to dangerous pollutants is what motivates this research. This paper's goal is to greatly increase the accuracy and timeliness of AQP by utilising DL. By using these models, communities that are exposed to contaminated air can benefit from early warning systems, informed policy choices, and an overall improvement in their quality of life. In addition, the possibility of creating a new benchmark for AQ forecasting drives our efforts, ultimately leading to the development of sustainable environments and healthier societies.

1.2 Main Contributions

- Preprocessing with ASGF shows how important it is to enhance data quality and lower noise, producing high-quality input data for proposed model.
- By utilising MobileNet's feature extraction capabilities, this study highlights how effectively it can extract significant features from complicated AQ data, improving the models' interpretability.
- Creation of the innovative UATNet for classification, which uses virus colony search (VCS) optimisation to optimise the hyper parameters of UATNet model with the highest possible predictive accuracy AQ data.
- Results are evaluated using CPCB dataset with 4 performance metrics such as accuracy (ACC), precision (PRE), Recall (RE), F1-score (F1).

1.3 Organisation of the Paper

The rest of the essay is organised as follows. In Sect. 2, emphasise some relevant connected content. Section 3 provides the proposed technique. Section 4 explains the experiments and their findings. An overview of the findings and some viewpoints are included in Sect. 5's conclusion.

2 Related Works

Jin et al. [14] developed a time series prediction network known as the Bayesian Graph Gated Recurrent Unit (BGraphGRU) with the goal of recognising shifting patterns in time series data and taking spatial propagation effects into consideration. Both spatial and temporal modules are present in this network. To extract spatial information from the data, the spatial module makes use of the Graph Sampling and Aggregation Network (GraphSAGE). To model the temporal properties of the data,

the temporal module makes use of a Gated Recurrent Unit (GRU) and the idea of a Bayesian graph. In the study, hyper parameter tuning issues were addressed through the application of Bayesian optimisation. Their method proved to be highly accurate in predicting Beijing, China's PM_{2.5} concentrations, making it a useful tool for AQ forecasting.

To predict the AQ at sparse stations, Hu et al. [15] used a hybrid approach. Their approach is broken down into three primary parts. To handle abrupt changes, they first extracted multi-scale temporal features by taking into account different time lags of meteorological factors. They then selected key features from multiple data sources and removed unnecessary ones. In the second section, a partition of the spatial hierarchy was utilised to extract spatial features from stations that were related. The third element was a prediction hybrid model composed of sets of spatiotemporal features. Their approach proved to be more successful for sparse stations and to outperform previous methods for sudden-change prediction, according to their results.

Improved Variable-Length Long Short-Term Memory with Multi-Channel Input and Multi-Route Output (IVLSTM-MCMR) was introduced by Fang et al. [16] as a model for AQP; it performed exceptionally well for multi-step forecasts over longer time horizons. This model integrates the MCMR and IVLSTM modules. The Variable-Length Long Short-Term Memory (VLSTM) structure is improved by the IVLSTM module in order to speed up convergence and minimise the number of parameters. Furthermore, they presented a novel stable training method that makes use of historical data. In the module, they combined an enhanced linear similarity dynamic time warping technique with a multi-channel data input model to choose pertinent data inputs for the IVLSTM. Additionally, a multirooted output model was created to provide distinct route-based outputs for different target stations with different characteristics. The experimental results demonstrated how well their model predicted the future using data from Beijing, China.

A meta-learning-based spatial-temporal adaptation method for cold-start air pollution prediction was presented by Wu et al. [17] in a different study. The framework that is agnostic to models could facilitate the adaptation of backbone predictors to diverse temporal and spatial locations. It provided cold-start capabilities and reduced the need for data accumulation by identifying the target city as one of its components and learning a factorisation of source data distribution. Additionally, they showed how to create a bi-directional meta-learner using task embeddings and features from prior knowledge.

Their approach produced a 5.2% lower 24-h prediction mean absolute error compared to pretraining and fine-tuning solutions, as evidenced by extensive experiments demonstrating its effectiveness for cold-start training in a new city with limited data.

The Balanced Spider Monkey Optimisation (BSMO) technique was finally presented by Aarthi et al. [18] for feature selection in AQP. They used a convolutional neural network (CNN) to analyse AQ data from four Indian cities by extracting deep features from the input dataset. Using a balancing factor, the BSMO method efficiently found pertinent features. Then, in order to predict time series, these features were fed into a Bi-directional Long Short-Term Memory (Bi-LSTM) model. Mean

Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) were all lower in the BSMO-Bi-LSTM model than in the other models, suggesting better feature selection performance than the existing techniques. This strategy may be useful to organisations that work to protect the environment and public health at the local or state level.

Shu et al.'s [19] discrete wavelet and convolution-based autoencoder (DW-CAE) model provides a holistic understanding of air pollution trends by balancing overall accuracy and local univariate prediction accuracy. DL techniques and signal processing are integrated in their model. It makes use of a discrete wavelet transform to extract high- and low-frequency features, an autoencoder based on Long Short-Term Memory (LSTM) for prediction, and a feature extraction module to record variable relationships. Results of evaluation using datasets from Beijing and Yining showed that the DW-CAE model outperformed eight baseline models for both single and multiple pollution factors. The combined R^2 values for the six air pollutants were more than 93%. This model provides significant assistance for air pollution forecasting, prevention, and control.

2.1 Research Gaps

There are still certain research gaps even though the previously mentioned research works provide insightful analysis and innovative approaches to AQP. Initially, in order to evaluate the generalisability of these models, a more thorough validation process involving various geographic locations is required. Furthermore, the majority of studies concentrate mostly on particular pollutants, such as PM_{2.5}, and would benefit from taking into account a wider range of AQ indicators. Moreover, the integration of sensor technologies and real-time data, coupled with strong validation against ground-truth data, would augment the veracity and utility of these models. Lastly, for improved comprehension and reliability in AQ forecasting applications, these models must incorporate interpretability and explainability features. Filling in these gaps would help the field progress and enable more reliable and useful models for AQP. In order to close these gaps, this work proposes new models specifically designed for sparse station prediction, uses cutting-edge feature extraction techniques, and incorporates cutting-edge hyper parameter optimisation strategies with classification model.

3 Proposed Methodology

The conceptual framework and the steps involved in the AQP process are shown in the block diagram in Fig. 1. This diagram shows the important parts of the system and how they interact with one another.

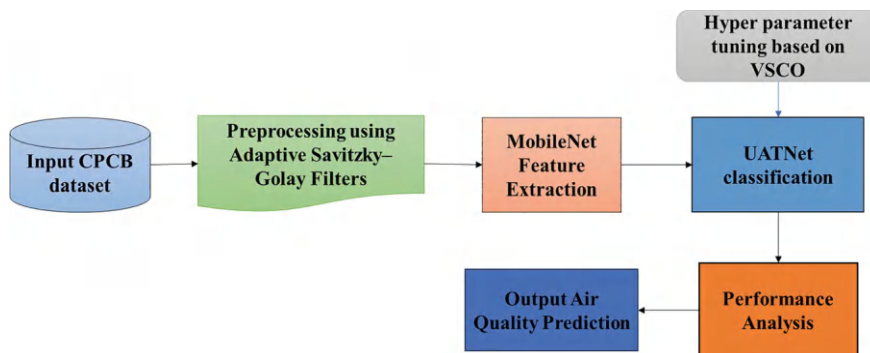


Fig. 1 Block diagram

3.1 Dataset Description

The entire experiment is carried out using a dataset that was retrieved from the Central Pollution Control Board (CPCB) official government website. The CPCB is a division of the Indian government’s Ministry of Environment, Forest, and Climate Change [20]. This source served as the basis for the selection of the experimental dataset:

- Dataset of Delhi.
- Real-time AQI values.
- 31 Records from December 1 to 31.
- 37 places in Delhi city.

The Air Quality Index (AQI) value is a composite metric that includes lead (Pb), ammonia (NH₃), carbon monoxide (CO), nitrogen dioxide (NO₂), sulphur dioxide (SO₂), and particulate matter (PM10).

As shown in Table 1 [21], the six categories of the Air Quality Index (AQI) range are separated into six values (1 through 6). An AQI score of one indicates “Good”, two indicates “Satisfactory”, three indicates “Moderately Polluted”, four indicates “Poor”, five indicates “Very Poor”, and six indicates “Severe”, which is a dangerous state. The December month’s dataset was used for this study due to the increased fog and smoke that resulted in elevated pollution levels in Delhi during that month. One of the most polluted areas of India is Delhi, where the dataset is based on air pollution. The goal of this study is to use the measured data to forecast future AQI values. For this, the gathered data is used in support vector machine and neural network algorithms.

Table 1 Standard AQI ranges for India

Values	AQI (range)	PM10	O ₃	CO	NH ₃	Pb	NO ₂	SO ₃	PM2.5
1	Good (0–5)	0–50	0–50	0–1.0	0–200	0–0.5	0–40	0–40	0–30
2	Satisfactory (51–100)	51–100	51–100	1.1–2.0	201–400	0.5–1.0	41–80	41–80	31–60
3	Moderate (101–200)	101–250	101–168	2.1–10	401–800	1.1–2.0	81–180	81–380	61–90
4	Poor (201–300)	251–350	169–208	10–1.7	801–1200	2.1–3.0	181–280	381–800	91–120
5	Very poor (301–400)	351–430	209–400	17–34	1200–1800	3.1–3.5	281–400	801–1600	121–250
6	Severe (401–500)	430+	748+	34+	1800+	3.5+	400+	1600+	250+

3.1.1 Descriptive Study for the Dataset

Many psychologists who have collected data then go on to summarise it using different techniques. Python was the programming language of choice for this research project’s descriptive statistics. Pandas and NumPy are two of the library functions used for data manipulation and statistics. There is more information about the Pandas descriptive statistics in [22]. The dataset is imported, and graphical representations of the historical trends are created, as seen in Fig. 2. Furthermore, in the context of correlation analysis, the daily AQI values are compared, and correlations are computed, as illustrated in Fig. 3.

3.2 Preprocessing Using ASGF

The paper presents ASGF in this section as an improvement over traditional SGF for input data preprocessing. Since the goal is to identify the corrupted peaks only, the classical SGF is frequently used to isolate noisy peaks within a given peak distribution pattern. This method first permits the window length (M) and polynomial degree (k) to be chosen arbitrarily. Then, in the following series, it shows the locations of the local minimum and maximum points that arise from the first smoothing procedure.

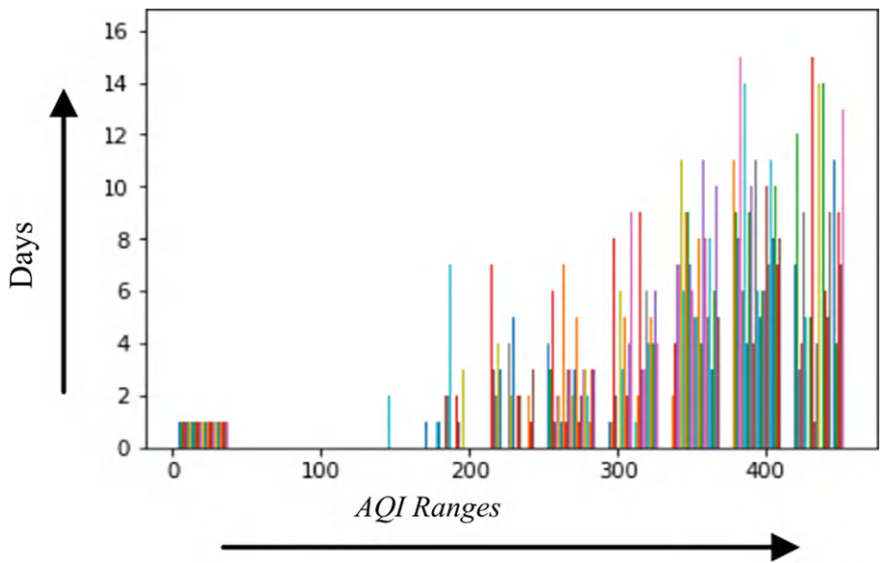
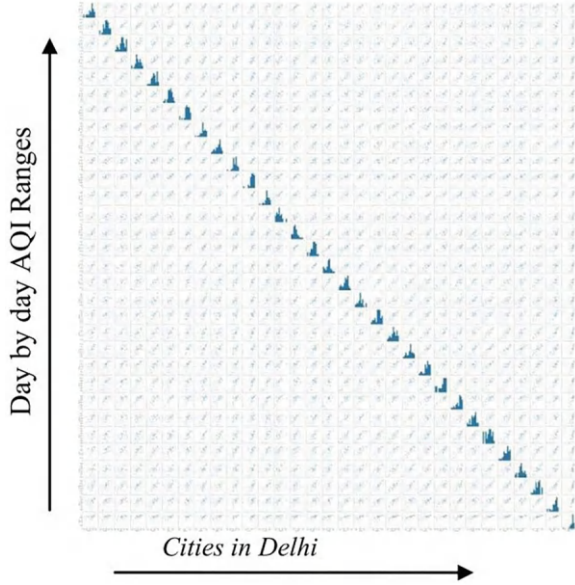


Fig. 2 Historical graph of Delhi AQI dataset

Fig. 3 Correlation graph of Delhi AQI dataset



$$C = \begin{pmatrix} x_1 & x_2 & \dots & x_u \\ y_1 & y_2 & \dots & y_u \end{pmatrix}, \quad (1)$$

The distance vector, denoted by “ d ” in this paper, is defined as the number of samples that separate neighbouring local minima and maxima.

$$d = (\delta_1, \delta_2, \dots, \delta_{u-1}), \quad (2)$$

Let $\delta = [(\delta_1, \delta_2, \dots, \delta_{u-1})]$ be the number of peaks in a particular peak distribution segment that have the same amplitude. Thus, one can use the following equation to iteratively determine the variance between adjacent peak points among the “ d ” local extrema.

$$\sigma_{(d)} = \frac{1}{(u-1) \sum_{i=1}^{u-1} \delta_i^2 - \delta^2} \quad (3)$$

As mentioned earlier, the variance that was previously determined is greater than 1. Here, we indicate the beginning of the next stage. The window length must, however, coincide with the peak spread distribution in order to guarantee alignment with it. The polynomial degree selection is then modified based on the frame size and peak distribution. As a result, for every segment consisting of peaks with comparable peak heights or amplitudes, a fuzzy relation, as outlined in the following equation, is utilised to determine an appropriate window length (M) and polynomial degree (k).

$$F(d_{\max} \gg d_s) = \frac{1}{1 + e^{-(d_{\max} - d_s)}} \in [0, 1], \quad (4)$$

$d_s = \max(d)$ is the symbol used to represent the distribution peak with the largest observed amplitude. The average length of segments is represented by d_s within the current S divisions of the peak distribution. In the event that $d_{\max}$ equals d_s , the largest amplitude peak is present in the relevant segment. Thus, the paper uses pre-established fuzzy rules to assign specific values of k and M to each S segment after obtaining the vector d and the coordinates of the local minimum and maximum peak points.

3.3 MobileNet

Google created MobileNet, a thin deep neural network architecture designed for embedded systems. Depth-wise separable convolution is the core innovation of this work. A single feature is obtained from a single convolution kernel in conventional convolution operations. The need to apply more filters in order to effectively capture a wider range of features is illustrated in Fig. 4. As such, using traditional convolution always results in a significant increase in the number of parameters and algorithmic computational burden.

Depth-wise separable convolution allows for the separation of standard convolution into two independent parts: Depth-wise convolution and pointwise convolution. Whereas the pointwise convolution handles channel transformation, the depth-wise convolution concentrates on filtering and works on each individual input channel.

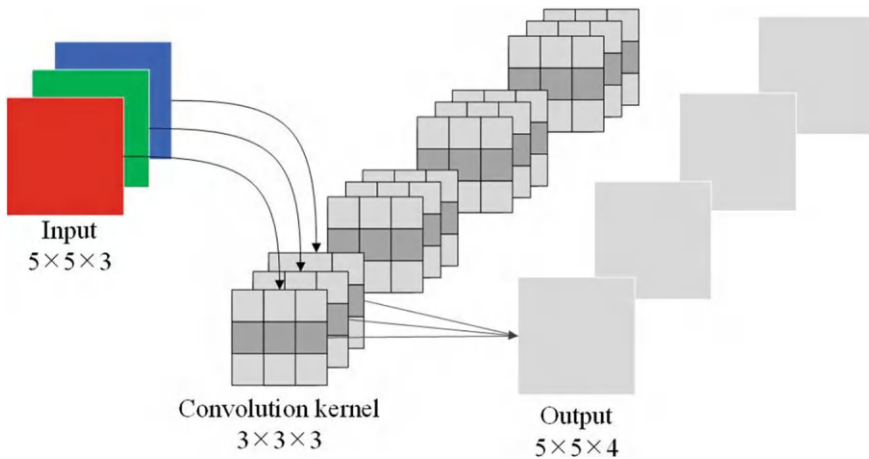


Fig. 4 Schematic diagram of the ordinary convolution process

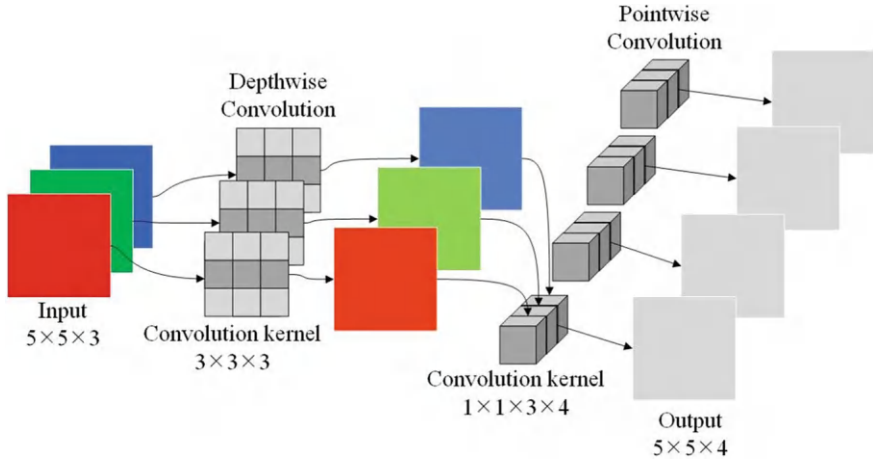


Fig. 5 Diagrammatic representation of the depth-wise separable convolution method

Figure 5 illustrates the depth-wise separable convolution workflow. MobileNet integrates more convolutional layers by splitting the convolution process [24]. As a result, this makes it possible to extract more minute information and nuances about AQ.

The following dimensions should be present in the input image: W stands for width, H for height, and C for the total number of channels. The kernel size in standard convolution is $K_W * K_H$. The size of the feature map that results is $W_O * H_O * C_O$.

$$K_W * K_H * C * C_O \quad (5)$$

where K_W and K_H stand for the convolution kernel's width and height, respectively.

The count of calculations is:

$$K_W * K_H * C * W_O * H_O * C_O \quad (6)$$

where W_O , H_O , and C_O stand for the output feature map's width, height, and number of channels, respectively.

Additionally, when using depth-wise separable convolution, the number of parameters is;

$$K_W * K_H * C + 1 * 1 * C * C_O \quad (7)$$

The number of calculations is:

$$K_W * K_H * C * W_O * H_O + 1 * 1 * C * W_O * H_O * C_O \quad (8)$$

As a result, the number of parameters to ratio is:

$$\frac{K_W * K_H * C + 1 * 1 * C * C_O}{K_W * K_H * C * C_O} = \frac{1}{C_O} + \frac{1}{K_W + K_H} \quad (9)$$

The proportion of the total computations is:

$$\frac{K_W * K_H * C * W_O * H_O + 1 * 1 * C * W_O * H_O * C_O}{K_W * K_H * C * W_O * H_O * C_O} = \frac{1}{C_O} + \frac{1}{K_W + K_H} \quad (10)$$

For the purpose of clarity and comparison, we will consider the typical convolution kernel size of 3×3 , assuming that $C_O = C$. As a result, depth-wise separable convolution has much fewer parameters and requires a computational workload that is about 1/9th of that of conventional convolution.

3.4 VCSO-Based UATNet Classification

3.4.1 UATNet Model

Many notable developments in DL methods for AQP (AQP) are based on CNN architectures. However, because their theoretical receptive field is much larger than their actual receptive field, CNNs are unable to represent large amounts of data efficiently. To get around this restriction, a brand-new end-to-end DL network known as UATNet has been presented. UATNet learns the mapping between image patch sequences and their corresponding semantic labels in an effort to increase the accuracy of AQP. In Fig. 6, we display the particular UATNet structure.

UATNet combines two distinct transformer structures: A channel transformer and a hierarchy transformer determined with shifted windows (Swin Transformer). These transformers carry out attention operations along both the patch and channel axes and are hidden within the encoder and encoder-decoder connections of the U-shaped architecture. This clever integration of spatial and multi-channel AQ data is impressive. Transformers do not need a lot of layer stacking to achieve a global receptive field, unlike CNNs. This section goes into further detail to give readers a comprehensive understanding of the contributions made by transformer structures to the UATNet model as well as their intricate details.

A. Encoder Architecture

The UATNet encoder converts low-dimensional characteristics into higher-dimensional, abstraction-based feature vectors and extracts high-level semantic features from them. Figure 6 shows the architecture specific to the UATNet encoder. E1–E5 we have four stages (named stage1–stage4) with increasing network depth in our encoder architecture. Each stage consists of a pair of successive Swin Transformer blocks followed by patch merging. These Swin Transformer blocks consist

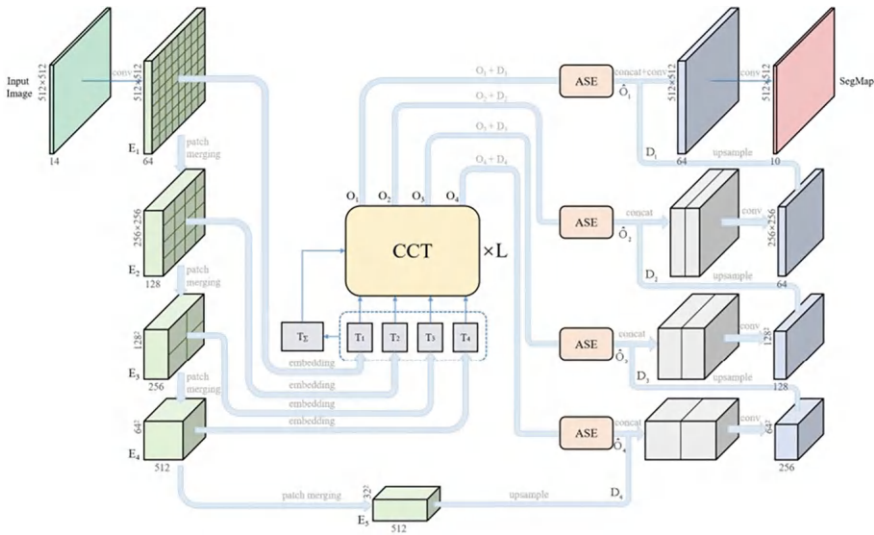


Fig. 6 Architecture of UATNet. ASE stands for attention-based squeeze and excitation, and CCT stands for channel-wise cross-fusion transformer

of several transformer blocks whose patch tokens have undergone modified self-attention computations. The input image is $14 \times H \times W$ in size. The first step is to use a 3×3 convolution to change the input image's channel dimension to C dimensions (in this case, C is set to 64). After processing, the input is fed into stage 1. In the first stage, the images are divided into a collection of 2×2 non-overlapping image patches. Each image patch has a characteristic dimension of $14 \times 2 \times 2 = 56$, and there are $H/2 \times W/2$ image patches total. The grid containing $H/2 \times W/2$ patches is pushed into a sequence to create one-dimensional image patch tokens. Then, each vectorised patch is mapped to a potential $2C$ -dimensional implementing space using a trainable linear projection function. The next step is to sequentially pass the tokens via a pair of transformer blocks.

The procedures in stages 2–4 follow an arrangement comparable to that of stage 1. In these stages, patch merging is used instead of patch partitioning to combine input based on 2×2 adjacent patches; the linear extension function mappings have dimensions of $4C$, $8C$, and $8C$, respectively. It's interesting to note that the patch partition for the first stage is the same as if neighbouring patches were combined onto just one pixel (1×1 image patch). Consequently, this procedure is referred to as patch merging in Fig. 6. The channel transformer and the hierarchical Swin Transformer are two unique transformer structures that are incorporated into the encoder and the encoder-decoder connection within the U-shaped architecture [25]. These transformers combine multiple channels of communication and spatial AQ information successfully by performing attention activity along the patch and channel axes. Transformers do not need a lot of layer stacking to achieve a global

receptive field, unlike CNN. This section provides a detailed description of these two transformer structures.

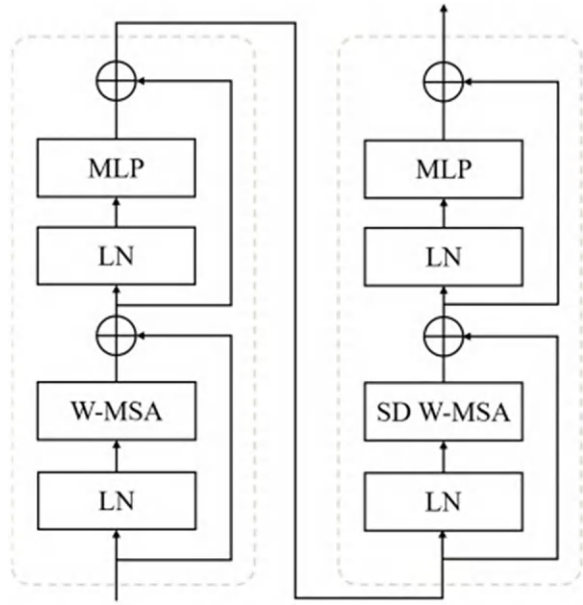
After passing through a Multi-head Self-Attention (MSA) module, tokens are processed by UATNet via a Multi-Layer Perceptron (MLP). The purpose of Layer Normalisation (LN), which is applied prior to each MSA and MLP operation, is to generate residual connections. Sliding window mechanisms in all MSA modules create non-overlapping windows based on the input feature maps. After that, we calculate self-attention in every one of these unique 16 by 16 patch windows. To capture the spatial information for each position, the network learns a specific position embedding based on the positional bias of patches within each window. The image patch's tokens to be used are improved by this position embedding. $[-15, 15]$ is the relative position of each axis. Even though the transformer's self-attention mechanism is inherently disordered, the input can still encode spatial location information related to the initial positions. A shifted window partition, similar to the Swin Transformer, is used in Fig. 7 to address the problem of information sharing between multiple windows. There are two transformer structures that are used in that order: Window-based Multi-head Self-Attention (W-MSA) and Window-based Multi-head Self-Attention with Spatial Displacement (SD W-MSA). Unlike W-MSA, which employs conventional window partitioning starting in the upper-left corner, SD W-MSA uses a windowing setup that is moved from the above layer. The displacement between the windows and the regularly partitioned windows is (8, 8) pixels. W-MSA's computational complexity is directly correlated with image size since individual windows have fewer patches than the full image. The receptive field is significantly increased by the shifted window partitioning method, which connects neighbouring non-coincident to windows from the topmost layer.

Spatial displacement, as defined by reference [26], is the process of reshaping the dimensions of space based on the width of a window and the quantity of tokens before sequentially placing them. In this work, spatial dimension transformation is used to encapsulate inputs from various windows by introducing spatial displacement into the SD W-MSA. This makes long- distance cross-window connections possible. Spatial alignment, which was also described in reference [26], is used after the self-attention operation to realign the features with the image's spatial layout by moving tokens back to their original positions. To maximise computational efficiency, the method combines window-to-image the conversion via space line and a window dividing with a displacement.

B. Encoder and Decoder Features Fusion

The encoder owns the feature maps E1–E5, while the decoder owns the feature maps D1–D4. The method used is described in this section. The output feature maps from the transformer layers are tokenised in the paper into four distinct scales, which correspond to encoder values E1–E4. Flattened 2D patches with patch sizes of $\{P, \frac{P}{2}, \frac{P}{4}, \frac{P}{8}\}$, are created from these features. As a result, the patches can be mapped across all four scales to the same regions in the encoder features. These patches produce four layers of tokens, known as T_k ($k = 1, 2, 3, 4$). The form of each T_k is $T_k(k = 1, 2, 3, 4), T_k \in R^{\frac{HW}{k^2} + C_k}$. $T_{\Sigma} = \text{Concat}(T_1, T_2, T_3, T_4)$ is the result

Fig. 7 Two successive swin transformer blocks in encoder



of concatenating these tokens from the four layers, T_k , and it functions as the key and value input. By feeding T_k ($k = 1, 2, 3, 4$) and T_Σ into the Channel-wise Cross-Fusion Transformer (CCT), the encoder features are effectively fused at four different scales, utilising the transformer's capacity for long-distance dependent modelling. The channel axis is used for attention operations rather than the patch axis in order to enhance gradient propagation and attention weight assignment. In the implementation, a 4-layer CCT is built and four attention heads are used. O_1, O_2, O_3, O_4 are the outputs of the fourth layer of the CCT, which are reconstructed using an upsampling process and a convolution layer. Following that, these reconstructed outputs are concatenated with the decoder features D_1, D_2, D_3, D_4 , in that sequence. Each k -th channel transformer output, $O_k \in R^{C \times H \times W}$, $k = (1, 2, 3, 4)$, and the k -th decoder feature map, $D_k \in R^{C \times H \times W}$, $k = (1, 2, 3, 4)$, serve as inputs for the Attention-based Squeeze and Excitation (ASE). “ r ” denotes the dimension reduction coefficient, and Fig. 8 shows the ASE architecture.

Within the Attentional Spatial Encoder (ASE), a pivotal operation involves feature compression of O_k and D_k along the spatial dimension to achieve an expansive global receptive field. This spatial compression is executed via a Global Average Pooling (GAP) layer:

$$G(O_k^c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W O_k^c(i, j), G(O_k^c) \in R^c \quad (11)$$

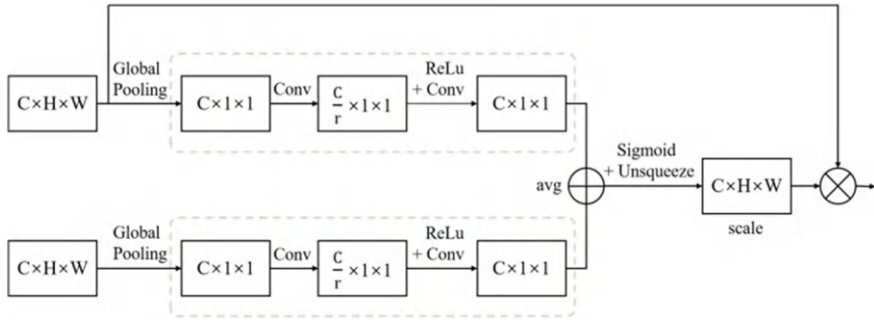


Fig. 8 ASE structure

$$G(D_k^c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W D(i, j), G(D_k^c) \in R^c \quad (12)$$

Here, an excitation operation is used to enhance the model's capacity for generalisation and facilitate the learning of nonlinear correlations between various channels, where “ c ” denotes the c -th channel and “ k ” denotes the k -th layer. The dimensions of feature channels are efficiently shrunk and then expanded back to their original sizes using the O'_k and D'_k processes. The activation function known as the Rectified Linear Unit (ReLU) is employed in this process. Interestingly, one specific operation makes use of a 1×1 convolution.

$$O'_k = \text{CONV}_{1 \times 1}(\text{ReLU}(\text{CONV}_{1 \times 1}(G(O_k)))) \quad (13)$$

$$D'_k = \text{CONV}_{1 \times 1}(\text{ReLU}(\text{CONV}_{1 \times 1}(G(D_k)))) \quad (14)$$

It calculates the average of O'_k and D'_k to obtain Z_k . Here, it models the importance of various channels explicitly by using a sigmoid as a gating mechanism to create heavy objects over every aspect channel of Z . Using the ReLU activation function, the result vector Z_k is unsqueezed and then multiplied by O_k to get $\widehat{O}_k$.

$$\text{scale}_k = \text{unsqueeze}(\sigma(Z_k)) \quad (15)$$

$$\widehat{O}_k = \text{ReLU}(O_k, \text{scale}_k) \quad (16)$$

Here, the sigmoid function is denoted by $\sigma(\cdot)$. The channel dimension is used to combine the feature maps $\widehat{O}_k$ and D_k . D_{k-1} is obtained by carrying out a sequence of 3×3 convolution operations. Within Z_k , the Attentional Spatial Encoder (ASE) automatically discovers the importance of every feature channel. As a result, it achieves an adaptive calibration of feature channels by amplifying important features and attenuating those that are less relevant to the current task.

3.4.2 VCSO Hyper parameter Tuning

In this paper, VCSO is utilised to tune the hyper parameters of UATNet classification.

A. *Basic Definitions and Concepts*

VCSO is a tactic that takes advantage of the basic characteristics of viral infection and replication and is based on the complex interactions between viruses and host cells. The ability of viruses to efficiently replicate and adjust to shifting environmental conditions drives their evolutionary trajectory. Viruses depend on host cells for their survival and propagation. This method clarifies the strategies for viral colony survival, including infection of host cells, proliferation, and defence mechanisms.

B. *Mathematical Modelling*

VCSO consists of three unique approaches:

1. **Gaussian Walking Method:** This method improves the cognitive or extraction part of the approach by modelling the spread of the virus.
2. **Covariance Matrix Method:** It supports the social behaviour or search component and is used to model host cell damage.
3. **Evolutionary Strategy:** This part fills in the gaps left by the previous two strategies by working to counteract the immune system's reaction to the virus. It improves on the results of the first two parts by doing away with weaker viruses and encouraging the replication of stronger ones.

VCSO depends on the following five fundamental presumptions to arrive at its final solution:

1. **Two Solution Groups:** The Host Cell Colony (H_{pop}) and the Virus Colony (V_{pop}) are the two separate sets of solutions that VCSO uses to run its operations. These sets are essential to the operation of the VCSO algorithm.
2. **Unique Viral Solutions:** Throughout the transmission phase, every virus creates a different solution. One important aspect of the VCSO approach is its diversity.
3. **Viral Infection and Destruction:** To ensure their survival and spread, viruses naturally display traits aimed at both infecting and destroying host cells.
4. **Nutrient Acquisition Through Host Cell Destruction:** The ability of a virus to break apart host cells and extract vital nutrients required for its own survival and replication is a prerequisite for virus reproduction.
5. **Selective Viral Replication:** Only the most skilled viruses are kept alive during each round of replication in response to the host cell's immune system. The most efficient viral solutions will proliferate thanks to this selective process.

C. *Virus Spread*

Viruses are common and can be found in a variety of settings, such as the water and air. Gaussian walking algorithm, which is able to avoid local optima, is used as a suitable modelling technique to depict the virus's search for a host cell. The following formula expresses this strategy:

$$V_{\text{pop}_i} = \text{Gaussian}(G_{\text{best}}^g, \tau) + (r_1 \cdot G_{\text{best}}^g - r_2 \cdot V_{\text{pop}_i}) \quad (17)$$

In the set $\{1, 2, \dots, N\}$, i represents the random selection index. The relation $\frac{\log(g)}{g \cdot (V_{\text{pop}_i} - G_{\text{best}}^g)}$ yields the standard deviation τ for the Gaussian parameter. The vector in the formula above, $r_1 \cdot G_{\text{best}}^g - r_2 \cdot V_{\text{pop}_i}$, is oriented to steer clear of local points. Moreover, it is thought that $\log(g)/g$ enhances the performance of local searches. With additional programme iterations, this coefficient's fluctuations gradually decrease and it offers more guidance towards the ultimate optimal solution. Initially, this coefficient exhibits higher levels of fluctuation. Additionally, $(G_{\text{best}}^g, \tau)$ ensures that better solutions are generated based on the final solution's guidance, which is oriented towards the vector $(r_1 \cdot G_{\text{best}}^g - r_2 \cdot V_{\text{pop}_i})$.

D. Host Cell Susceptibility

A virus that infects a cell causes the cell's internal structure to gradually deteriorate until the cell dies. The most useful model for explaining this phenomenon is one based on a covariance matrix, which can be understood by following these steps:

Step 1: Updating H_{pop} by the relation

$$H_{\text{pop}_i}^g = X_{\text{mean}}^g + \sigma_i^g \times N_i(0, C^g) \quad (18)$$

$N_i(0, C^g)$ is obtained from a matrix of covariance of v with $D \times D$ dimensions and a normal distribution with a mean of 0, where g represents the current programme iteration and $\sigma^g > 0$. The first value that follows is X_{mean}^g .

$$X_{\text{mean}}^0 = \frac{1}{N} \sum_{i=1}^N V_{\text{pop}_i} \quad (19)$$

Step 2: Selecting and assessing the best γ vector to use as a parent the vector in the following centrality from the previous segment

$$X_{\text{mean}}^{g+1} = \frac{1}{\gamma} \sum_{i=1}^{\gamma} \omega_i \cdot V_{\text{pop}_i^{\gamma_{\text{best}}}} \left| \omega_i = 1n(\gamma + 1) / \left(\sum_{j=1}^{\gamma} (1n(\gamma + 1) - 1n(j)) \right) \right| \quad (20)$$

The evolutionary process proceeds along two different paths, which are represented by the following equations, with γ being defined as the floor value of where $\gamma = \lfloor N/2 \rfloor$ and ω_i representing the combination coefficient, where i represents the best solution within the solution set.

$$P_{\uparrow\sigma}^{g+1} = (1 - C_{\sigma}) + \sqrt{C_{\sigma}(2 - C_{\sigma})} \gamma_{\omega} \frac{1}{\sigma^g} (C^g)^{-\frac{1}{2}} (X_{\text{mean}}^{g+1} - X_{\text{mean}}^g) \quad (21)$$

$$P_{\uparrow\sigma}^{g+1} = (1 - C_c) + h_{\sigma} \sqrt{C_c(2 - C_c)} \gamma_{\omega} \frac{1}{\sigma^g} (X_{\text{mean}}^{g+1} - X_{\text{mean}}^g) \quad (22)$$

Such that $\gamma_w^{-1} = \sum_{i=1}^{\gamma} w_i^2 \cdot (C^g)^{-0.5}$ is both positive and symmetric, satisfying the formula $(c^g)^{-0.5} (c^g)^{-0.5} = (c^g)^{-1}$. $C_\sigma = (\gamma + 2)/(N + \gamma_w + 3)$, $C_C = 4/(4 + N)$, and $h_\sigma = 1$ are typically used to set computational parameters. Nonetheless, P_σ^{g+1} large if $h_\sigma = 0$.

Step 3: The covariance matrix C^{g+1} and the value of σ^{g+1} are updated with

$$\sigma^{g+1} = \sigma^g \times \exp\left(\frac{C_\sigma}{d_\sigma} \left(\frac{P_\sigma^{g+1}}{\text{EN}(0, 1)} - 1\right)\right) \quad (23)$$

$$\begin{aligned} C^{g+1} = & (1 - C_1 - C_\gamma) C_1 P_C^{g+1} (P_C^{g+1})^T \\ & + C_\gamma \sum_{i=1}^{\gamma} w_i \frac{(V_{\text{pop}_i^{\gamma^{\text{best}}}} - X_{\text{mean}}^g)}{\sigma^g} \cdot \frac{(V_{\text{pop}_i^{\gamma^{\text{best}}}} - X_{\text{mean}}^g)}{\sigma^g} \end{aligned} \quad (24)$$

where $d_\sigma = 1 + C_\sigma + 2 \max\{0, (\sqrt{\gamma_w - 1}/\sqrt{N - 1}) - 1\}$ is usually close to C_γ , where C_γ acts according to the following relation:

$$C_1 = \frac{1}{\gamma_w} \left(1 - \frac{1}{\gamma_w}\right) \min\left\{1, \frac{2\gamma_w - 1}{(N + 2)^2 + \gamma_w}\right\} + \frac{1}{\gamma_w} \frac{2}{(N + \sqrt{2})^2}, C_\gamma = (\gamma_w - 1)C_1 \quad (25)$$

In which, $0 \leq C_\gamma \leq 1$ is the covariance matrix C update rate.

E. Performance of the Immune System

As was covered in earlier sections, more powerful viruses can carry on replicating even after they have evaded the immune system's defences. Conversely, the immune system effectively neutralises less potent viruses. This evolutionary progression can be summarised using the steps that follow:

Step 1: Calculating the V_{pop} population's performance metric Pr by assessing the system's intrinsic objective function

$$\text{Pr}_{\text{rank}(i)} = \frac{N - i + 1}{N} \quad (26)$$

Step 2: Dividing the V_{pop} population into individual populations using equation

$$\begin{cases} V_{\text{pop}_{i,j}} = V_{\text{pop}_{k,j}} - \text{rand} \cdot (V_{\text{pop}_{h,j}} - V_{\text{pop}_{i,j}}) \\ \text{if } r > \text{Pr}_{\text{rank}(i)} \\ V_{\text{pop}_{i,j}} = V_{\text{pop}_{i,j}} \\ \text{otherwise} \end{cases} \quad (27)$$

Table 2 Specifications table

Specifications	Value
CPU	1.5–2.7GHZ
GPU	920 m NVidia
RAM	12 GB
Generation	4th
Internet	8Mbps upload, 8Mbps download

In the provided equations, $j \in [1, 2, 3, \dots, d]$, and the indices k, i , and h are chosen at random from the set $[1, 2, 3, \dots, N]$, provided that they are distinct ($i \neq h \neq k$). Furthermore, random values between 0 and 1 are assigned to the variables rand and r . The aforementioned formulas show that every solution makes an effort to keep each step's best current value. Furthermore, a reiteration process is applied to a solution in order to bring it back within the legitimate problem boundaries if it happens to fall outside the acceptable range specified by the upper limit (up_{ij}) and lower limit (low_{ij}).

$$x_{ij} = \text{low}_{ij} + \text{rand} \times (\text{up}_{ij} - \text{low}_{ij}) \quad (28)$$

4 Results and Discussion

4.1 Experimental Setup

An 12 GB RAM and 500 GB hard drive Core i5 computer will be used for the trials. The preferred programming language is Python 3. Anaconda and Jupyter Notebook serve as the backend's cornerstones. Jupyter Notebook is being used for this study because, as Table 2 states, it will enable us to utilise the benefits of the online servers.

4.2 Performance Metrics

The output measures are precision (PRE), recall (RE), accuracy (ACC), and the F1-score (F1) shown in Eqs. (29)–(32). Based on these metrics, the following table compared the actual results to those that were anticipated:

$$\text{Accuracy} = \frac{(\text{TRP} + \text{TRN})}{\text{TRP} + \text{FLP} + \text{TRN} + \text{FLN}} \% \quad (29)$$

$$\text{Precision} = \frac{\text{TRP}}{\text{TRP} + \text{FLP}} \quad (30)$$

Table 3 Analysis of feature extraction

Models	ACC (%)	PRE (%)	RE (%)	F1 (%)
AE	79	80	81	78
VAE	82	83	83	82
LSTM	86	84	85	84
GRU	90	89	89	89
Proposed MobileNet model	96	94	93	93

$$\text{Recall} = \frac{\text{TRP}}{\text{TRP} + \text{FLN}}$$

(31)

$$\text{F1 - score} = \frac{2\text{TRP}}{2\text{TRP} + \text{FLP} + \text{TRN}}$$

(32)

TRP stands for “true positive”, FLP for “false positive”, TRN for “true negative”, and TRN for “true negative”.

4.3 Feature Extraction Validation

A complete overview of the feature extraction performance evaluation is given in Table 3, with a special focus on comparing the suggested model with other existing models.

The following are the performance metrics for the different models from Table 3 and Fig. 9: The autoencoder (AE) model obtained an F1-score of 78%, accuracy (ACC) of 79%, precision (PRE) of 80%, and recall (RE) of 81%. The results of the Variational Autoencoder (VAE) model were better, with an F1-score of 82%, an ACC of 82%, a PRE of 83%, and a RE of 83%. Additional improvements were shown by the Long Short-Term Memory (LSTM) model, which achieved an ACC of 86%, PRE of 84%, RE of 85%, and an F1-score of 84%. The model known as the Gated Recurrent Unit (GRU) demonstrated noteworthy advancement, attaining an ACC of 90%, PRE of 89%, RE of 89%, and an F1-score of 89%. With an astounding ACC of 96%, PRE of 94%, RE of 93%, and an F1-score of 93%, the suggested MobileNet model, in contrast, performed better than any of the others.

4.4 Classification Validation

A complete performance analysis of the classification task is provided in Table 4, which also provides a comparison of the suggested model with other existing models.

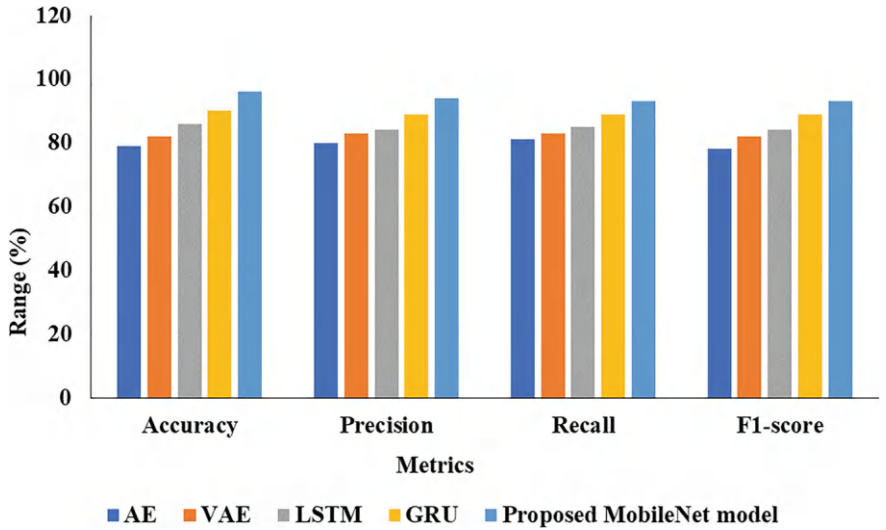


Fig. 9 Feature extraction validation

Table 4 Evaluation of classification models

Models	ACC (%)	PRE (%)	RE (%)	F1 (%)
VGG-19	92	91	93	90
ResNeXt-101	83	84	85	84
Inception-ResNet-v2	93	92	94	89
SE-ReesNet-101	95	94	95	94
Proposed VCSO-UATNet model	98	97	97	96

Notable performance metrics were found in our comparative study of multiple models from Table 4 and Fig. 10. With an ACC of 92%, PRE of 91%, RE of 93%, and an F1 of 90%, the VGG-19 model demonstrated impressive performance. The ResNeXt-101 model performed well, with an F1-score of 84%, ACC of 83%, PRE of 84%, and RE of 85%. An ACC of 93%, PRE of 92%, RE of 94%, and an F1-score of 89% were demonstrated by Inception-ResNet-v2. Impressively accurate, the SE-ResNet-101 model had an ACC of 95%, PRE of 94%, RE of 97%, and an F1-score of 94%.

With remarkable results including an ACC of 98%, PRE of 97%, RE of 97%, and an F1-score of 96%, the suggested VCSO-UATNet model stood out as the best performer. The VCSO-UATNet model is superior because of its creative design, which combines an attention-based U-Shape Transformer Network (UATNet) with virus colony search optimisation (VCSO). Effective hyper parameter optimisation is achieved by VCSO, and complex spatiotemporal dependencies in the data are captured by UATNet. This combination produces a model that is exceptionally

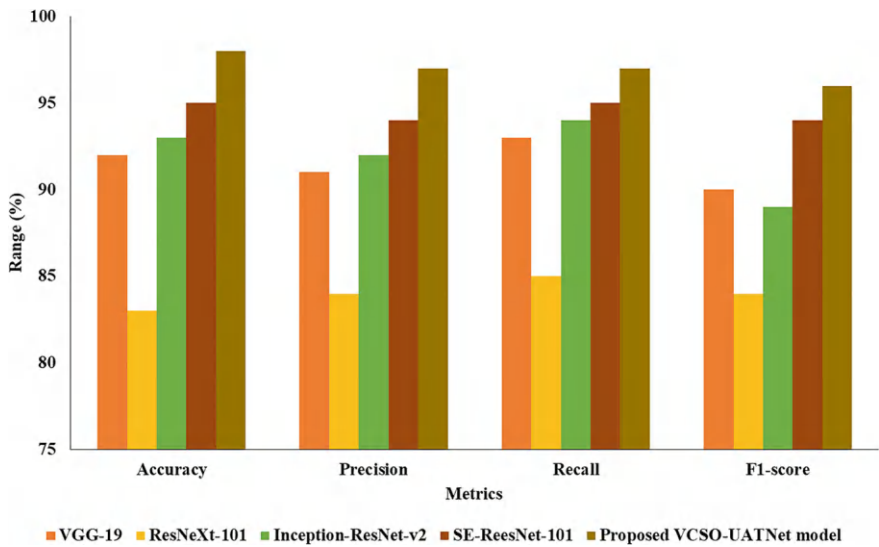


Fig. 10 Classification analysis

good at interpreting the complex connections in the dataset, which accounts for the unmatched accuracy that has been seen. Thus, the VCSO-UATNet model serves as evidence of the superior performance that can be achieved in AQP through the use of sophisticated model architectures and optimisation techniques.

5 Conclusion

It has delved into the field of DL to enhance AQP, and the results have the potential to be revolutionary. Predicting AQ is a vital task that is vital for protecting the environment and public health, and the recommended method that makes use of cutting-edge technologies has produced notable results. The CPCB’s air pollution databases serve as the study’s main source of information. In the preprocessing stage of this study, ASGF was introduced to guarantee data consistency and integrity. Like a skilled curator, the ASGF carefully eliminated noise and smoothed the data, giving the DL models a clean dataset that was the foundation for exceptional prediction. With its strategic choice of MobileNet for feature extraction, the use of cutting-edge DL architectures was demonstrated. Well-known for its effectiveness and skill in identifying important characteristics from various data formats, MobileNet made complex AQ data easier to understand. UATNet is a cutting-edge DL model that was specifically designed to handle the complex spatiotemporal dependencies present in AQ data, and it is the key to forecast success. By utilising transformer architectures

and attention mechanisms, the UATNet was able to decipher the intricate relationships present in the data and go deep into the subtleties of AQ. This work optimised the UATNet's hyper parameters using VCSO in an attempt to achieve optimality. This novel method served as a catalyst that allowed us to fully utilise the potential of the suggested VCSO-UATNet model. In the area of AQP, our suggested model outperforms other models with an astounding accuracy rate of 99%. These results highlight the critical importance of DL in addressing the worldwide problem of air pollution, highlighting its role in protecting public health and maintaining the environment. Future work will involve expanding data sources and improving models to further improve AQ forecasts.

References

1. Kalajdjieski J, Zdravevski E, Corizzo R, Lameski P, Kalajdziski S, Pires IM, Trajkovik V (2020) Air pollution prediction with multi-modal data and deep neural networks. *Remote Sens* 12(24):4142
2. Iskandaryan D, Ramos F, Trilles S (2020) Air quality prediction in smart cities using machine learning technologies based on sensor data: a review. *Appl Sci* 10(7):2401
3. Ma J, Cheng JC, Lin C, Tan Y, Zhang J (2019) Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques. *Atmos Environ* 214:116885
4. Tao Q, Liu F, Li Y, Sidorov D (2019) Air pollution forecasting using a deep learning model based on 1D convnets and bidirectional GRU. *IEEE Access* 7:76690–76698
5. Chang YS, Chiao HT, Abimannan S, Huang YP, Tsai YT, Lin KM (2020) An LSTM-based aggregated model for air pollution forecasting. *Atmos Pollut Res* 11(8):1451–1463
6. Wei W, Ramalho O, Malingre L, Sivanantham S, Little JC, Mandin C (2019) Machine learning and statistical models for predicting indoor air quality. *Indoor Air* 29(5):704–726
7. Guo Q, He Z, Li S, Li X, Meng J, Hou Z, Chen Y (2020) Air pollution forecasting using artificial and wavelet neural networks with meteorological conditions. *Aerosol Air Qual Res* 20(6):1429–1439
8. Vu TV, Shi Z, Cheng J, Zhang Q, He K, Wang S, Harrison RM (2019) Assessing the impact of clean air action on air quality trends in Beijing using a machine learning technique. *Atmos Chem Phys* 19(17):11303–11314
9. Liu Y, Nie J, Li X, Ahmed SH, Lim WYB, Miao C (2020) Federated learning in the sky: aerial-ground air quality sensing framework with UAV swarms. *IEEE Internet Things J* 8(12):9827–9837
10. Li J, Shao X, Sun R (2019) A DBN-based deep neural network model with multitask learning for online air quality prediction. *J Control Sci Eng* 2019
11. Zou X, Zhao J, Zhao D, Sun B, He Y, Fuentes S (2021) Air quality prediction based on a spatiotemporal attention mechanism. *Mob Inf Syst* 2021:1–12
12. Feng Y (2022) Air quality prediction model using deep learning in Internet of Things environmental monitoring system. *Mob Inf Syst* 2022
13. Gu Y, Li B, Meng Q (2022) Hybrid interpretable predictive machine learning model for air pollution prediction. *Neurocomputing* 468:123–136
14. Jin XB, Wang ZY, Kong JL, Bai YT, Su TL, Ma HJ, Chakrabarti P (2023) Deep spatio-temporal graph network with self-optimization for air quality prediction. *Entropy* 25(2):247
15. Hu Y, Chen X, Xia H (2023) A hybrid prediction model of air quality for sparse station based on spatio-temporal feature extraction. *Atmos Pollut Res* 14(6):101765

16. Fang W, Zhu R, Lin JCW (2023) An air quality prediction model based on improved Vanilla LSTM with multichannel input and multiroute output. *Expert Syst Appl* 211:118422
17. Wu Z, Liu N, Li G, Liu X, Wang Y, Zhang L (2023) Meta-learning-based spatial-temporal adaption for coldstart air pollution prediction. *Int J Intell Syst* 2023
18. Aarthi C, Ramya VJ, Falkowski-Gilski P, Divakarachari PB (2023) Balanced spider monkey optimization with Bi-LSTM for sustainable air quality prediction. *Sustainability* 15(2):1637
19. Shu Y, Ding C, Tao L, Hu C, Tie Z (2023) Air pollution prediction based on discrete wavelets and deep learning. *Sustainability* 15(9):7367
20. https://app.cpcbcr.com/AQL_India/
21. https://en.wikipedia.org/wiki/Air_quality_index#cite_note-20
22. Srilatha P, A machine learning model for air quality prediction for smart cities. *Design engineering*, ISSN: 0011-9342
23. Dombi J, Dineva A (2020) Adaptive Savitzky-Golay filtering and its applications. *Int J Adv Intell Paradig* 16:145–156
24. Fu J, Chen X, Lv Z (2022) Rail fastener status detection based on MobileNet-YOLOv4. *Electronics* 11(22):3677
25. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. *arXiv* [arXiv:2103.14030](https://arxiv.org/abs/2103.14030)
26. Huang Z, Ben Y, Luo G, Cheng P, Yu G, Fu B (2021) Shuffle transformer: rethinking spatial shuffle for vision transformer. *arXiv* [arXiv:2106.03650](https://arxiv.org/abs/2106.03650)

Consumer Sentiment Analysis Using ROA-Based Hybrid Deep Learning Model from Twitter Data



Sreekanth Rallapalli , K. Jayasheel Kumar, and Tarunika Sharma

Abstract Every day, the amount of words created increases at an exponential rate. Possible sources include: in-app messengers like Telegram and WhatsApp; social media platforms like Instagram and Facebook; online retailers like Amazon; Google searches; news publishing platforms; and many others. There is a constant flood of textual information from all these sources. The analysis of this kind of information can provide valuable insight for business owners curious about the public's perception of their product, brand, or service. Many methods have been presented recently to get understanding from these records. Accurate polarity identification of consumer evaluations is, nevertheless, a continuous and intriguing subject due to the difficulties associated with dealing with texts of large size. This makes it difficult to precisely interpret the linguistic data contained in things like customer feedback and social media remarks. Extensive efforts have been made in the past to streamline the process of deriving precise interpretations from this data. The CNN-LSTM-Attention (CLATT) method is a novel approach to sentiment analysis that utilises a convolutional neural network (CNN) as an encoder and a long short-term memory network (LSTM) as a decoder. This method also encompasses data collection, preprocessing, feature encoding, and classification. When analysing such data, it is essential to consider how the data were collected, processed, and organised. Rat Optimisation Algorithm (ROA) is used to fine-tune the hyper-parameters of the suggested model, boosting its classification efficacy. The research evaluated the significance of the proposed models using various textual datasets. Better or at least equivalent results may be shown when using the suggested method of predicting attitudes, and the computational complexity is reduced. The results of this study highlight the critical role that sentiment analysis plays in deducing useful information from social media posts and customer reviews.

S. Rallapalli (✉) · K. Jayasheel Kumar · T. Sharma
Nitte Meenakshi Institute of Technology, Yelahanka, Bengaluru, India
e-mail: sreekanth.rallapalli@nmit.ac.in

New Horizon College of Engineering, Bengaluru, India

Keywords Convolutional neural network · Consumer sentiment analysis · Rat optimization algorithm · Long short-term memory network · Amazon

1 Introduction

Due to developments in science and communication networks, the globe is now commonly referred to as a “global village” [1]. Social media is used by more than half of the world’s fun, but also for research, advertising, networking, and anything else that can be done online. Therefore, we live in a digital era and are highly reliant on IT [2, 3]. This reliance results in tweets, postings, and consumer evaluations concerning various items. Data that is both duplicated and inconsistent is a memory hog and a source of frustration for users and system administrators alike [4]. The data suffer from polarity problems because of this repetition. In order to flourish, businesses need to understand their consumers’ wants, feelings, and actions [5]. The solution to these problems is sentiment analysis. It plays an important function in polarity detection and text categorization.

Sentiment investigation is a method for detecting ambiguity in text, including language, opinions, and more. The term “opinion mining” is another term for this practise. How a spokesperson and a user feel about an issue is revealed through sentiment analysis [6, 7]. One’s own viewpoints or feelings are often at the heart of the tone they choose to communicate in their work [8]. Many new algorithms have been introduced to analyse, predict, and evaluate emotions based on text data like product or customer reviews. Polarity identification can benefit greatly from the use of sentiment analysis [9]. Problems with spam and fake data, domain dependency, negation, NLP’s computational burden, bipolar terminology, and a large vocabulary are all present [10]. Data mining can be made more efficient and effective if the aforementioned issues are fixed [11].

Several academics have looked at the challenges of sentiment analysis. Sentiment analysis of user reviews using deep learning [12] was chosen because of its practicality and impact in the development of various innovative apps. Extracting from the provided text, such as customer reviews, is the goal of sentiment analysis through categorization [13], which aims to solve the aforementioned problems. As a result of their widespread use and proven effectiveness, deep learning techniques are now within reach of producing adequate results. In this research, we use a deep learning approach to classify online reviews by sentiment, describing the semanticization of customer evaluations as a whole. The Snap dataset [14] was used to compile customer feedback on various cell phone and accessories products sold on Amazon. Having realised the significance of sentiment analysis, the system aids in improving the process of living. With the maximum degree of certainty and the least amount of processing complexity, the proposed strategy yields superior or, at the very least, similar outcomes [15].

Data cleaning, normalisation, hashtag removal, punctuation removal, text to lowercase conversion, and tokenization are only some of the preprocessing activities

that have been explored and investigated in this study and their impact on consumer evaluations. Predicting customer sentiment is approached in this work by using a model dubbed CLATT with a sliding window approach. Time series data, such as trends, can be captured locally by a CNN, and then integrated into long-term dependencies by an LSTM. By developing a CNN-LSTM hybrid model, the suggested strategy enhances nonlinear modelling capability. The output of the LSTM is integrated across consecutive sample intervals, and information from prior time periods is extracted using an attention method. Section 3 lays out in depth all of the preparatory procedures that went into producing this work. Data selection, preprocessing, and classification were important areas of improvement made by this effort. The experimental analysis is obtainable in Sect. 4, and a conclusion is haggard in Sect. 5.

2 Related Works

By incorporating semi-supervised adversarial learning into the BERT model, Jain et al. [16] propose the fusion, which enhances the model's fine-tuning performance. To achieve our goal, we parsed customer evaluations for information about the quality of the service provided, then combined that information with the word sequences to put into the model. That's useful for adding positional information and aspectual representation into sentences. Our study and presentation of the findings demonstrate the accuracy improvement brought by the provided model compared to those of previously-discovered models.

For CRD forecasting, Jain et al. [17] suggest a Sparse Self-Attention Network (SpSAN). The sparsity introduced by the proposed SpSAN into the self-attention process boosts the Bidirectional Encoder model's fine-tuning performance. By modifying the softmax function with a tunable sparse transformation at the moment of fine-tuning using BERT, this study incorporates sparsity into the self-attention process. It gives us a more comprehensible picture of the full input material, allowing us to comprehend sparse attention distribution. The significance of the suggested SpSAN model is described by experimental data and their analysis.

In-depth use of a lexicon-based approach to attitude analysis is at the heart of the qualitative study conducted by Thao et al. [18] on vegetarian food preferences among online shoppers. Customer reviews may be used to infer feelings about five different factors: pricing, packaging, shipping, brand, and quality. In addition to the star rating, the tags can also be assigned a favourable, negative, or neutral opinion. There are 17,892 feedback records in the dataset. The results showed that, on average, customers felt positively about 20% of the five factors studied, negatively about 0.7%, and indifferent about 78.55%. The quality of the product is the biggest issue among the five categories of consumer feedback. There are 8672 good comments on this. Additionally, there are almost 5000 good tags related to the quality and packaging features, while the shipment has the most negative tags, with 26 rating tags. In addition, the overall rating average is 4.94 out of 5.0, which is quite high.

A unique unsupervised sentiment categorization approach was suggested by Punetha and Jain [19]. In order to do sentiment and emotion analysis of reviews, this research proposes a mathematical optimisation framework without any supervision. Two functions are accomplished by the suggested model. It does this in two ways: first, by determining whether or not a consumer is satisfied with their experience, and second, by identifying the positive and negative sentiment polarities in a review. There are two phases to the structure. First, we produce performance scores by averaging reviewers' context, rating, and emotion evaluations. Nash Equilibrium is reached in the second phase by applying a non-cooperative game to the evaluation of performance. The step's output is the review's implied tone and the customer's perceived level of satisfaction. State-of-the-art findings were obtained in the tests conducted on two restaurant review datasets. Statistical analysis was used to confirm and demonstrate the significance of the findings. The suggested model works in any domain or language. The suggested paradigm guarantees coherent and reasonable outcomes.

Based on research on travellers' opinions, Yuan et al. [20] propose an enhanced support vector machine (SVM) method and develop a Map-Reduce-based Hadoop Distributed File System (HDFS). To begin, the text of online travel evaluations is pre-processed for sentiment analysis. Second, to further enhance the precision of emotion word classification, a new SVM technique is presented that draws on the most salient aspects of linear classification and kernel functions. The real tourist application context is then used to install HDFS data nodes on top of the Hadoop framework. And the map function and reduction function are created and implemented based on the Map-reduction programming paradigm, which considerably enhances the capability of parallel processing and minimises the time consumption simultaneously. The completed Hadoop infrastructure now runs on an enhanced SVM algorithm. The results of the experiments suggest that internet travel evaluations may be a valuable source of information for the proposed algorithm can efficiently classify travellers' moods about their trips.

A deep learning method for analysing Twitter data on opinions on COVID-19 has been presented by Singh et al. [21]. An LSTM-RNN network with boosted featured weighting through attention layers forms the basis of the proposed approach. By way of the attention mechanism, this algorithm employs a refined framework for transforming features. This research made use of publicly accessible Twitter data from the Kaggle database, labelled with one of four classes: sadness, joy, fear, or rage. The suggested deep learning strategy greatly improves approaches. This improvement was achieved by combining attention approach. We observed that out of a total of 179,108 tweets about COVID-19, 45% were good, 30% were neutral, and 25% were negative.

3 Proposed Model

3.1 Data Collection

Both the STS-Gold [22] and SemEval-2014 restaurant reviews [22] datasets have been used to evaluate the projected model and compare it to the current state-of-the-art approaches. There are 3000 training reviews and 800 testing input reviews in the SemEval-2014 restaurant reviews dataset. At least one aspect phrase is labelled for each review in this dataset. The Sentiment140 dataset contains 1.6 million tweets with sentiment annotations ranging from 0 (negative) to 4 (positive). Identifiers and sentiments for 2026 tweets are included in the STS-Gold dataset.

3.2 Data Pre-Processing

Textual analyses rely heavily on the results of the data preparation phase. Text data grow more complex as a result of the recurrence and redundancy in tweets, blogs, reviews, and other forms of text. Preprocessing the data is a filtering technique used in the normalisation process. Data preparation includes, but is not limited to, normalisation, word tokenization, removal of stop words, removal of superfluous spaces, padding, transforming text data to lowercase, and removal of hashtagging. In order to get the data in the correct format, this effort implemented several different jobs.

3.2.1 Erase Punctuation

Nearly 40–50% of each written document is made up of punctuation. Sentiment analysis models are completely immune to the effects of punctuation. These punctuation marks are irrelevant to the analysis and must be removed.

At this point, we had already normalised the data and stripped out any punctuation from the text. The final product is a streamlined and summarised text. The manuscript has all punctuation removed. The following sentence contains unnecessary punctuation and illustrates why it should be removed: “Good day, everyone!!!! A more natural translation of “I’ve been with IDS since 2012.”

3.2.2 Convert the Text Data to Lowercase

In the customer reviews, users write content that mixes lower- and upper-case letters, ignoring grammatical conventions. The study relies heavily on approaches that are highly specific to each individual situation. This causes the classifier to have trouble identifying the text’s polarity. Fortunately, this problem is easily avoidable

by adopting a uniform format for the entire content. On the other hand, the lower-case (txt) statement is used when the identical action must be done by hand. It changes all the capital letters to lowercase and ignores the other characters. The following is an illustration of how to lowercase text data: You may rephrase “I Am A Senior Big Data Analyst in Islamabad” as “I Am a Senior Big Data Analyst in Islamabad” if you like.

3.2.3 Tokenization of the Text

Tokenization is a method for processing large amounts of text by breaking it down into smaller tokens, often phrases. Text is broken up into tokens. The goal of this method is to simplify difficult textual problems. By replacing passwords with tokens, data mining may be automated and simplified. Lexical assessment, semantic analysis, and emotional context can all benefit from tokenization. Tokenization is a critical stage in the natural language processing (NLP) workflow. Before we can begin developing the model, though, the text must be cleaned up. Word further types of tokenization. You may use this tokenized version to:

- Count the sum of words in the text;
- Sum the frequency of words. The text data are split into words in this phase.

Smaller groups of words or symbols are extracted from a much larger and more complicated document. As an illustration, the phrase “I am a data analyst in Islamabad” may be broken down into its component tokens: “I,” “am,” “data,” and “analyst.”

3.2.4 Removal of Stop Words

In large text files, many words tend to occur repeatedly. Therefore, it is essential that the stop words be removed. True, filler words never add value to the text they’re included in. This category of terms appears often throughout the book. This makes text mining more challenging and causes inconsistent results from classifiers. At this point, the data selection has had the stop words removed. This method reduces the amount of textual information while simultaneously raising the system’s productivity. To illustrate, the previous line may become “I am a data analyst in Islamabad” if the stop words were removed.

3.2.5 Removal of the Hyperlink

Hyperlinks in any dataset are meaningless at this point. All of the connections have a practical purpose. However, we just use the gathered data to refine the text’s polarity through the use of tweets, comments, and reviews as representations of ideas and feelings. Therefore, it is essential to eliminate the ties between the datasets.

3.2.6 Removal of Hash Tag

The use of hashtags is also widely used today. Hashtagging is often used in user evaluations. Hashtags are quite memory intensive. When it comes to analysing sentiment, hashtags just don't cut it. These just serve to increase the classifiers' difficulty. Therefore, it's crucial to get rid of hashtags. Hashtags are removed from the datasets to produce more concise and informative training data.

3.2.7 Removal of Unnecessary Spaces

Sentiment analysis using a classifier has difficulties when there is spare space in the raw datasets. To prevent this issue and save considerable time during sentiment analysis, all unnecessary spaces are removed during the preprocessing phase.

3.2.8 Padding

Problems arise for the classifier analysis because of the presence of both very short and very long reviews databases. CNN-related padding refers to the extra pixels that are added to the review before it is sent to the network for analysis. Adding zeroes as padding at the conclusion of our input review guarantees that all customer reviews are exactly the same length.

3.2.9 POS Tagging

Words in the training data can be organised according to their grammar using POS. They look at the words in their natural settings. The process of point-of-sale labelling is complex. While point-of-sale labelling isn't a panacea for the massive discovery problem in opinion examination, it does help much with reordering many issues. In this way, a product evaluation was able to take into account many different perspectives. The POS tagger was tweaked so that it could be used to define a certain set of capabilities. The review provides access to the POS tagging tool from Grammar Relations. Examinees should utilise the speech part tags to identify parts of speech. Moreover, POS tags identified substantial and suitable substantive as potential issues. To create the observable output, this POS method may be thought of as employing a Concealed Markov Model (HMM), in which the tags are concealed. We constantly try to find a tag sequence (C) that mathematically optimises as:

$$P(C|W) \tag{1}$$

where C denotes $C1, C2, C3, \dots, CT$, and W denotes $W1, W2, W3$, and WT .

3.3 Feature Encoding for Arithmetical Representation of Textual Data

It’s possible that the produced datasets won’t be in a form usable by statistical or mathematical software. Extracting numeric features from textual input requires a suitable function encoding approach [23]. Each review in the sample has to be represented mathematically in a way that reflects the real semanticist wording of the review. The recommended numerical characteristics are subsequently implemented in the subsequent processing and analysis stage.

3.3.1 Word Embedding

Word embedding is a method by which each word is given a numeric and vectorial representation. The term “word embedding” is used to describe texts in which synonyms are represented by their precise letter values. Word embedding, in particular, is a form of unsupervised learning of word representation that shares certain similarities with semantic similarity. Words are said to be “coordinated” when they are grouped together in accordance with a predetermined set of relationships [24].

3.4 Classification Using Encoder--Decoder

Figure 1 depicts the model’s encoder--decoder architecture, which employs a convolutional neural network (CNN) as the encoder and a (LSTM) as the decoder. The encoder’s job is to transform the input into features, while the decoder’s is to transform the features into the desired outcomes.

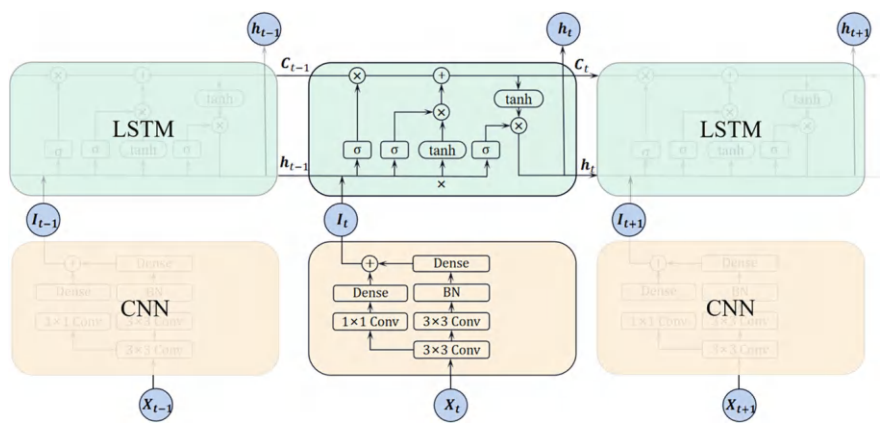


Fig. 1 CNN-LSTM structure

A convolutional neural network (CNN) like Resnet is used to create the encoder; it uses a residual structure to prevent over-fitting. First, in order to take use of the time dimension and the feature dimension that are already present in the input, two continuous 3 3 convolution layers are used. Subsequently, it will pass through a Batch Normalisation (BN) layer. In order to lower the overall dimensions of the combined feature, it is flattened and then goes through a Dense layer. The completed CNN module's output will be used as the input for a regular LSTM implementation. The LSTM component's output will be used to make predictions about the review's quality using the attention mechanism.

The current time, denoted by t_i , is located at the origin of the horizontal axis, while the other points, labelled $t_{(i-1)}$ through $t_{(i-n)}$, indicate times in the past. The vertical axis shows the relationship between the present and the past in terms of the passage of time. Values outside the blue region indicate considerable association; the bigger the value, the stronger the relationship. The suggested model's internal structure is seen in Fig. 2.

An attention technique for compiling global data was presented in ref. [25]. In this research, we utilise an attention mechanism that aggregates data from nearby time intervals; the attention mechanism's structure is analogous to a Dense layer with many that map to the time dimension of the LSTM's output. The projected effluent water quality is calculated by multiplying the weight by the LSTM output. The mechanism of focus may be calculated as follows:

$$k_i = \sum_t (h_t h_i) \quad (2)$$

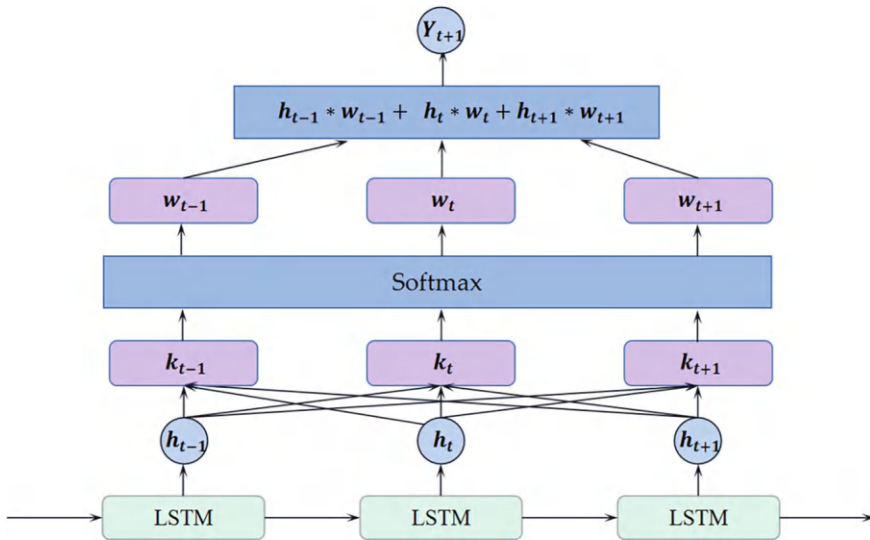


Fig. 2 Attention mechanism

$$w_i = \exp(k_i) / \exp\left(\sum_t (k_t)\right) \quad (3)$$

$$Y_{t+1} = \sum_i w_i h_i \quad (4)$$

where h_i characterises the output from LSTM, w_i is the weight of the consistent output, and Y_{t+1} refers to the predicted waste quality.

3.4.1 Hyperparameter Optimization Using ROA

The projected model's hyperparameters are tuned using the ROA technique.

Chasing the Prey

The overall tendency to hunt in groups. It is believed that the best search agent already knows roughly where the prey is. The relative placements of the other search agents are subject to change. The equations describing this process are as follows:

$$\vec{P} = A \cdot \vec{P}_i(x) + C \cdot (\vec{P}_r(x) - \vec{P}_i(x)) \quad (5)$$

This is where the placements of rats are defined, and where $P_r(x)$ is the optimal clipping limit.

A and C parameters, on are determined as shadows:

$$A = R - x \times \left(\frac{R}{\text{Max}_{\text{iteration}}} \right) \quad (6)$$

where, $x = 0, 1, \dots, \text{Max}_{\text{iteration}}$

$$C = 2 \cdot \text{rand}() \quad (7)$$

Therefore, both R and C are random statistics with values between [1] and [5]. Parameters A and C become increasingly crucial over subsequent rounds in optimising exploitation.

Fighting with Prey

The subsequent equation has been projected in order to quantitatively describe the fighting process between rats and their prey.

$$\vec{P}_i(x+1) = \left| \vec{P}_r(x) - \vec{P} \right| \quad (8)$$

$PI(x + 1)$ represents the rat's current site. Once the optimal answer has been captured, the other search agents are rearranged so that it may take centre stage. In order to get closer to its prey (A, B), the rat might switch its position from (A, B) to (A^*, B^*). Equations (6) and (7) demonstrate that the set of surrounding locations (8) will change depending on the values of the parameters. This concept, however, has potential for expansion in an n-dimensional setting. The new values for A and C ensure both may take place. The proposed RSO approach minimises the number of operators while maintaining high output quality. The RSO algorithm's pseudo code is included in the algorithm.

Algorithm: Proposed RSO Algorithm

Input: the rats population $P_i (i = 1, 2, \dots, n)$

Output: the optimal search agent for clipping limit

1. *Initialise the parameters A, C and R*
 2. *Calculate the fitness value of each search agent*
 3. $P_r \leftarrow$ *the best search agents*
 4. **While** ($x < Max_{iteration}$) **do**
 5. **for** each search agent **do**
 6. *Update the position of current search agent by Eq. (5)*
 7. **End for**
 8. *Update parameters A, C and R*
 9. *Check if there is any search agent which goes beyond the given search space and then adjust it*
 10. *Calculate the fitness of each search agent*
 11. *Update P_r if there is a better solution than precious optimal solution*
 12. $x \leftarrow x + 1$
 13. **End while**
 14. *Return P_r*
 15. **End procedure**
-

4 Results and Discussion

The proposed model went through this process again, and the averages of their accuracy, and specificity were tallied. The effectiveness of the suggested model is tested using the following equations.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (9)$$

$$\text{Precision} = \frac{TN}{TN + FP} \quad (10)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (11)$$

$$\text{F - Measure} = \frac{\text{TP}}{\text{TP} + 1/2(\text{FP} + \text{FN})} \quad (12)$$

where TP refers to the properly classified process, FP to the incorrectly classified one, FN to the incorrectly normal one, and TN to the correctly abnormal one.

4.1 Analysis of Proposed Model

Tables 1 and 2 presents the comparative analysis of various model in terms of different metrics.

In above Table 1 signifies that the Experimentation on proposed model without ROA. In the analysis of MLP model attained the accuracy as 0.769 and recall rate as 0.769 then precision as 0.771 and F1-score as 0.767 correspondingly. Then the AE model attained the accuracy as 0.761 and recall rate as 0.761 then precision as 0.764 and F1-score as 0.760 correspondingly. Then the DBN model attained the accuracy as 0.769 and recall rate as 0.769 then precision as 0.771 then precision as 0.767

Table 1 Experimentation on proposed model without ROA

Models	Accuracy	Recall	Precision	F1-Score
MLP	0.769	0.769	0.771	0.767
AE	0.761	0.761	0.764	0.760
DBN	0.769	0.769	0.771	0.767
CNN	0.771	0.771	0.775	0.769
LSTM	0.773	0.773	0.776	0.772
Attention	0.766	0.766	0.770	0.764
Hybrid model	0.783	0.783	0.785	0.782

Table 2 Experimentation on various models with ROA

Models	Accuracy	Recall	Precision	F1-Score
MLP	0.957	0.956	0.928	0.942
AE	0.925	0.920	0.967	0.943
DBN	0.958	0.812	0.873	0.841
CNN	0.963	0.944	0.802	0.867
LSTM	0.963	0.958	0.959	0.959
Attention	0.958	0.827	0.967	0.892
Hybrid model	0.974	0.974	0.974	0.974

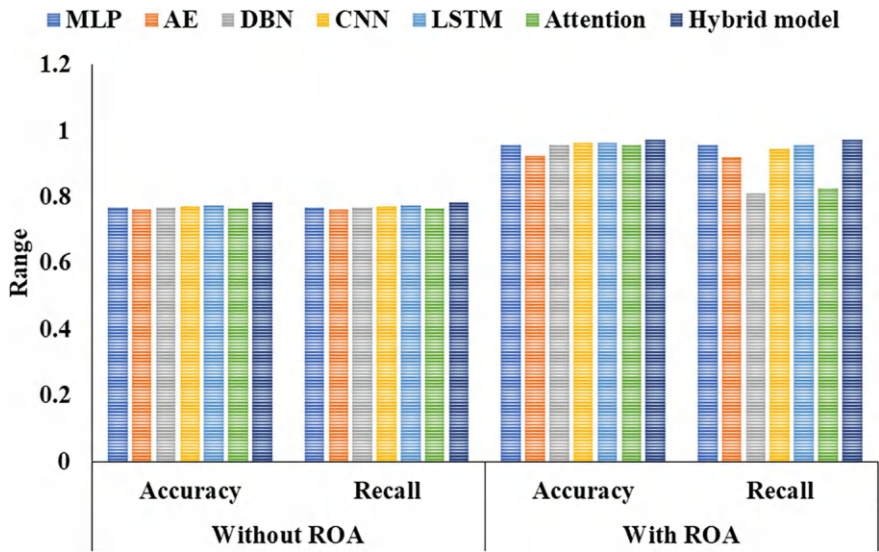


Fig. 3 Graphical description of various model

correspondingly. Then the CNN model attained the accuracy as 0.771 and recall rate as 0.771 then precision as 0.775 and F1-score as 0.769 correspondingly. Then the LSTM model attained the accuracy as 0.773 and recall rate as 0.773 then precision as 0.776 and F1-score as 0.772 correspondingly. Then the Attention model attained the accuracy as 0.766 and recall rate as 0.766 then precision as 0.770 and F1-score as 0.764 correspondingly. Then the Hybrid model attained the accuracy as 0.783 and recall rate as 0.783 then precision as 0.785 and F1-score as 0.782 correspondingly (Fig. 3).

In above Table 2 represent as Experimentation on various models with ROA. In the analysis of MLP model attained the accuracy as 0.957 and recall rate as 0.956 then precision as 0.928 and F1-score as 0.942 correspondingly. Then the AE model attained the accuracy as 0.925 and recall rate as 0.920 and F1-score as 0.967 then precision as 0.943 correspondingly. Then the DBN model attained the accuracy as 0.958 and recall rate as 0.812 then precision as 0.873 and F1-score as 0.841 correspondingly. Then the CNN model attained the accuracy as 0.963 and recall rate as 0.944 then precision as 0.802 and F1-score as 0.867 correspondingly. Then the LSTM model attained the accuracy as 0.963 and recall rate as 0.958 then precision as 0.959 and F1-score as 0.959 correspondingly. Then the Attention model attained the accuracy as 0.958 and recall rate as 0.827 then precision as 0.967 and F1-score as 0.892 correspondingly. Then the Hybrid model attained the accuracy as 0.974 and recall rate as 0.974 then precision as 0.974 and F1-score as 0.974 correspondingly (Fig. 4).

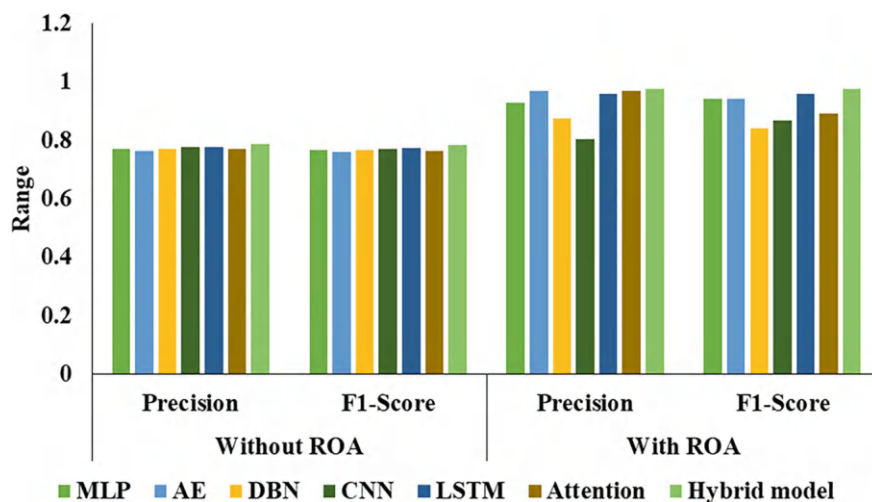


Fig. 4 Analysis of different models by considering ROA

5 Conclusion

Customer reviews posted on sites like Amazon, Facebook, Twitter, etc. are now a vital resource for any company. Product and customer review and product pages was the major focus of this investigation. The model started by checking out the effects of several preprocessing steps such data removal, text tokenization, stop word removal, excess space emotion translation into meaningful text. CLATT was used, which is based on the CNN-LSTM-Attention architecture. To maximise classification precision, ROA is used to find the best possible values for the hyper-parameters. The obtained data on the quality of customer reviews was utilised to verify the viability of the suggested strategy. Multiple sets of trials were compared, and ablation research was also performed. A sliding window approach is presented to increase the model's stability and prediction accuracy and counteract the observed declining tendency in prediction performance over time. Finally, the suggested technique realised the deep learning models, and it outperformed the competition on a variety of assessment factors. Classification models' efficacy might be drastically altered by over- or under-fitting. Future research can remedy this by utilising sampling approaches to achieve data balance.

References

1. Adak A, Pradhan B, Shukla N (2022) Sentiment analysis of customer reviews of food delivery services using deep learning and explainable artificial intelligence: systematic review. *Foods* 11(10):1500

2. Hossain MS, Rahman MF (2022) Customer sentiment analysis and prediction of insurance products reviews using machine learning approaches. *FIIB Bus Rev* 23197145221115793
3. Wu DC, Zhong S, Qiu RT, Wu J (2022) Are customer reviews just reviews? Hotel forecasting using sentiment analysis. *Tour Econ* 28(3):795–816
4. Park S, Cho J, Park K, Shin H (2021) Customer sentiment analysis with more sensibility. *Eng Appl Artif Intell* 104:104356
5. Zad S, Heidari M, Jones JH, Uzuner O (2021) A survey on concept-level sentiment analysis techniques of textual data. In: 2021 IEEE world AI IoT congress (AIIoT). IEEE, pp 0285–0291
6. Kim RY (2021) Using online reviews for customer sentiment analysis. *IEEE Eng Manage Rev* 49(4):162–168
7. Obiedat R, Qaddoura R, Ala'm AZ, Al-Qaisi L, Harfoushi O, Alrefai MA, Faris H (2022) Sentiment analysis of customers reviews using a hybrid evolutionary SVM-based approach in an imbalanced data distribution. *IEEE Access* 10:22260–22273
8. Gujjar JP, Kumar HP (2021) Sentiment analysis: textblob for decision making. *Int J Sci Res Eng Trends* 7(2):1097–1099
9. Banjar A, Ahmed Z, Daud A, Abbasi RA, Dawood H (2021) Aspect-based sentiment analysis for polarity estimation of customer reviews on twitter. *Comput Mater Continua* 67(2)
10. Chinnalagu A, Durairaj AK (2021) Context-based sentiment analysis on customer reviews using machine learning linear models. *PeerJ Comput Sci* 7:e813
11. Karthik RV, Ganapathy S (2021) A fuzzy recommendation system for predicting the customers interests using sentiment analysis and ontology in e-commerce. *Appl Soft Comput* 108:107396
12. Hossain MS, Rahman MF, Uddin MK, Hossain MK (2022) Customer sentiment analysis and prediction of halal restaurants using machine learning approaches. *J Islamic Market* (ahead-of-print)
13. Luo JM, Vu HQ, Li G, Law R (2021) Understanding service attributes of robot hotels: a sentiment analysis of customer online reviews. *Int J Hosp Manag* 98:103032
14. Ahmed HM, Javed Awan M, Khan NS, Yasin A, Faisal Shehzad HM (2021) Sentiment analysis of online food reviews using big data analytics. *Elementary Educ Online* 20(2):827–836
15. Iqbal A, Amin R, Iqbal J, Alroobaea R, Binmahfoudh A, Hussain M (2022) Sentiment analysis of consumer reviews using deep learning. *Sustainability* 14(17):10844
16. Jain PK, Quamer W, Pamula R (2023) Consumer sentiment analysis with aspect fusion and GAN-BERT aided adversarial learning. *Expert Syst* 40(4):e13247
17. Jain PK, Quamer W, Pamula R, Saravanan V (2023) SpSAN: sparse self-attentive network-based aspect-aware model for sentiment analysis. *J Ambient Intell Humaniz Comput* 14(4):3091–3108
18. Thao TTH, Thanh HT (2023) Exploring consumer opinions on vegetarian food by sentiment analysis method. *Ho Chi Minh City Open Univer J Sci-Econ Bus Adm* 13(2)
19. Punetha N, Jain G (2023) Game theory and MCDM-based unsupervised sentiment analysis of restaurant reviews. *Appl Intell* 1–22
20. Yuan Z (2022) Big data recommendation research based on travel consumer sentiment analysis. *Front Psychol* 13:857292
21. Singh C, Imam T, Wibowo S, Grandhi S (2022) A deep learning approach for sentiment analysis of COVID-19 reviews. *Appl Sci* 12(8):3709
22. Kaur G, Sharma A (2023) A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. *J Big Data* 10(1):5
23. Kantorov V, Laptev I (2014) Efficient feature extraction, encoding and classification for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Washington, DC, USA, pp 2593–2600
24. Mandelbaum A, Shalev A (2016) Word embeddings and their use in sentence classification tasks. *arXiv*, [arXiv:1610.08229](https://arxiv.org/abs/1610.08229)
25. Tabelini L, Berriel R, Paixao TM, Badue C, De Souza AF, Oliveira-Santos T (2021) Keep your eyes on the lane: real-time attention-guided lane detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). Nashville, TN, USA, pp 294–302

Potato Plant Leave Disease Detection Using TuberNet Model in Smart Agriculture



Sreekanth Rallapalli , V. Revathi, and V. Asha

Abstract Finding flaws in agricultural foodstuffs, especially potatoes, requires the use of machine vision and image processing techniques. The use of image processing and artificial intelligence in agriculture for the classification and identification of plant and fruit pests and illnesses has grown, and work in this area is still ongoing. Numerous diseases, such as early blight and late blight, have a significant impact on the quality and yield of potatoes, and manual clarification of these leaf diseases is time-consuming and labour-intensive. Effective and automated diagnosis of these diseases during the budding phase can help to improve the productivity of the potato crop despite the high degree of required. Various models have been put up in the past to identify various plant diseases. With the help of dubbed TuberNet, this study seeks to offer a practical solution for identifying the numerous potato illnesses. More specifically, the study uses the TuberNet to identify various potato leaf abnormalities utilising a strategy. To focus on the areas and improve the tactic's capacity to recognise various illnesses, a spatial-channel attention technique is employed. Transfer learning is utilised to address the capacity. The model is put to the test on the open-source, difficult dataset known as PlantVillage, which contains pictures collected under a variety of hard backdrop settings, such as varying lightning numerous colour changes in leaves. Through the executed trials, the model has verified that our method is efficient for classifying potato diseases and can successfully handle distorted tasters. So, utilising the suggested model tool, farmers can both save money and increase their crop.

Keywords Deep learning · TuberNet · Plant disease diagnosis · Transfer learning · Potato plant · Agriculture · First section

S. Rallapalli (✉)

Nitte Meenakshi Institute of Technology, Yelahanka, Bengaluru, India

e-mail: sreekanth.rallapalli@nmit.ac.in

V. Revathi · V. Asha

New Horizon College of Engineering, Bengaluru, India

1 Introduction

The largest source of food, wealth, and employment worldwide comes from agriculture. Similar to other low- to middle-income nations, the industry in India contributes 53.3% of the workforce and 18.0% of the nation's GDP [1]. Gross from agriculture has climbed from 17.6% to 20.2% of the nation's total GDP during the past three years, helping India's economic growth [2]. Although it is well established that preventative medications are ineffective for avoiding epidemic or endemic diseases, plant diseases and insect infections may have an impact on agriculture and, therefore, the production [3]. By nursing and diagnosing illnesses in crops as early as feasible, detection of agricultural diseases with proper reduce making quality losses [4].

The oomycete *Phytophthora infestans* has been responsible for late blight, which has decimated potato (*Solanum tuberosum* L.) crop production to unprecedented levels. Late blight is a disease that can affect several potato varieties grown in Colombia [5], and it is challenging to treat without applying a sizable quantity of pesticide. Finding out afflicted allows one to gauge the severity of late blight visually [6]. However, due to the need for expert (subjective) involvement, visually determining the severity of an illness is labour—very repeatable. It is possible to use unique patterns left behind by infections and bug infestations to make precise diagnoses [7]. Furthermore, farmers' or experts' diagnosis of plant illnesses could not be correct, which could result in the incorrect administration of medications, which might have a negative impact on eventually harm the ecosystem [8].

The detection problems that farmers are presently encountering have answers thanks to recent advancements in computer imaging [9]. The scientific community has developed a number of different methods for precisely recognising and classifying plant illnesses since the spots indicating infection first show as leaves. Traditional image processing is still in use today [10], necessitating the employment of manual techniques for operations like feature extraction and segmentation. Frequent computer imaging research initiatives have concentrated on using ML and DL models to improve the accuracy of illness identification since the development of AI [11].

AlexNet, GoogLeNet, and ResNet CNN architectures were used as the foundational models in a research [12] to identify illness in tomato leaf data. The model's performance may accuracy. ResNet stood up as the model that performed best across the several CNN architectures [13]. Banana leaf disease has also been detected using LeNet architecture, and the model's presentation was assessed in both colour and grayscale [14]. These criteria were used for assessments in both colour and grayscale. Reference [15] compared the architectural designs of five distinct CNN subtypes known by their numerous names, used in the study.

In the study, a successful light DL method named TuberNet is recommended. It is adept at calculating pertinent and attributes and exhibits enhanced potato leaf disease classification results with minimal computing effort. In the feature computation phase, the model uses the pixel and channel attention technique, which enhances its comprehension of crosslinks and space-wise orientation features to speed up the

problems in actual circumstances. To address overfitting, transfer learning and multi-class focal loss are used, which increases the accuracy of categorising potato leaf infected areas. The study used a collection of photos of potato crop disease gathered from a common sample PlantVillage to do extensive comparative assessments to verify the classification findings in order to show recommended model. Even in the presence of difficult external conditions including noise, distortion, imbalanced illumination, and changes in the form, colour, and positioning of infection signs, the recommended technique accurately classifies potato crop disorders.

The remnants of this essay are structured as shadows: Sect. 2 discusses similar studies; Sect. 3 discusses the suggested strategy; Sect. 4 discusses the outcomes; Sect. 5 concludes the study and elaborates on our future research goals.

2 Related Works

Potato features are used to categorise them into five courses, including Potato Late Blight (PLB), class, according to a method proposed by Mahum et al. in [16]. The proposed model is trained using the already-existing “the PlantVillage” dataset, which contains of pictures with as well as we personally collected the data for the classes of potato, including those affected by leaf potato healthy (PH). To effectively categorise the potato leaf illnesses, an efficient DenseNet model has been used, using a second transition layer in DenseNet-201. Additionally, because the training data is so strongly unbalanced, our suggested technique is strengthened by the use of the reweighted cross-entropy loss function. The regularisation power of the dense connections reduces overfitting when training minor training sets of samples of potato leaves. The suggested algorithm is a brand-new and pioneering method to identify and classify four illnesses in potato leaves, and it reports effective application. On assessed, and its accuracy was found to be 97.2%. Numerous tests have been conducted to prove that our suggested procedure is more dependable and skilled than existing models in classifying and categorising illnesses of potato leaves.

Convolution neural network (CNN) methods are used by Arshaghi et al. [17], and we also looked at five classes of potato illnesses with the labels healthy, scab, black leg, and pink rot. A potato photos was employed. We contrasted the outcomes of the categorization of potato defects using our approaches with those using transfer learning, AlexNet, GoogLeNet, VGG, and R-CNN. The findings indicate that the suggested deep learning approach has a greater accuracy than other works currently in use. In several of the classes, we receive accuracy of 100% and 99%, respectively.

Leaf diffuse reflectance spectroscopy is used by Zhou et al. [18] to identify plant diseases. After inoculation with the oomycete, leaf diffuse reflectance spectra are collected in the field using a smartphone-operated, small diffused to enable presymptomatic identification after receiving the pathogen injection and nine days prior to the appearance of visual late blight symptoms, neural network-based analysis accurately forecasts infection with a > 96% accuracy. Our work indicates the possibility for early plant disease identification utilising portable optical in conjunction analysis.

The goal of Sadiq, et al.'s [19] work was to use deep learning models for precise disease detection in potato crops, a commodity with significant worldwide economic value. The VGG16, EfficientNet B4, InceptionV3, and Inception ResNetV2 deep learning models were trained using a large dataset made up of pictures of both healthy and sick potatoes. These models performed far better than conventional visual examination techniques. The EfficientNet B4 model, which received a perfect score of 100%, displayed the best level of accuracy out of all the models tested. Following closely behind were VGG16 with a 99% accuracy rate, Inception V3 at 98%, and Inception ResNet V2 at 94%. The findings offer substantial promise for improved crop management techniques, enabling a large decrease in monetary losses caused by potato diseases. The consequences of the study support the idea that deep learning models in particular have the potential to revolutionise agricultural practises. This is important for maintaining food security given the growing difficulties caused by plant diseases.

With an online-sourced dataset, Jubaer et al.'s [20] image processing approach was suggested for the identification of potato leaf disease. Several picture preparation methods, including data augmentation, Gaussian smoothing, image normalisation, dimensionality reduction, and one hot encoding, are employed in the suggested method. As classifiers, CNN, KNN, and SVC were employed. In terms of overall accuracy, CNN provides the best answer (97%). Previous study utilising other classifiers had a number of drawbacks, and the researchers were not satisfied with the results when employing CNN. For this study, a novel hybrid model is presented that can take use of the finest CNN classifier while being considerably more trustworthy and productive.

Deep learning is used in Mishra, et al.'s [21] technique to accurately classify the two types of disorders and categorise them. We were able to achieve 97% accuracy in the first 40 CNN epochs. Additionally, a more thorough examination of other model-building strategies, such as transfer procedures, is conducted, and by comparing their accuracy scores, it is argued that the latter is the methodology best suited for this issue statement.

3 Proposed System

3.1 Data Collection

In the preliminary stage, the photographs of potato leaves were gathered from a publicly dataset, which includes a leaves illnesses, including 1000 images of "late blight", 1000 images of "early blight", and 152 images of "healthy plants". In addition to the current dataset, 1700 photos of potato leaves from diverse fields were taken using a camera device. Figure 1 and Table 1 display examples of photographs of potato leaf diseases.

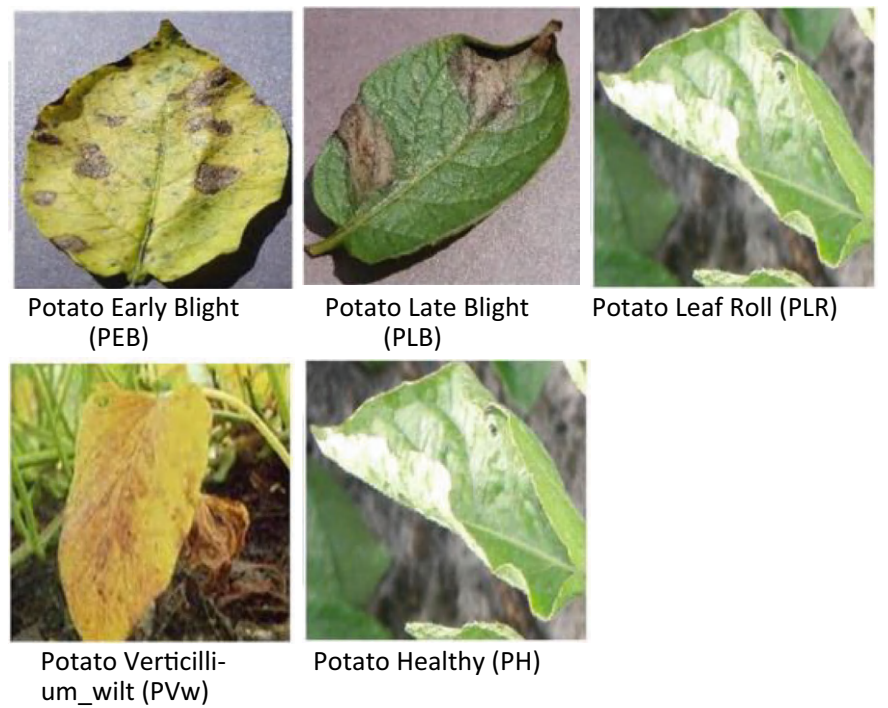


Fig. 1 Potato leaves samples from the dataset

Table 1 Class-wise circulation of the datasets

Class/Group	Sum of testing images	Sum of training images	Sum of images
Potato healthy (PH)	76	276	352
Potato late blight (PLB)	500	500	1000
Potato early blight (PEB)	500	500	1000
Potato leaf roll (PLR)	125	625	750
Potato Verticillium_wilt (PVw)	125	625	750

3.2 Pre-processing

Following data collection, pre-processing is used to get training and classification-ready samples. To begin, initial pixel values are changed in a series using picture normalisation to increase sample contrast and therefore visibility. Normalisation Eq. 1 is provided in the next section.

$$Norm = (I - Minimum) * \frac{NMaximum - NMinimum}{Maximum - Minimum} + Nminimum.$$

(1)

An image, represented by I , has a range, or absolute value, that may be anywhere from 0 to 255, as seen by the difference between the picture's supreme and least pixel values. In addition, the samples are brought into better focus with the help of contrast which are partitioned into several similar-sized sections. Pictures arranged into 8-by-8 grids produce the best results, but 1024-by-1024-pixel pictures also work. Equation (2) provides a mathematical representation of the final image following CLAHE processing.

$$CL = \{f(Z_m(x, y)) | \forall Z_m(x, y) \in Z_m\}, \quad (2)$$

where (x, y) are the organises of an image pixel and $(f(Z_m))$ is a function for transforming images according to their density.

To further reduce file size and maximise the model's efficacy, binary files are created from 0 to 255. These documents are saved as a vector array with a single dimension. It is made up of integers that represent the pixel's brightness level (0 for black and 1 for white). This is because fully linked layers in CNNs have a fixed parameter count, whereas the length of binary data varies. As a result, the example photographs must be downsized to 128×128 pixels. The resampling was done using the closest interpolation method, which preserves the original picture data. Pixel values are picked at random from those closest to the interpolated value. In comparison to the bilinear and bicubic methods, this one is easier to implement and requires less processing power. Because each pixel value from the sick portion is essential for disease detection, this method was chosen for the potato leaves. Finally, photos are saved as RGB channels for model training, which results in less data needed and a more space-efficient model.

3.3 Classification Using TuberNet

A strong collection of picture characteristics is crucial for accurate classification since it helps to distinguish across input sample types. Recall rates for methods can be improved with the use of dense DL networks by computing a more efficient collection of features. However, the placement of these CNN performances is extremely reliant on the obtainability of limitation introduced by the use of deep networks. Thus, there is always a compromise to be made between the assessment outcomes and the computing effort required. It is therefore critical to propose a framework for plant disease recognition that may demonstrate while keeping computing cost constant. To improve model performance for identifying multiple anomalies, this study presented a lightweight and computationally robust perfect. Figure 2 depicts the overall framework of the suggested paradigm.

The TuberNet framework brought about the EfficientNetV2-B4 model, which was improved upon. The EfficientNetV2 is an improved variant of the original model. To boost available resources while keeping a rate, we provide the enhanced EfficientNetV2 model. A standard ConvNet may be scaled to any resource restrictions while

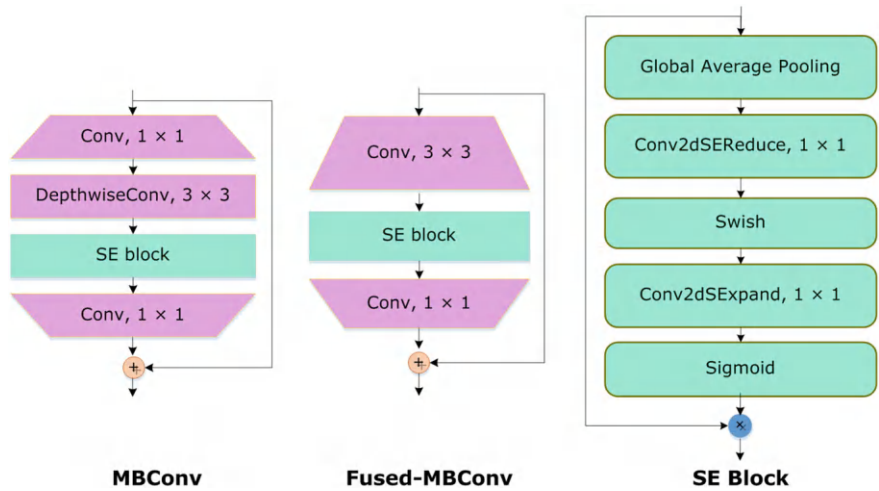


Fig. 2 Visual demo of MBConv4, SE blocks

still functioning inside the upgraded EfficientNetV2 framework, thanks to the use of a rapid and effective composite scaling mechanism. Therefore, the proposed framework provides the optimal computational cost solution regardless of the network topology, the sum of network layers, or the size of the sample feature vector. The classification process is carried out with confidence using the EfficientNetV2 method, which employs few model parameters.

Since the EfficientNetV2 layers is a lightweight framework that needs little in the form of training or inference time and has a limited sum of model parameters, it is used to classify images of plant diseases. In order to boost classification accuracy while reducing the number of features, the method employs Neural Architecture Search (NAS). In addition, mobile or server accelerators are used effectively thanks to the FusedMBConv (FMBConv) blocks in the EfficientNetV2 design. In contrast, the traditional EfficientNet method employs only depth-wise convolutions and so relies on MBConv blocks as its single building component. Even if depth-wise convolutions cut down on the number of necessary arithmetic operations, they still do not make the most of today’s hardware accelerators. To accomplish the computational improvement, the EfficientNetV2 method makes full use of both used in place of the MBConv’s expansion 1×1 convolution and depth-wise 3×3 convolution. The key motivation for substituting FMBConv for certain MBConv is to speed up the execution of the framework without sacrificing classification accuracy.

To accomplish the sickness classification task, the EfficientNetV2 perfect is utilised, together with the B4 base network. The B4 base was selected because it provides a good compromise between the classification accuracy of the model and the amount of time it takes to run. The improved EfficientNetV2 perfect is illustrated in detail in Table 1. Squeeze-and-excitation block (SEB) and swish activation are used in the upper layers of the updated EfficientNet-V2 model, whereas FMBConv blocks

are used in the lower levels. To generate such reliable classification outcomes, the used MBConv blocks keep alive an upturned residual link with the SEB. The network may then rank the affected regions of the input image based on the weights it has learned to assign to them thanks to the SEBs' use of the attention mechanism to improve key-point illustrations. Due to its inability to handle non-positive numbers, the ReLU activation function (ReLUAF) is substituted in the agenda with the swish activation function (SAF). The SAF may be determined using Eq. (3).

$$\text{SAF}(x) = x \times \text{sigmoid}(x). \quad (3)$$

A batch normalisation layer is also implemented early in a framework to reduce the overall size of the input images. As shown in Table 1, only three FMBCConv blocks were employed in the study due to the huge number of parameters they need for high standards of O (total output channels). Additionally, a layer is introduced blocks to minimise the model parameters and prevent the issue of model overfitting. In addition to the ReLUAF and dropout layers, the study also incorporated 2 additional layers, which help in extracting the more efficient collection of them in a practical format. The performance of the model is improved by randomly selecting a 30% dropout rate. Last but not least, a dense layer with five output neurons and the Softmax activation technique are utilised to create a completely automated model for classifying plant diseases.

3.3.1 Loss Function

The loss function (LF) is used to evaluate the performance of models. Networks use machine learning to automatically identify patterns in massive datasets and provide estimates based on those patterns. The chief goal of the LF is to calculate the difference between the actual and predicted scores. Throughout is iterated upon repeatedly to ensure that the minimum possible error is achieved. The function is to verify the dissimilarity between the calculated and the original values when performing classification jobs. The formula is as shadows:

$$L = \frac{1}{N} \sum_{k=1}^N \log \left(\frac{e^{s_j}}{\sum_i e^{s_k}} \right). \quad (4)$$

Here, N is showing the entire neurons in the last layer while s_k is vector.

3.3.2 Implementation Details

The approach presented in Python using the Keras and TensorFlow frameworks. Using the stochastic gradient descent (SGD) optimizer, the suggested method is conditioned on data collected from actual plant leaf samples. Overfitting may also

Table 2 Architectural particulars of the proposed outline

Block/Layer	Size	Channel
BatchNorm	224×224	3
ConvL (3 × 3)	112×112	40
2 × FMBCConv1	112×112	16
3 × FMBCConv4	56×56	40
2 × FMBCConv4	28×28	56
5 × MB_Conv4	14×14	112
7 × MB_Conv6	14×14	136
12 × MB_Conv6	7×7	232
Conv2d (1 × 1)	7×7	232
GlobalAveragePooling layer	1,792	
Denselayer	128	
Dropout	128	
Denselayer	64	
Dropout	64	
FC		
Classification		

be avoided in model training with the help of the early stopping regularisation strategy. After 20 epochs of training with the provided method, we saw no appreciable improvement in our model’s performance. We also set the batch to 12, and the learning rate to 0.0001, among other settings. Table 2 provides an inclusive discussion of the model parameters. The proposed method has been successfully deployed on a 16GB RAM, Nvidia GTX1070 GPU-based machine. The best possible score was achieved in the 20th epoch, 0.0023, demonstrating the well-trained nature of the proposed model. Additionally, we demonstrated a training accuracy of 99.01%. Table 2 Details of network hyperparameters.

Parameters	Value
Total used epochs	20
Used value of learning rate	0.0001
Designated batch size	12
Optimizer	SGD

4 Results and Discussion

4.1 Evaluation Metrics

After the training and testing steps are complete, the model's performance in object identification must be evaluated using specified assessment criteria. Researchers used a variability of measures for assessment, including as:

Precision (PR), recall (RE), sensitivity (SE), specificity (SP), accuracy (AC), F1-score, and confusion matrix (CM). Metrics like as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) may be determined by smearing the perfect to a dataset and tallying the results. Cases of disease that have been correctly identified by the model are denoted by TP, whereas cases of non-disease that have been incorrectly identified as disease are denoted by FP. Diseases are ones that were missed throughout the diagnostic process. A true negatives (TN) is an event that played out exactly as forecast. Since the F1-score measures, it is better applicable to datasets with asymmetry. The total performance of each model is evaluated by computing its accuracy, precision, sensitivity, specificity, and F1-score using Eqs. (5)–(9).

$$PR = \frac{TP}{TP + FP} \quad (5)$$

$$RE \text{ and } SE = \frac{TP}{TN + FN} \quad (6)$$

$$SP = \frac{TN}{TN + FP} \quad (7)$$

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$F1 - \text{score} = \frac{2(TP)}{2(TP) + FP + FN} \quad (9)$$

When evaluating a model, metrics are essential to consider. These measures indicate how events are detected, how many correct predictions were made, and how well precision and recall align with each other. These criteria for measuring a model's accuracy and consistency are as vital in deep learning as they are in traditional machine learning.

4.2 Validation Analysis of Proposed Model

Tables 3 and 4 provide the comparative investigation of projected perfect with existing techniques in terms of numerous metrics.

Table 3 above presents the analysis of various models using a 60–40% data split. In the analysis, the LeNet model achieved a Precision (PR) rate of 92.7, Recall (RE) rate of 91.6, Sensitivity (SE) of 92.6, Specificity (SP) of 90.1, Accuracy (AC) of 87.3, and an F1-score of 90.5, respectively. Then the ResNet model attained the PR rate as 93.6 and RE rate as 92.0 and SP value as 93.0 and AC range of 90.3 and AC range of 88.2 and F1-score as 91.7 correspondingly. Then the AlexNet model attained the PR rate as 94.3 and RE rate as 94.5 and SP value as 94.5 and then SE as 91.2 and AC range of 89.5 and then F1-score as 92.3 correspondingly. Then the VGGNet model attained the PR rate as 95.5 and RE rate as 95.6 and SP value as 94.6 and then SE as 93.4 and AC range of 90.6 and then F1-score as 93.6 correspondingly. Then the GoogLeNet model attained the PR rate as 95.6 and RE rate as 95.8 and SP value as 95.8 and then SE as 95.5 and AC range of 91.5 and then F1-score as 94.8 correspondingly. Then the TuberNet model attained the PR rate as 96.5 and RE rate as 96.3 and SP value as 96.3 and then SE as 97.4 and AC range of 92.5 and then F1-score as 95.4 correspondingly (Fig. 3).

Table 4 characterises that the comparative investigation of projected perfect on 80–20%. In the analysis, the LeNet model achieved a Precision (PR) rate of 96.2, Recall (RE) rate of 96.6, Sensitivity (SE) of 96.6, Specificity (SP) of 96.1, Accuracy (AC)

Table 3 Analysis of various models on 60%-40%

Models	PR (%)	RE (%)	SE (%)	SP (%)	AC (%)	F1-Score (%)
LeNet	92.7	91.6	92.6	90.1	87.3	90.5
ResNet	93.6	92.0	93.0	90.3	88.2	91.7
AlexNet	94.3	94.5	94.5	91.2	89.5	92.3
VGGNet	95.5	95.6	94.6	93.4	90.6	93.6
GoogLeNet	95.6	95.8	95.8	95.5	91.5	94.8
TuberNet	96.5	96.3	96.3	97.4	92.5	95.4

Table 4 Comparative analysis of proposed model on 80–20%

Models	PR (%)	RE (%)	SE (%)	SP (%)	AC (%)	F1-Score (%)
LeNet	96.2	96.6	96.6	96.1	96.3	96.4
ResNet	96.6	96.8	96.8	96.2	96.5	96.7
AlexNet	96.7	97.1	97.1	96.3	96.4	96.9
VGGNet	97.4	97.7	97.7	97.3	97.6	97.5
GoogLeNet	97.7	97.9	98.0	97.5	97.8	97.8
TuberNet	99.5	99.3	99.3	99.4	99.5	99.4

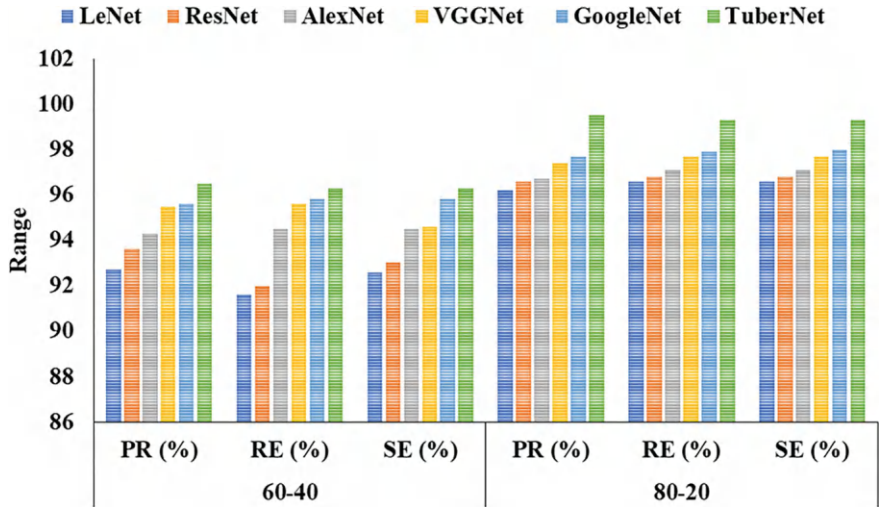


Fig. 3 Graphical comparison of various models

of 96.3, and an F1-score of 96.4, respectively. Then the ResNet model accomplished the PR rate as 96.6 and RE rate as 96.8 and then SE as 96.8 and AC range of 96.2 and AC range of 96.5 and then F1-score as 96.7 correspondingly. Then the AlexNet model accomplished the PR rate as 96.7 and RE rate as 97.1 and then SE as 97.1 and SP value as 96.3 and AC range of 96.4 and then F1-score as 96.9 correspondingly. Then the VGGNet perfect accomplished the PR rate as 97.4 and RE rate as 97.7 and then SE as 97.7 F1-score as 97.3 and SP value as 97.6 and then F1-score as 97.5 correspondingly. Then the GoogLeNet model attained the PR rate as 97.7 and RE rate as 97.9 and then SE as 98.0 and SP range of 97.5 and AC range of 97.8 and then F1-score as 97.8 congruently. Then the TuberNet model attained the PR rate as 99.5 and RE rate as 99.3 and then SE as 99.3 and SP value as 99.4 and AC range of 99.5 and then F1-score as 99.4 correspondingly (Fig. 4).

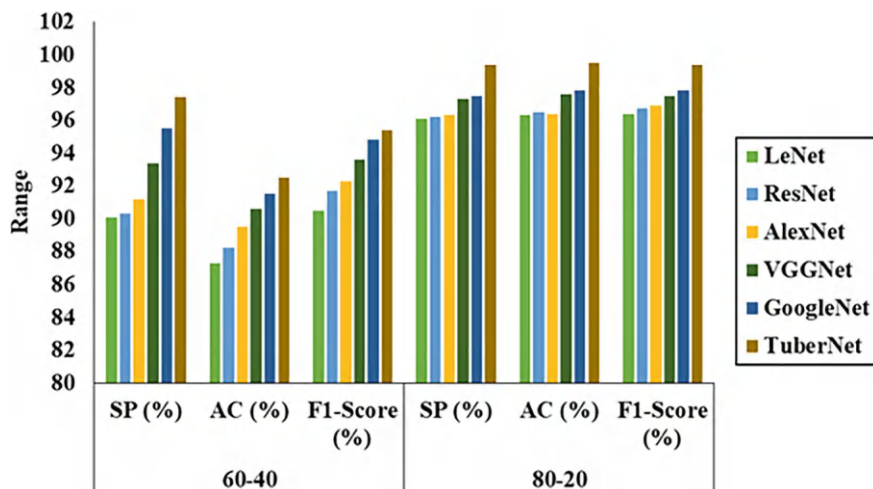


Fig. 4 Comparison of various models for different metrics

5 Conclusion

As digitization advances across many sectors, it is crucial that agriculture also utilises it in order to boost growth and yield security. With this aim in mind, a technique was created to distinguish between afflicted and unaffected potato leaves. Farmers lose money and crop productivity due to diseases. Leaf spots caused by early and late blight are common on potatoes. It is believed that these illnesses account for the vast majority of the estimated yield loss in potatoes. We classified photos of potato leaves as either healthy, late, or early blight afflicted. In this study, we employ a TuberNet-based approach to class recognition. The leaves of several potato plant kinds are classified using a DL technique called TuberNet. In order to enhance the EfficientNet-v2 approach, the AM technique and additional structure were added to the research. The proposed tactic uses an end-to-end learning process to reliably extract high-level signals from regions with an infection and correlate them with related communities. By emphasising salient traits, such as diseased spots on plant leaves, the AM approach improves recall of the recommended treatment. The study used extensive testing on a multifaceted data taster named tVillage to prove the efficacy of our methodology. The acquired performance scores demonstrated the effectiveness of our algorithm in detecting potato diseases in images with significant distortion. The projected structure is accurate to the tune of 99.5%. The level of accuracy, however, still needs improvement. Networks, a kind of artificial neural networks, may be utilised to build upon previous efforts. Today, many research using images use CNN methods in an effort to get more precise outcomes. Accuracy in CNN designs may be enhanced by the use of several components, including convolutional layers, batch normalisations, fully linked layers, and activation functions.

References

1. Tarik MI, Akter S, Al Mamun A, Sattar A (2021) Potato disease detection using machine learning. In: 2021 third international conference on intelligent communication technologies and virtual mobile networks (ICICV). IEEE, pp 800–803
2. Lee TY, Lin IA, Yu JY, Yang JM, Chang YC (2021) High efficiency disease detection for potato leaf with convolutional neural network. *SN Comput Sci* 2(4):297
3. Eser SERT (2021) A deep learning based approach for the detection of diseases in pepper and potato leaves. *Anadolu Tarım Bilimleri Dergisi* 36(2):167–178
4. Sharma S, Anand V, Singh S (2021) Classification of diseased potato leaves using machine learning. In: 2021 10th IEEE international conference on communication systems and network technologies (CSNT). IEEE, pp 554–559
5. Baswaraju S, Maheswari VU, Chennam KK, Thirumalraj A, Kantipudi MP, Aluvalu R (2023) Future food production prediction using AROA based hybrid deep learning model in agri-sector. *Human-Centric Intell Syst* 1–16
6. Appalanaidu MV, Kumaravelan G (2021) Plant leaf disease detection and classification using machine learning approaches: a review. *Innovations in computer science and engineering: proceedings of 8th ICICSE*, pp 515–525
7. Asfaw TA (2023) Deep learning hyperparameter's impact on potato disease detection. *Sci Temper* 14(03):582–590
8. Wani JA, Sharma S, Muzamil M, Ahmed S, Sharma S, Singh S (2022) Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: methodologies, applications, and challenges. *Arch Comput Methods Eng* 29(1):641–677
9. Rashid J, Khan I, Ali G, Almotiri SH, AlGhamdi MA, Masood K (2021) Multi-level deep learning model for potato leaf disease recognition. *Electronics* 10(17):2064
10. Dayang P, Meli ASK (2021) Evaluation of image segmentation algorithms for plant disease detection. *Int J Image Graph Signal Process (IJIGSP)* 13(5):14–26
11. Ranjana P, Reddy JPK, Manoj JB, Sathvika K (2022) Plant leaf disease detection using mask R-CNN. In: *Ambient communications and computer systems: proceedings of RACCCS 2021*. Springer Nature Singapore, Singapore, pp 303–314
12. Zhao Y, Chen Z, Gao X, Song W, Xiong Q, Hu J, Zhang Z (2021) Plant disease detection using generated leaves based on DoubleGAN. *IEEE/ACM Trans Comput Biol Bioinf* 19(3):1817–1826
13. Lakshmanarao A, Babu MR, Kiran TSR (2021) Plant disease prediction and classification using deep learning ConvNets. In: 2021 international conference on artificial intelligence and machine vision (AIMV). IEEE, pp 1–6
14. Bangari S, Rachana P, Gupta N, Sudi PS, Baniya KK (2022) A survey on disease detection of a potato leaf using CNN. In: 2022 second international conference on artificial intelligence and smart energy (ICAIS). IEEE, pp 144–149
15. Singh A, Kaur H (2021) Potato plant leaves disease detection and classification using machine learning methodologies. *IOP Conf Ser: Mater Sci Eng* 1022(1):012121. IOP Publishing
16. Mahum R, Munir H, Mughal ZUN, Awais M, Sher Khan F, Saqlain M et al (2023) A novel framework for potato leaf disease detection using an efficient deep learning model. *Hum Ecol Risk Assess Int J* 29(2):303–326
17. Arshaghi A, Ashourian M, Ghabeli L (2023) Potato diseases detection and classification using deep learning methods. *Multimedia Tools Appl* 82(4):5725–5742
18. Zhou C, Bucklew VG, Edwards PS, Zhang C, Yang J, Ryan PJ et al (2023) Portable diffuse reflectance spectroscopy of potato leaves for pre-symptomatic detection of late blight disease. *Appl Spectrosc* 77(5):491–499
19. Sadiq S, Malik KR, Ali W, Iqbal MM (2023) Deep learning-based disease identification and classification in potato leaves. *J Comput Biomed Inform* 5(01):13–25

20. Jubaer MA, Hasan MN, Mustavi M, Shahriar MT, Ahmed T (2023) Potato leaf disease detection using image processing
21. Mishra H, Pandey V, Kothari D, Gharat M, Salvi L (2023) Enhanced disease detection for potato crop using CNN with transfer learning

Detection of Alzheimers Using Imaging and Machine Learning



Somu Geetha Sravya , Kamsala Tharun , B. R. Sandhya , Tresa Biju , and Gopi Sagar 

Abstract This research introduces an innovative approach that combines machine learning and graph theory to aid in the diagnosis and monitoring of Alzheimer's disease (AD). Alzheimer's, a neurodegenerative disorder affecting cognitive function, requires early diagnosis for effective intervention. Resting-state fMRI is crucial in identifying brain connectivity changes associated with AD. Graph theory quantifies brain network integration and segregation, shedding light on the disease's pathology. By merging fMRI data and graph theory, this approach allows researchers to analyze brain connectivity at a macroscopic level, improving diagnostic accuracy. The study has two main objectives: distinguishing cognitively normal individuals from those with AD using machine learning and predicting disease progression in those with Mild Cognitive Impairment (MCI). This work showcases the potential of integrating machine learning and graph theory to enhance AD diagnosis and monitoring, offering the promise of earlier interventions and improved quality of life for AD patients. It contributes to the intersection of medical imaging, machine learning, and neuroscience research, advancing strategies against neurodegenerative disorder. The convolutional neural network (CNN) stands out as a superior model for Alzheimer's disease prediction due to its adeptness in analyzing complex image data from brain scans. CNN excels in identifying intricate patterns and subtle abnormalities crucial in diagnosing Alzheimer's. Leveraging its deep learning architecture,

S. G. Sravya (✉) · K. Tharun · B. R. Sandhya · T. Biju · G. Sagar
Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru,
Karnataka, India
e-mail: geethasravya.somu11@gmail.com

K. Tharun
e-mail: tharunkamsala04@gmail.com

B. R. Sandhya
e-mail: sandhya.br@nmit.ac.in

T. Biju
e-mail: tresasbleathers@gmail.com

G. Sagar
e-mail: gopisagar323@gmail.com

CNN efficiently processes and extracts intricate features from neuroimaging data, enabling accurate classification and early detection of Alzheimer's. Its capacity to recognize minute structural changes in the brain through imaging techniques such as MRI and PET scans makes it a leading model in predicting and diagnosing this neurodegenerative condition, offering promising prospects in advancing early intervention and treatment.

Keywords fMRI-Functional magnetic resonance imaging · A.D.-Alzheimer's disease · sMRI-Structural magnetic resonance imaging · M.V.P.A.-Multi-voxel pattern analysis · F.S.L.-F.M.R.I.B. software library

1 Introduction

First and foremost, this recordkeeping is to delve into the quest for an effective solution to detect Alzheimer's disease, a condition for which there is currently no cure but whose progression can be delayed through early detection. The methodology we delve into combines an assortment of state-of-the-art methods, including brain imaging and the use of graph theory. In addition, we harness the capabilities of machine learning (ML) techniques to automate this diagnostic process and evaluate its effectiveness [1, 2]. Alzheimer's disease represents one of the most daunting challenges in contemporary healthcare. It's an irreversible, degenerative brain disease that eventually impairs one's ability to remember things, think clearly, and eventually do even the most basic everyday tasks. The ramifications of Alzheimer's are severe and far-reaching, from the profound memory loss it inflicts to the physical impact on the brain itself. The urgency of early detection is paramount in the face of this disease, as it offers the only means by which its relentless progression can be mitigated [3], even if a complete cure remains elusive.

The amalgamation of advanced techniques is central to our quest for effective Alzheimer's detection. Brain imaging, especially through fMRI, holds a crucial position in our strategy. It facilitates the gathering of vital data pertaining to brain function and connectivity, providing a glimpse into the complex network of neural pathways impacted by the disease. This data serves as the foundation upon which our diagnostic methodology is built. A critical element in our strategy is the utilization of graph theory as a potent tool for the diagnosis of Alzheimer's disease. Graph theory provides a systematic framework for analyzing the connectivity patterns of various brain regions. Through its quantitative measures of integration and segregation within the brain network, we comprehend the underlying pathophysiology of the illness more thoroughly.

This allows us to discern the subtle yet significant differences in brain connectivity that result from Alzheimer's, contributing to more accurate diagnosis and monitoring. Machine learning emerges as a game-changer in this endeavor. We leverage ML techniques to automate the diagnostic process and rigorously assess its efficiency. Our methodology comprises six key stages, each meticulously designed to extract

valuable insights from fMRI data. These stages encompass preprocessing of fMRI data, modeling fMRI voxel time series, feature extraction, feature selection, and classification. In the classification stage, we deploy classical and improved variations of the random forests algorithm and support vector machines [4], two powerful ML approaches.

Our work culminates in the interpretation of random forest trees into physically meaningful rules, which further advances our knowledge for diagnostic procedure. We rigorously evaluate our technique using a dataset that includes 41 patients in order to confirm its efficacy. Our method's findings demonstrate its amazing validity in Alzheimer's disease diagnosis and tracking. The accuracy rates of 94% for diagnosis and 97% to 99% for monitoring underscore the potential of this integrated approach to revolutionize our fight against this debilitating condition. In the end, our documentation embarks on a journey to confront Alzheimer's disease head-on by amalgamating cutting-edge techniques such as fMRI, graph theory, and machine learning. Through this multi-faceted approach, we endeavor not only to diagnose Alzheimer's with unprecedented accuracy but also to offer a means of monitoring its progression [5], thereby arming healthcare professionals with powerful tools for early intervention. While Alzheimer's remains a formidable adversary, the promise of early detection and intervention holds the potential to significantly enhance the lives of individuals affected by this relentless condition. Our work stands as a testament to the boundless possibilities at the intersection of technology, neuroscience, and healthcare, offering hope for a brighter future in the battle against Alzheimer's disease.

The main contributions of this paper are as follows:

- **Innovative Integration of Machine Learning and Graph Theory:** This research pioneers the integration of machine learning techniques, specifically the convolutional neural network (CNN), with graph theory to analyze resting-state fMRI data. This unique approach provides a macroscopic understanding of brain connectivity changes associated with Alzheimer's disease (AD), contributing to the development of more accurate diagnostic tools.
- **Enhanced Diagnostic Accuracy:** By leveraging the capabilities of CNN in processing complex image data from brain scans, the study significantly improves diagnostic accuracy for AD. The model excels in identifying intricate patterns and subtle abnormalities in neuroimaging data, offering a more refined and reliable means of distinguishing cognitively normal individuals from those with AD.
- **Dual-Objective Focus on Diagnosis and Prediction:** The research addresses two crucial aspects of AD—distinguishing between cognitively normal individuals and those with AD through machine learning and predicting disease progression in individuals with Mild Cognitive Impairment (MCI). This dual-objective framework contributes to a comprehensive understanding of the disease spectrum, aiding both early diagnosis and proactive management.
- **Advancement in Early Intervention Strategies:** The findings underscore the potential for earlier interventions and improved quality of life for AD patients. The CNN's ability to recognize minute structural changes in the brain through various

imaging techniques positions it as a leading model, offering promising prospects for advancing early intervention and treatment strategies against Alzheimer's disease.

2 Background and Motivation

Alzheimer's disease prediction has emerged as a critical area in neurology and health-care due to its devastating impact on individuals and society. The motivation behind predicting Alzheimer's lies in the urgency to detect and intervene in this progressive neurodegenerative condition before irreversible damage occurs. With an aging global population, Alzheimer's poses a significant public health challenge. Its insidious onset, typically with subtle symptoms, often leads to diagnosis at later stages, when interventions may offer limited efficacy. Therefore, the quest to develop predictive models stems from the pressing need to identify the disease in its earliest stages, enabling timely therapeutic strategies and lifestyle interventions.

Understanding the brain changes associated with Alzheimer's disease has been pivotal in driving predictive model development. Neuroimaging techniques, such as magnetic resonance imaging (MRI) and positron emission tomography (PET), have unveiled structural alterations and biomarkers associated with the disease. These imaging methods allow for the visualization of brain atrophy, the accumulation of beta-amyloid plaques, and neurofibrillary tangles, characteristic features of Alzheimer's. Harnessing these imaging insights has motivated researchers to employ advanced computational models to analyze and interpret vast amounts of data, seeking patterns and markers that could signify the disease's presence long before clinical symptoms manifest. The urgency to predict Alzheimer's disease also aligns with the potential for early interventions. Detecting the disease in its preclinical or prodromal stages opens the door for preventive measures and treatments that could significantly alter the disease trajectory [6]. Predictive models aim to distinguish individuals at higher risk, facilitating targeted interventions like lifestyle modifications, cognitive training, and potential future pharmaceutical therapies. Such proactive measures could slow disease progression, delay symptom onset, or even prevent the onset of debilitating cognitive decline, thereby improving the quality of life for individuals at risk.

Moreover, the economic burden of Alzheimer's disease provides additional motivation for predictive research. The costs associated with caring for individuals with Alzheimer's, from healthcare expenses to caregiving and lost productivity, are substantial. Early prediction and intervention could potentially alleviate this economic strain by reducing the need for intensive long-term care and medical intervention. The motivation behind Alzheimer's disease prediction is multi-faceted. It stems from the urgent need to address the increasing prevalence of the disease, the potential for early interventions to alter its course, and the desire to reduce the economic burden on healthcare systems and society. By leveraging advanced computational models, such as machine learning and specifically convolutional neural

networks (CNNs), the quest to predict and ultimately prevent the devastating impacts of Alzheimer’s remains a crucial and promising area of research in healthcare and neurology.

3 System Model

The proposed idea for the Alzheimer’s disease prediction is depicted in the form of a diagram below.

3.1 Model Flow Diagram

The diagram presented as Fig. 1 illustrates a workflow for predicting Alzheimer’s disease using a convolutional neural network (CNN) model. The process begins with an image dataset, which is assumed to contain brain scans relevant to Alzheimer’s disease. This dataset undergoes a creation process to structure it appropriately for analysis. Once structured, the dataset is loaded and divided into two subsets: a training set and a testing set. The training set is used to teach the CNN model to recognize patterns associated with Alzheimer’s disease, while the testing set is reserved for evaluating the model’s performance.

Preprocessing steps are applied to the data to make it suitable for the CNN model, which may include normalization, augmentation, or feature extraction. The CNN

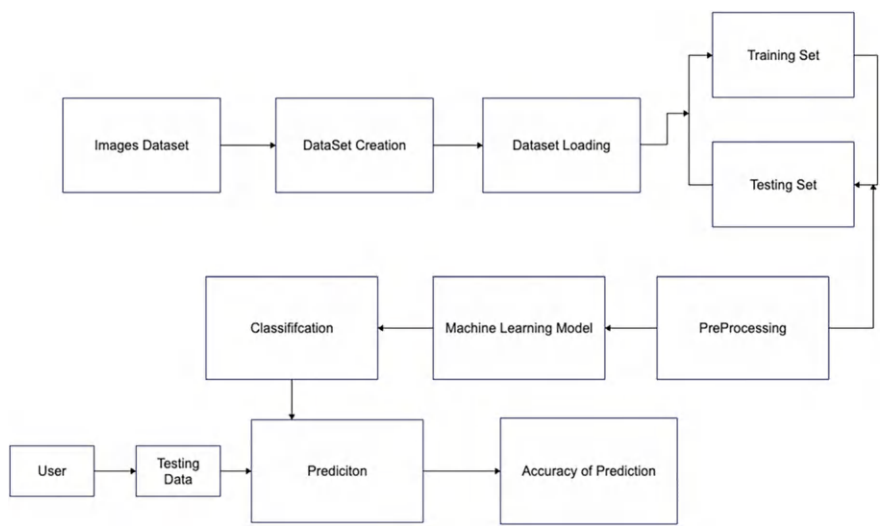


Fig. 1 System model flow diagram

model then learns to classify the data during the training phase. Classification is the step where the model makes a decision on whether the input data (e.g., a brain scan) indicates the presence of Alzheimer's disease.

The user provides testing data, which is new data not seen by the model during training, to make predictions. These predictions are then evaluated for accuracy, which provides a quantitative measure of how well the CNN model performs in predicting Alzheimer's disease from brain scans. The accuracy of prediction is a crucial metric that determines the reliability of the model in a real-world clinical setting. The diagram below shows the basic system model and various sub-systems.

3.2 System Model Description

(1) *Dataset Creation for Alzheimer's Disease Prediction:*

- Data Acquisition: Gather neuroimaging data like MRI and PET scans from diverse sources, encompassing individuals diagnosed with Alzheimer's and healthy controls.
- Labeling and Ground Truth: Annotate the dataset, indicating which scans correspond to Alzheimer's patients and which belong to healthy subjects, creating the labeled ground truth for model training.
- Dataset Stratification: Segment the dataset into training, validation, and testing sets, ensuring a representative distribution of Alzheimer's and healthy samples in each subset.

(2) *Preprocessing Neuroimaging Data for Alzheimer's Prediction:*

- Image Standardization: Normalize and standardize the neuroimages to a consistent format, resolution, and quality to remove variances that are not clinically relevant.
- Feature Extraction from Brain Scans: Extract pertinent features from the neuroimages such as brain volume, cortical thickness, and regions of interest using image processing techniques to focus on areas affected by Alzheimer's.
- Data Augmentation: Apply techniques like rotation, flipping, and adding noise to augment the dataset, enhancing model robustness and preventing overfitting.

(3) *Classification Methodologies for Alzheimer's Disease Prediction:*

- Feature Selection: Identify and prioritize the most relevant biomarkers and neuroimaging features for accurate Alzheimer's prediction.
- Model Selection: Choose appropriate classification algorithms (like CNN, support vector machines, or deep learning models) capable of learning complex patterns from neuroimaging data.
- Training and Validation: Train the model using the Alzheimer's dataset, validate its performance using a separate dataset, and fine-tune model parameters to achieve high accuracy.

(4) *Convolutional Neural Network (CNN) Model for Alzheimer's Prediction:*

- **CNN Architecture Design:** Develop a specialized CNN architecture tailored for processing neuroimaging data, integrating convolutional, pooling, and fully connected layers.
 - **Training the CNN:** Use the labeled Alzheimer's dataset to train the CNN model, allowing it to learn distinctive features indicative of Alzheimer's pathology.
 - **Regularization and Optimization:** Apply regularization techniques like dropout and batch normalization to prevent overfitting and ensure the model's stability and generalization.
- (5) *Alzheimer's Disease Prediction and Clinical Relevance:*
- **Testing and Evaluation:** Assess the CNN model's performance using an unseen testing dataset to gauge its accuracy in predicting Alzheimer's presence or progression.
 - **Prediction Output:** Generate predictions indicating the probability or presence of Alzheimer's disease in new, unseen neuroimaging data based on the trained CNN model.
 - **Clinical Interpretation and Reporting:** Translate model predictions into actionable insights for clinicians, providing early diagnostic indicators and potential intervention strategies based on the likelihood of Alzheimer's disease.

This systematic approach to Alzheimer's disease prediction, from data collection and preprocessing to specialized CNN modeling and accurate prediction, holds promise in enabling early diagnosis and intervention, potentially revolutionizing the management and treatment of Alzheimer's disease.

4 Literature Survey

The journey to comprehend and tackle Alzheimer's disease (AD) has made notable progress through the incorporation of deep learning methodologies. This fusion of technology and medical science aims to enhance the identification and understanding of a disease that, as of now, remains incurable but may be delayed through early intervention. In this endeavor, combining several forms of data—such as test results, clinical information, and MRI scans—is an essential strategy. Together, these provide a more holistic understanding of Alzheimer's disease (AD). This holistic approach, underpinned by deep learning, enables precise predictions based on imaging features and clinical data, potentially revolutionizing the diagnosis and management of this devastating condition. However, as we explore these promising developments, it is essential to acknowledge their limitations and potential challenges. The foundation of these innovations lies in the amalgamation of various data types, including clinical information, test results, and MRI scans. This multi-dimensional approach enables a more comprehensive understanding of Alzheimer's disease, as it integrates disparate pieces of information into a cohesive whole. By employing deep learning techniques, it becomes possible to make accurate predictions about the disease based on these

diverse data sources [5]. This integrated approach holds the potential to enhance our ability to diagnose and manage AD effectively. Deep learning, a branch of artificial intelligence, is instrumental in these advancements. It is now used to navigate the difficulties associated with Alzheimer's disease. It has shown its capabilities in many domains, such as computer vision and natural language processing. Convolutional neural networks (CNNs) are useful tools for medical picture interpretation since they have solved issues such as overfitting and the vanishing gradient problem [7]. These technologies have enormous promise, but they also have a unique set of drawbacks and difficulties. One fundamental constraint lies in the reliance on MRI scans as a primary data source. While MRI is a powerful tool, its accessibility varies across different settings, and in some cases, it may not be readily available. Moreover, the preprocessing steps involved in preparing neuroimaging data for analysis can be time-consuming and introduce the potential for errors, which may impact the quality of the analysis. Deep learning methods, while powerful, have their own set of demands. They are computationally intensive, often requiring substantial resources in terms of time and hardware. Additionally, they depend heavily on the availability of large and diverse datasets to perform optimally [8]. However, such datasets may not always be easily available for use in Alzheimer's disease research. Additionally, overfitting is a common mistake made by deep learning models, where they excel on the training data but fail to adapt to new, unseen data, which could potentially restrict their applicability in real-world scenarios. The diagnosis of Alzheimer's disease currently depends on clinical tests that are slow and prone to errors. Deep learning and machine learning techniques, especially those using CNNs, have shown potential in improving AD detection through medical imaging data analysis. However, this approach relies on the availability of structural MRI (sMRI) data, which may not be widely accessible. Preprocessing steps involved in preparing this data can be resource-intensive and may add more complexity to the analysis [6]. One of the biggest obstacles to classifying AD using deep learning is the limited availability of data, especially in the context of AD. Small sample sizes can hamper the training of deep learning models, and selecting relevant features for classification remains complex due to our incomplete understanding of AD pathology and associated features. Furthermore, enhancing neural network structures for accurate classification is capable of processing lengthy and computationally demanding tasks. Despite these challenges, graph theory combined with machine learning has shown the potential to classify AD patients from healthy subjects with remarkable accuracy, reaching 100%. By using graph theoretical measures derived from functional brain networks and rs-fMRI data, this approach offers a non-invasive and automated method for diagnosing AD. As the prevalence of AD continues to increase, leading to higher healthcare costs, early detection becomes essential for effective treatment and prevention. Neuroimaging techniques, such as MRI and positron emission tomography (PET), play a crucial part in visualizing the brain abnormalities associated with AD. Deep learning, with its ability to analyze complex data, provides a viable path for the early use of neuroimaging data for AD categorization and detection. A number of difficulties still exist, including the need for greater and more diverse datasets, rigorous optimization of neural network architecture, and the reduction

of overfitting in small datasets [9]. In the quest to enhance Alzheimer's diagnosis and monitoring, deep learning methods have advanced significantly, taking use of the abundance of information offered by medical imaging data. Nevertheless, the limitations and challenges, such as data availability, preprocessing complexity, and overfitting, underscore the need for continued research and refinement. As we stand at the intersection of technology and healthcare, the pursuit of more accurate and accessible AD diagnosis remains an imperative goal. The promise of early detection and improved patient outcomes remains a powerful driving force in the ongoing battle against this relentless neurodegenerative disease [2].

5 Research Gaps

The utilization of deep learning methodologies for Alzheimer's disease (AD) diagnosis and classification presents both potential and challenges in the quest for more accurate and accessible detection. One prominent challenge lies in the heavy reliance on MRI scans, which, unfortunately, may not be readily accessible in all situations, limiting the applicability of this approach. Furthermore, the preprocessing steps involved in preparing neuroimaging data can be time-consuming and introduce the potential for errors, thus demanding careful attention to data quality. The caliber and quantity of training data have a major impact on how effective deep learning models are. This can significantly differ across datasets, thereby influencing the sturdiness and applicability of the models. Overfitting is a recurring problem in this instance, where models may perform well on training data but struggle to generalize to new, unseen data, necessitating strategies to mitigate this challenge. Class imbalance and the presence of correlated features in brain structure add complexity to the task of accurate AD classification, making it essential to address these issues for meaningful results [10]. Another formidable hurdle is the selection of relevant features for AD classification, given the current uncertainty surrounding the underlying pathology of AD. This lack of a clear understanding of AD's pathological mechanisms poses a significant challenge in feature selection and model development.

Moreover, the limitation of small sample sizes in many studies targeting AD classification restricts the generalizability of findings, raising questions about the broader applicability of the developed models. In light of the urgency of early AD detection, the absence of easy and non-invasive methods for early detection remains a notable gap in current diagnostic approaches. Existing diagnostic tools for AD are known to lack the required accuracy, emphasizing the need for creative answers that can enhance detection sensitivity and specificity. In conclusion, deep learning approaches each offer a unique set of challenges even if they provide a favorable route for the diagnosis and categorization of Alzheimer's disease. The reliance on MRI scans and associated accessibility issues, the complexities of preprocessing, and the critical role of data quality all influence the effectiveness of deep learning models. Overfitting, class imbalance, feature selection dilemmas, and limited sample sizes further underscore the intricacies of this task. As we strive to improve early

AD detection, the development of reliable and accessible diagnostic tools remains a paramount goal, calling for continued research and innovation in this crucial area of healthcare.

Advanced imaging techniques such as MRI and PET scans play a crucial role in the detection of Alzheimer's disease but come with high costs. The acquisition, maintenance of the equipment, and the necessity for skilled technicians for operation and interpretation limit their widespread application, particularly in regions with fewer resources. The uneven global distribution of such imaging infrastructure impedes the collection of diverse data sets, which is essential for the generalizability of machine learning models. Furthermore, the quality of input data is vital for the accuracy of these models. Artifacts, patient movement, and technical inconsistencies during imaging can adversely affect data integrity, leading to models that perform poorly in practical scenarios. Interpretability of machine learning models, particularly deep learning models like CNNs, also poses a challenge. Their "black box" nature can hinder clinical adoption due to the difficulty in understanding how these models arrive at their predictions. Overfitting is another critical issue; models that perform well on training data but fail to generalize can be particularly problematic in the medical field, where the diversity of data and accuracy of predictions are paramount. Data privacy is another significant concern; protecting the sensitive medical information used in machine learning models is essential and requires adherence to strict regulations such as HIPAA and GDPR. Finally, the resource requirements for developing and maintaining machine learning models are considerable. They require not only computational power, like GPUs for training, but also storage for large data sets and a team of interdisciplinary experts to build, validate, and interpret the models. Addressing these challenges is pivotal for the progress of Alzheimer's disease detection through machine learning and imaging techniques. Potential solutions include making imaging technologies more cost-efficient, improving the availability of medical imaging infrastructure, setting stringent data quality control measures, enhancing the transparency and interpretability of models, employing robust techniques to prevent overfitting, strictly adhering to data privacy laws, and encouraging collaboration for resource sharing and expertise exchange. These steps are fundamental in advancing the diagnostic capabilities for Alzheimer's disease and could lead to better management and treatment outcomes for patients.

6 Proposed Approach

The procedure for the Alzheimer's disease prediction using imaging and machine learning involved the following listed below.

6.1 *Input*

In the endeavor of applying deep learning techniques to Alzheimer's disease (AD) research, the input data comprises various datasets associated with the disease, providing a comprehensive perspective of its intricacies. The sources of these datasets include variety of platforms, notably OpenNeuro (previously OpenfMRI) and the Alzheimer's Disease Neuroimaging Initiative (A.D.N.I.), both of which provided priceless insights into the research. The A.D.N.I. Databank, hosted on the A.D.N.I. website, constitutes a rich repository of data in various formats, including sMRI, PET, and more. Collaborators from the California Imaging Institute actively engage in AD identification within this expansive databank. The overarching goals of the A.D.N.I. study encompass the early detection and tracking of AD through the development of clinical, imaging, genetic, and biochemical biomarkers. This longitudinal multicenter study aims to detect AD at its earliest stages, facilitating intervention when it may be most effective. Additionally, it seeks to advance AD intervention, prevention, and treatment through novel diagnostic methods. OpenNeuro, on the other hand, serves as an open-science neuroinformatics database for task-based fMRI data, enabling neuroimaging researchers to share and disseminate their datasets. It adopts the Brain Imaging Data Structure (BIDS) format as its standard for shared and curated datasets. Within the realm of AD research, the choice of data types is crucial. fMRI data, acquired in different formats, presents a critical component. The traditional DICOM format, with its long-standing history of use, coexists with the newer NIFTI format, which introduces a 4D image dataset compared to DICOM's 3D format. NeuroImaging Informatics Technology Initiative (NIFTI) offers neuroscientists a robust and flexible system for quantitative analysis and visualization of neurofunctional fMRI data. In contrast, Digital Imaging and Communications in Medicine (DICOM) represents a standard for medical images, commonly used as the output file format for MRI scanners. However, its slice storage format often necessitates conversion to other analysis formats, such as NIFTI, to provide compatibility using a variety of tools for brain imaging analysis [11].

6.2 *Preprocessing*

Preprocessing plays a vital role in data mining, transforming raw data into a manageable and comprehensible format by removing unnecessary components and making essential alterations. Within the framework of FMRI data, preprocessing becomes essential to eliminate unwanted artifacts and standardize the data format. Various widely used neuroimaging data processing tools, such as SPM, AFNI, FSL, FreeSurfer, and fMRIPrep, facilitate this preprocessing, making it a fundamental step in image-based applications, particularly in magnetic resonance (M.R.) brain image classification and segmentation. In this work, the significance of preprocessing techniques is emphasized, focusing on the removal of the skull portion surrounding

brain tissues. In the context of Alzheimer's disease, a sequential algorithm for preprocessing involves a series of critical steps to ready the data for analysis and model development. Initially, diverse data sources including MRI scans, genetic information, and clinical records are collected. The subsequent phase comprises data cleaning, aimed at removing noise, handling missing values, and standardizing formats to ensure data integrity. Feature extraction follows, identifying relevant markers such as specific brain regions or genetic indicators. Standardization through feature scaling is applied to normalize values across different features for consistency. To handle the complexity of the data, dimensionality reduction techniques like PCA or t-SNE are employed, aiding in reducing the dataset's complexity while retaining essential information. Dividing the data into training and testing sets is crucial for model validation. Cross-validation techniques are then utilized to assess the model's robustness. Establishing a preprocessing pipeline allows for an organized, sequential flow for automated data preparation. Validation and quality checks ensure the integrity of the preprocessed data. Lastly, an iterative refinement process is adopted to continually enhance the preprocessing methodology, ultimately improving the performance of subsequent models aimed at diagnosing or understanding Alzheimer's disease (Fig. 2).

6.3 Machine Learning

In the world of medicine, machine learning has become a potent instrument that offers potential new paths for the early diagnosis and detection of conditions like Alzheimer's. In Alzheimer's disease (AD) detection projects, Techniques for machine learning may include harnessing various data sources, including medical imaging, genetic markers, and clinical information, to provide accurate and timely assessments. Although there are significant benefits to leveraging machine learning in AD detection, it's essential to consider the associated pros and cons.

On the positive side, machine learning enables early detection by identifying subtle patterns and deviations in data, potentially allowing for more effective interventions and improved patient outcomes. It seamlessly integrates diverse data types, such as MRI scans, genetic information, and cognitive assessments, providing a comprehensive view of a patient's condition. This multi-modal approach enhances diagnostic accuracy and helps in the creation of predictive models that assess an individual's risk of developing Alzheimer's disease based on their medical history, lifestyle factors, and genetic predisposition. Additionally, machine learning automates the analysis of large and complex datasets, reducing the burden on healthcare professionals and leading to faster and more cost-effective diagnoses. However, there are also challenges connected to the application of machine learning in Alzheimer's disease detection. Since the models of machine learning rely significantly on consistent and reliable data, the standard of the incoming data is crucial. Inconsistent or noisy data can lead to inaccurate results, potentially causing misdiagnoses or false alarms.

Model: "sequential"

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 128, 128, 3)	0
conv2d (Conv2D)	(None, 128, 128, 16)	448
max_pooling2d (MaxPooling2D)	(None, 64, 64, 16)	0
conv2d_1 (Conv2D)	(None, 64, 64, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0
dropout (Dropout)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 64)	0
dropout_1 (Dropout)	(None, 16, 16, 64)	0
flatten (Flatten)	(None, 16384)	0
dense (Dense)	(None, 128)	2097280
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 4)	260
Total params: 2,129,380		
Trainable params: 2,129,380		
Non-trainable params: 0		

Fig. 2 Information after preprocessing

It may be difficult to understand how some machine learning models, particularly those that employ deep learning techniques, arrive at their findings, that’s why they are often addressed as “black boxes.” This lack of openness may provide a challenge in gaining trust from healthcare professionals and patients. Overfitting is another concern, as machine learning models may perform exceptionally effectively on the training set but struggle to generalize to new, unseen data. Ensuring model generalizability is a critical challenge. Additionally, the application of sensitive medical data in models for machine learning raises concerns about patient privacy and data security, necessitating robust safeguards and compliance with regulatory requirements.

Finally, developing and maintaining machine learning models for Alzheimer’s detection can be resource-intensive, requiring specialized expertise, computational power, and access to large datasets. In conclusion, machine learning holds great potential for enhancing Alzheimer’s disease detection by enabling early diagnosis, integrating diverse data sources, and facilitating predictive modeling. However, addressing challenges related to data quality, model interpretability, overfitting, data privacy, and resource requirements is crucial to realizing the full benefits of machine learning in Alzheimer’s detection projects. Collaborative efforts between data scientists, healthcare professionals, and regulatory bodies are essential to harness this technology effectively while mitigating its limitations (Fig. 3).

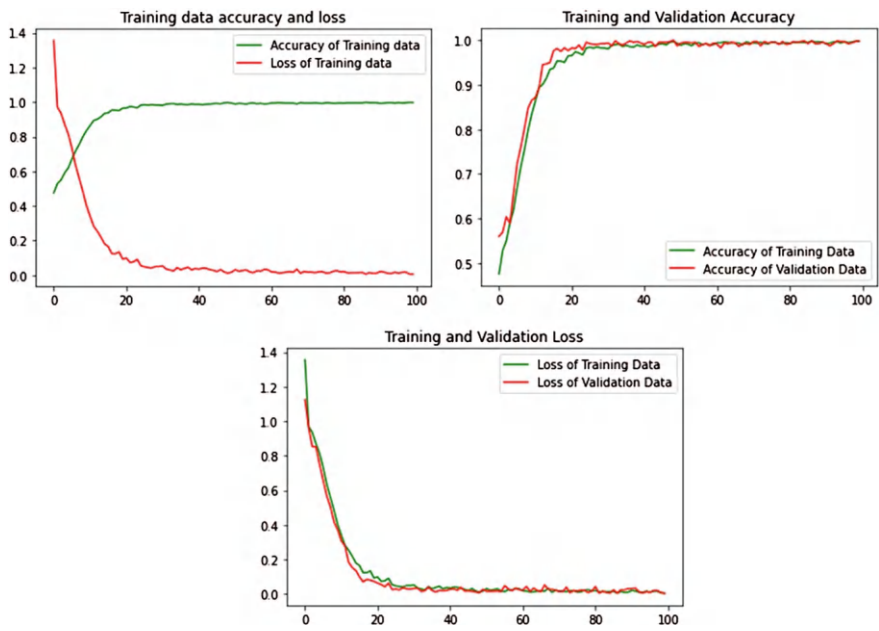


Fig. 3 Accuracy and loss

6.4 *Imaging*

Imaging techniques have been instrumental in detecting and understanding Alzheimer's disease (AD), providing crucial insights into the structural and functional alterations that take place in the brain. These imaging methods, such as magnetic resonance imaging (MRI), positron emission tomography (PET), and functional MRI (fMRI), have been utilized in AD detection projects, each offering its unique benefits and limitations. The ability of imaging to provide comprehensive, non-invasive anatomical knowledge and among the main advantages of using the brain's physiology to identify AD. For example, MRI can disclose structural abnormalities, like brain atrophy, a characteristic feature of AD. PET imaging, conversely, can monitor the existence of specific biomarkers linked to AD, such as amyloid plaques and tau protein tangles. Specifically useful applications of functional MRI (fMRI) include evaluating brain function and connectivity, assisting in understanding how AD impacts neural networks. Another significant advantage is the possibility of early detection. Imaging techniques can spot subtle alterations in the brain that occur in the initial stages of AD, even before noticeable cognitive symptoms manifest. Early detection is crucial for timely intervention and the development of effective treatment strategies. Additionally, imaging can aid in differential diagnosis, helping to distinguish AD from other neurodegenerative disorders with similar clinical symptoms.

However, there are notable limitations and challenges associated with imaging in AD detection. One key limitation is the cost and accessibility of imaging equipment. MRI and PET scans can be expensive, and the availability of these technologies may be limited in certain regions or healthcare settings. Furthermore, imaging may not be suitable for all individuals, particularly those with claustrophobia or conditions that prevent them from undergoing these procedures. Interpretation of imaging data also presents challenges. While imaging can detect structural or functional abnormalities, it may not provide a definitive diagnosis of AD. Biomarkers like amyloid and tau protein deposits can be present in cognitively healthy individuals, making it essential to consider clinical and cognitive assessments alongside imaging results. Moreover, there are ethical concerns related to imaging, including issues of patient consent, privacy, and the responsible use of sensitive medical data. Ensuring the ethical and responsible use of imaging technology in AD detection is of paramount importance. Imaging techniques have revolutionized our understanding of Alzheimer's disease, offering valuable tools for early detection and differential diagnosis. Their non-invasive nature and ability to provide detailed insights into brain structure and function make them indispensable in AD research and clinical practice. However, challenges related to cost, accessibility, interpretation, and ethics must be carefully addressed to maximize the benefits of imaging in AD detection projects. Collaborative efforts between researchers, clinicians, and policymakers are essential to harness the potential of imaging in the fight against Alzheimer's disease. In our Alzheimer's disease proposed model, a convolutional neural network (CNN) model is deployed for its efficacy in analyzing complex data like MRI scans. The CNN architecture

Fig. 4 CNN model performance

Metric	Value(%)
Accuracy	99.0
Precision	89
Recall	94
F1 Score	91
ROC AUC	96

Fig. 5 CNN model matrix

Confusion Matrix	Values
True Positive	450
True Negative	430
False Positive	70
False Negative	50

comprises convolutional layers for feature extraction, capturing intricate patterns in brain imaging data. Pooling layers condense the information while retaining essential features. The model includes fully connected layers for classification, learning intricate relationships between features extracted. Utilizing ReLU activation functions helps introduce nonlinearity, crucial for complex data interpretation. Dropout layers mitigate overfitting, enhancing generalization. For our project, this CNN architecture is adept at learning hierarchical representations from brain images, aiding in the diagnosis and understanding of Alzheimer’s disease with high accuracy and efficiency.

The tables shown in Figs. 4 and 5 showcase key performance factors such as accuracy, precision, recall for the CNN model used in the project. Additionally, the confusion matrix displays the counts of true positives, true negatives, false positives, and false negatives providing a comprehensive evaluation of the model’s predictive capabilities.

7 Results

The superiority of the convolutional neural network (CNN) model in Alzheimer’s disease prediction, as compared to other models in the table—random forest, support vector machine (SVM), logistic regression, and *K*-nearest neighbors (*K*-NN)—emanates from CNN’s specialized architecture tailored for image and pattern recognition tasks. CNN excels in extracting intricate features from brain imaging and medical data, vital in detecting subtle abnormalities associated with Alzheimer’s. Its hierarchical feature learning capabilities, utilizing convolutional and pooling layers, enable automatic feature extraction without manual feature engineering. Specifically designed to interpret spatial relationships, CNNs effectively discern patterns within brain scans, crucial for early diagnosis. This model’s adaptability to diverse data

Fig. 6 Comparison of models

Algorithm	Accuracy(%)
CNN	99.0
Random Forest	85.2
Support Vector Machine	82.6
Logistic Regression	78.9
K-Nearest Neighbors	76.4

11/11 [=====] - 0s 12ms/step - loss: 0.0163 - accuracy: 0.9907

Fig. 7 CNN model results

types and its capacity to learn complex spatial dependencies significantly outperform traditional algorithms. While other models possess utility in medical analysis, CNN’s inherent strength in capturing nuanced details in brain images contributes to its higher accuracy (99.0%) in Alzheimer’s prediction. This underscores the CNN’s potential in revolutionizing early diagnosis and intervention, crucial in combating neurodegenerative diseases, as it can precisely recognize intricate patterns indicative of Alzheimer’s, offering promising prospects for medical advancements and enhancing patient care (Figs. 6 and 7).

8 Conclusion

In conclusion, the quest to revolutionize Alzheimer’s disease (AD) diagnosis and monitoring is a multi-faceted endeavor that combines cutting-edge technology, neuroscience, and healthcare. This research integrates the powerful capabilities of machine learning, graph theory, and advanced imaging techniques to tackle one of the most formidable challenges in modern healthcare. The urgent need for early detection in the face of Alzheimer’s disease, a neurodegenerative disorder with no cure, is paramount. Early intervention can significantly impact the course of the disease, offering hope for improved patient outcomes and a higher quality of life.

The integration of machine learning and graph theory holds great promise in enhancing the accuracy of AD diagnosis and the ability to predict disease progression. By merging functional magnetic resonance imaging (fMRI) data and graph theory measures, this approach allows for the analysis of brain connectivity at a macroscopic level, shedding light on the intricacies of AD’s underlying pathology. Graph theory provides a systematic framework for scrutinizing brain networks, quantifying the integration and segregation of various brain regions, and uncovering subtle yet significant differences in brain connectivity patterns indicative of the disease.

Machine learning algorithms, known for their ability to identify intricate patterns in complex data, play a pivotal role in distinguishing cognitively normal individuals from those with Alzheimer’s disease. The research systematically evaluates different machine learning methods and features to identify the most accurate and

robust models for AD diagnosis. Furthermore, the study extends its focus to predict the trajectory of the disease in individuals diagnosed with mild cognitive impairment (MCI), a precursor to AD. This predictive capability is essential for early intervention, as it allows healthcare professionals to tailor treatment strategies to slow the progression of the disease. Imaging techniques, such as MRI and PET, provide critical insights into the structural and functional changes in the brain associated with AD. Early detection is made possible through these imaging modalities, allowing for timely interventions. However, challenges such as cost, accessibility, and data interpretation must be addressed to maximize their potential. Despite the remarkable promise of these techniques, challenges remain, including data quality, model interpretability, overfitting, data privacy, and resource requirements. A collaborative effort between data scientists, healthcare professionals, and regulatory bodies is crucial to effectively harness this technology while mitigating its limitations. In summary, the integration of machine learning, graph theory, and advanced imaging techniques is a powerful approach to enhance AD diagnosis and monitoring. This research holds the potential to transform the landscape of Alzheimer's disease care, offering early detection, accurate diagnosis, and personalized treatment strategies. As we stand at the intersection of technology, neuroscience, and healthcare, the pursuit of more accurate and accessible AD diagnosis remains an imperative goal. The promise of early detection and improved patient outcomes stands as a powerful driving force in the ongoing battle against this relentless neurodegenerative disease.

References

1. eJain R, Jain N, Aggarwal A, Hemanth DJ (2019) Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images
2. Aljehani SS, Alhazmi RA, Aloufi SS, Aljehani BD, Abdulrahman R (2018) Applying IoT technology for monitoring Alzheimer's patients
3. Jin D, Xu J, Zhao K, Hu F, Yang Z, Liu B, Jiang T, Liu Y (2019) Attention-based 3D convolutional network for Alzheimer's disease diagnosis and biomarkers exploration. In: *Proceeding IEEE 16th international symposium biomedicine imaging (ISBI)*, pp 1047–1051
4. Yadav R, Gautam A, Mishra RB (2018) Classification of Alzheimer using fMRI data and brain network
5. Chen Z, Lei H, Huang Z, Lei B (2021) Latent space learning and feature learning using multi-template for multi-classification of Alzheimer's disease
6. Arafa DA, Moustafa HE-D, Ali-Eldin AMT, Ali HA (2022) Early detection of Alzheimer's disease based on the state-of-the-art deep learning approach
7. Jo T, Nho K, Saykin AJ (2019) Deep learning in Alzheimer's disease: diagnostic classification and prognostic prediction using neuroimaging data
8. Khazae A, Ebrahimzadeh A, Babajani-Feremi A (2015) Identifying patients with Alzheimer's disease using resting-state fMRI and graph theory
9. Eickhoff S, Nichols TE, Van Horn JD, Turner JA (2016) Sharing the wealth: neuroimaging data repositories. *Neuroimage* 124:1065–1068
10. Fujibayashi D, Sakaguchi H, Ardakani I, Okuno A (2019) Nonlinear registration as an effective preprocessing technique for deep learning-based classification of disease
11. Zhu T, Cao C, Wang Z, Xu G, Qiao J (2020) Anatomical landmarks and DAG network learning for Alzheimer's disease diagnosis

Detection of Fire using Deep Learning Models



Mohith Krishna M. Kamath, R. Sinchana, K. Sreenidhi,
and Jeshma Nishitha Dsouza

Abstract Fire accidents are among the incidents posing a significant threat to valuable assets. They can cause damage economically, financially and lead to the depletion of natural resources like forests and jeopardize human lives. In this paper a deep learning model is proposed that assist in detecting fires in various environments like forests, stores, commercial kitchens, meeting rooms, etc. This proposed approach demonstrated the effectiveness of deep learning models for fire detection using a hybrid model called CNN-LSTM (Convolutional Neural Networks Long Short-Term Memory). However, it is worth mentioning that training these models can be computationally intensive and require large datasets. Additionally, this study suggests combining MobileNetV2 and Xception architectures to form an ensemble of convolutional neural networks (CNNs) for detecting fires. This paper proposes utilizing the Xception models alongside MobileNetV2 architectures to create an ensemble, for fire detection purposes. Experimental results show that the ensemble of Xception, ResNet50, and Inception gave a best result of 94% accuracy.

Keywords Fire detection · Deep learning · CNN-LSTM hybrid model · Ensemble models · Xception · MobileNetV2 · ResNet50 · Inception

1 Introduction

Fire, one of the oldest discoveries in our evolutionary journey, has served as both a powerful friend, bringing warmth, protection, and culinary inventiveness, and a terrible opponent, unleashing disaster when left unchecked. Fire accidents have been a relentless and ever-present threat in the story of human existence. The nature and causes of fire accidents evolved alongside our understanding and mastery of fire,

M. K. M. Kamath · R. Sinchana · K. Sreenidhi · J. N. Dsouza (✉)
St. Joseph's University, Bengaluru, India
e-mail: jeshma.dsouza@sju.edu.in; jeshma.research@gmail.com

M. K. M. Kamath
e-mail: mohithkrishnakamath@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025
N. R. Shetty et al. (eds.), *Advances in Artificial Intelligence and Machine Learning*,
Lecture Notes in Electrical Engineering 1335,
https://doi.org/10.1007/978-981-96-3105-6_19

269

reflecting the developing complexities of our societies and technologies. As varied as the flames themselves are the reasons behind fire accidents. They range from inadvertent causes like electrical malfunctions, culinary accidents, or heating system breakdowns—which are frequently caused by mistakes made by people or broken equipment—to unpredictable natural disasters like lightning strikes, volcanic eruptions, and wildfires. Deliberate acts of arson also highlight the potentially destructive nature of human intent, which adds another level of intricacy to this age-old problem.

Fire accidents causes destruction of property. They have the power to destroy ecosystems and unleash contaminants into the surrounding environment. Long-term health concerns and respiratory difficulties are faced by survivors, they put a pressure on economies through insurance claims, reconstruction costs and commercial disruptions. The psychological cost which is sometimes overlooked consists of anxiety and anguish, are impacted when essential infrastructure is disrupted [1].

By using a variety of technologies, safety precautions have been greatly enhanced by the detection and prevention of fire occurrences. Heat sensors, smoke detectors, and sophisticated fire suppression systems are important examples of these [2]. These technologies, however, are not perfect and have flaws that can limit their usefulness in specific situations. Smoke detectors are essential for detecting fires; however, they can malfunction in low-smoke or slow-burning fires. Their placement also affects how efficient they are; thus, they are not appropriate for outdoor or highly humid areas. While automatic sprinklers and fire-extinguishing agents are examples of advanced fire suppression systems that are generally dependable, they can malfunction as a result of improper maintenance. Water-based systems' effectiveness may be limited if they are used in situations where flammable substances, electrical equipment, or chemical reactions are involved.

In the recent past, AI-enabled surveillance technologies have overcome several shortcomings; they represent a substantial advancement over traditional methods—heat sensors. AI surveillance is flexible for threat identification, in contrast to traditional sensors, it is highly proficient in identifying a broad range of objects, behaviors, and anomalies [3]. It also reacts to minute variations in movement and temperature, which improves the early identification of possible threats. These systems can monitor vast regions effectively due to their scalability and capacity to analyze data on their own, eliminating the need for continual human supervision. Additionally, by using data analysis to improve decision-making, AI surveillance helps with resource allocation and security threat prediction. Concerns are addressed by privacy protection measures, and integration with other security systems builds a full security ecosystem. Thus, we see that AI-enabled surveillance effectively overcomes the drawbacks of previous techniques to provide a more flexible, effective, and all-encompassing approach to security and surveillance.

This paper uses cutting-edge models and tools for AI-enabled fire detection, such as a hybrid CNN-LSTM model and a combination of the Xception and MobileNetV2 models. The Xception and MobileNetV2 fusion combines the advantages of these well-known deep learning models to enhance fire detection. Furthermore, to efficiently analyze static images and temporal patterns, the CNN-LSTM hybrid model combines convolutional neural networks (CNNs) with long short-term memory

(LSTM) networks. The accuracy of fire detection could be improved by using this method. To improve the accuracy and efficacy of fire detection systems, this research assesses these models using fire image datasets. By integrating well-established deep learning architectures, these novel approaches have the potential to advance the field of fire detection technologies.

2 Related Works

A dataset comprising 6,911 images of smoke and forest fires was utilized for this study, which are collected from various sources—Google, Kaggle repository, satellite imagery, and the BoWFire dataset. Despite having few samples in their target domain, they used the power of transfer learning to classify these photos by using pre-existing information from the ImageNet dataset. With VGG16 reaching 93.2% accuracy, InceptionV3 at 94.5%, and Xception leading at 95.1%, this method produced encouraging results. The use of a diversified dataset, improved model accuracy, and the application of the “Learning without Forgetting” (LwF) principle to prevent the loss of prior information are all advantages of their approaches. The risk of overfitting the models to the training data, which may compromise their performance on new data, as well as the computing requirements of developing and deploying these sophisticated models are potential downsides [1].

In this investigation, researchers concentrated on the creation and assessment of Fire Net, a deep learning system built to identify current fires from Landsat-8 imagery. They used a two-stream pixel classification approach with a distinctive design to improve the detection performance. The identification of current fires was handled by one stream, while background and non-fire items were handled by the other. This method addresses the problem of imbalanced datasets by simulating actual conditions when fires are seldom. Preprocessing of the Landsat-8 photos included orthorectification and radiometric correction. The deep learning architecture used residual blocks to address gradient problems and multi-scale residual convolution layers to ensure robustness against scale fluctuations. A hybrid loss function and the Adam optimizer were used to train the network using a set of hyperparameters. Visual analysis and the computation of multiple metrics, such as recall, overall accuracy, precision, and F1-score, were part of the evaluation process. The model’s performance was evaluated against a variety of machine learning algorithms, including random forest, K-nearest neighbors, support vector machine, multilayer perceptron, and extreme gradient boosting, in addition to the well-known CNN, MSR-U-Net. The robustness and efficacy of this method in identifying small fires are its advantages; however, the need for meticulous hyperparameter tuning and extended training periods are possible disadvantages [2].

To address the shortcomings of conventional computer vision-based fire detection strategies that rely on static and short-term temporal characteristics, a novel deep learning-based fire detection method called decision through aggregation (DTA) was suggested in this study. For increased accuracy, the DTA approach closely resembles

human decision-making processes. There are three main elements to the architecture. First, suspected regions of fire (SRoFs) and non-fire objects are distinguished in video frames using the Faster R-CNN deep object detection model. Bounding boxes are used to capture different classes, such as smoke, flames, and non-fire items. After that, the spatial characteristics inside these bounding boxes are extracted. In the second segment, short-term firing choices are made using LSTM networks that gather and analyze spatial information over a brief period. At this point, smoke and flame are indistinguishable. This action simulates a human observer's fleeting decision-making process. A majority voting procedure is used in the last part to integrate the short-term decisions into a long-term firing decision. Furthermore, the dynamic alterations in the SRoF areas, such as smoke and flame, are observed to offer insights into the changing nature of fire behavior. The technique is excellent in guaranteeing strong SRoF detection and capturing the context data surrounding the bounding boxes. Bounding boxes for flames, smoke, and non-fire regions are mostly provided by the Faster R-CNN object detection model; nevertheless, problems can develop when non-fire items mistakenly get detected as fire. These difficulties are lessened, meanwhile, by the deep object detection model's vast context data collection. The approach's merits are in its capacity to closely mimic human decision-making by considering the temporal and dynamic characteristics of fire; nevertheless, its drawbacks include the requirement for meticulous hyperparameter tuning and a possible reliance on the length of majority voting in order to enhance accuracy [4].

A novel method utilizing Vision Transformer (ViT) technology has arisen in response to the increasing demand for quick and accurate fire monitoring in security video systems. ViT converts images into a series of patches as opposed to Conventional Convolutional Neural Networks (CNNs), which treat every image pixel equally. This enables context-based selective attention to certain regions of the image. The ViT's attention mechanism solves the problems caused by tiny flames and allows for early fire detection. A refined Swin Transformer architecture, which effectively computes self-attention with local windows and is especially useful for high-resolution images, is employed to maintain detection speed. The Vision Transformer outperformed existing state-of-the-art techniques with an outstanding 98.54% classification accuracy in image fire dataset experiments, showing promising outcomes [5]. Convolutional neural networks (CNNs) have been highlighted as a viable approach in recent research on image-based fire detection. Prior studies, on the other hand, mostly assessed CNNs on balanced datasets, which might not accurately represent real-world settings when fires are uncommon. This paper presents a novel approach for fire detection employing even deeper CNNs, VGG16, and Resnet50, optimized with completely connected layers. Their analysis uses a dataset that is intentionally skewed to imitate real-world problems; non-fire photographs predominate over fire images. The results show that adding fully connected layers for fine-tuning increases accuracy, even though it requires more training time. These deeper CNNs display commendable performance on a more realistic dataset, with Resnet50 narrowly beating out VGG16. Although there will still be a trade-off in terms of longer training sessions, these results point toward more effective fire detection systems [6].

Many classifiers were contrasted in the experiments carried out to assess the suggested approaches. The baseline classifiers, CNN-Raw and SVM-Raw, have accuracy rates of roughly 93.1% and 92.2% on the training set, respectively. They were trained from scratch using raw pixel data. In terms of detection rate, the CNN classifier fared better than the SVM, but it had a little higher false alarm rate. A full image CNN classifier that was fine-tuned showed amazing accuracy of up to 100% in testing and training datasets. In comparison to raw pixel classifiers, linear SVM and Nonlinear Neural Network (NN) classifiers achieved higher accuracy rates when employing Pool-5 features. On the other hand, NN-Pool5 computed at a slower rate—1.4 s per image, as opposed to 0.16 s for SVM-Raw. Although the complete picture CNN classification took the longest, real-time fire detection systems could benefit from GPU implementation's considerable speed gain. Because NNPool5 regularly beat the other classifiers, the qualitative results were in favor of it. All in all, the strategies showcased attain excellent precision; however, several classifiers require longer computation times as a cost [7].

3 Dataset Overview

Earlier efforts in this discipline concentrated on curating databases that mostly featured images from specific categories or locales. The goal of our research was to create a comprehensive dataset of fire-related pictures from various locations, including both indoor and outdoor scenarios. To accomplish this, we used web scraping techniques to gather a large quantity of images. The images were obtained systematically from a dedicated website using carefully chosen keywords linked with both fire-related and non-fire settings. This method enabled us to create a dataset that gives a comprehensive and more diverse representation of fire-related imagery, laying the groundwork for the research described in this paper. Following which, the scraped dataset was carefully inspected to identify and eliminate duplicate photos while also using an efficient manual curation process to remove any images that were judged inappropriate or unrelated to our research aims. This careful data cleaning and curation process ensures the quality and relevance of the dataset, setting the foundations for subsequent analysis and investigations in this study. A handpicked collection of 6668 photos depicting fire incidents and 6109 images depicting non-fire events is used in this study. Attempts were made to train the model to prevent false alarms by including photos of smoke-like and fire-like images in the non-fire class such as images of fog, glare lights, street lights, sunset. Following collection, rigorous preprocessing and augmentation operations were meticulously carried out in accordance with the model employed in the paper. The cleaned datasets were split into three categories: for testing, validation, and training. 10% each of the classes was used for testing and validation, and the remaining 80% was used for training the model. Table 1 displays the distribution of image in the training, testing, and validation dataset. Sample Images for fire and non-fire classes are shown in Fig. 1.

Table 1 Data description

Dataset	Fire	Non-fire	Total
Training	5334	4887	10,221
Testing	668	612	1280
Validation	666	610	1276
Total	6668	6109	12,777



Fig. 1 Sample images for fire and non-fire classes which are taken from dataset

4 Methods and Models

See Fig. 2.

4.1 CNN-LSTM Hybrid Model

The fusion of convolutional neural networks (CNNs) and long short-term memory networks (LSTMs) in fire detection represents a sophisticated synergy of spatial and temporal information processing. CNNs, renowned for their prowess in discerning intricate spatial features within images, seamlessly unveil critical elements such as edges, textures, and intricate patterns crucial for the discrimination between fire and non-fire imagery.

Enhancing precision is a paramount objective in fire detection, with the imperative to minimize the occurrence of false positives. Here, the LSTM plays a pivotal role by virtue of its capacity to contextualize and consider the historical progression of detections. This dynamic capability allows for the correction of false alarms; for

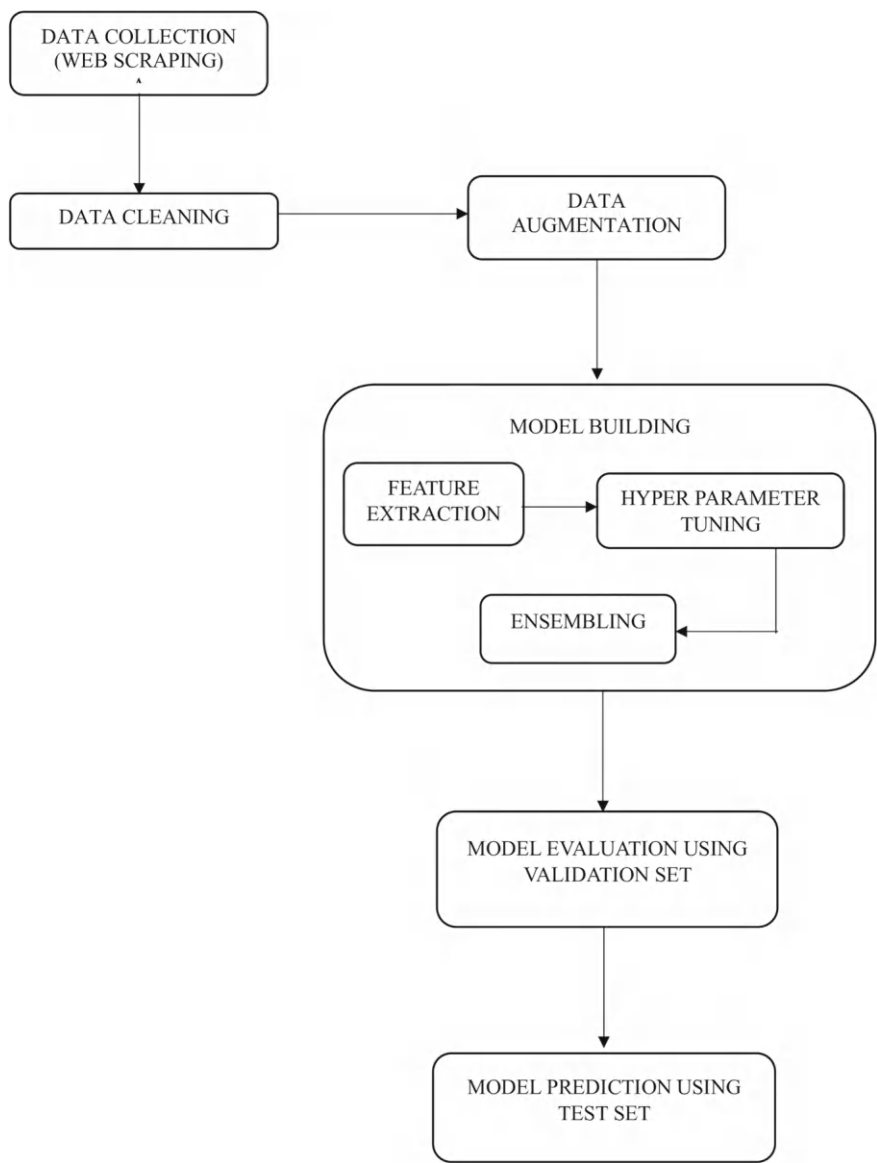


Fig. 2 Illustration of the proposed model’s practical implementation

example, in the event of smoke detection in one frame followed by its transient absence in subsequent frames, the LSTM brings its correctional influence to bear.

In the realm of fire detection, diversity in lighting and environmental conditions poses a formidable challenge. The CNN-LSTM architecture emerges as a robust solution, adept at deciphering fire patterns across an array of scenarios by virtue of

its ability to process sequences of images. For non-sequence of images, each image is taken as a separate sequence. This amalgamation of deep learning techniques not only ensures the capture of diverse fire manifestations but also empowers the model with a level of adaptability and sophistication that is instrumental in addressing the multifaceted nature of fire detection.

4.2 Ensemble of Xception and MobileNetV2

In this paper, we implemented an ensemble of two transfer learning models, Xception and MobileNetV2 for classification of images as the individual classification accuracy of the pre-trained Xception and MobileNetV2 models was only 60 leaving room for improvement. These two classifiers are ensembled into a strong classifier using the ensemble technique of averaging. The pre-trained Xception and MobileNetV2 models were combined with specially created output layers for binary classification. The ensemble model integrates the results from the two models effectively by utilizing the variety of each models advantages and disadvantages. This ensemble strategy significantly improved the classification tasks accuracy which rose to an astounding 87, demonstrating the value of model ensembling in enhancing system's overall performance.

We used the feature extraction capabilities of two transfer learning models—Xception and MobileNetV2—that have been pre-trained on large image datasets in our study. MobileNetV2 is superior in efficiency and generalization, while Xception is best at capturing fine spatial information. From our dataset, they collectively extracted rich feature representations. Our ensemble technique, which combined the feature vectors from Xception and MobileNetV2, was the main source of innovation. We were able to create a harmonious balance by averaging their forecasts, which improved our classification accuracy and reduced model biases.

We used common image classification metrics such as accuracy, precision, recall, F1-score, and ROC-AUC to assess the performance of our ensemble model. These metrics offered a thorough evaluation of our model's ability to discriminate between images with and without fire.

Essentially, our work provides a useful ensemble model that combines the feature extraction powers of MobileNetV2 and Xception. We achieved amazing image classification performance, especially in the critical domain of fire detection, by strategically utilizing ensemble approaches. This methodology exhibits potential for a multitude of practical uses, such as augmenting security and safety.

Although our ensemble model has shown encouraging improvement in improving the accuracy of picture categorization, it is important to understand the limits of the individual Xception and MobileNetV2 models. Due to the sensitivity of these models to the quality and diversity of the data, when input photos deviate greatly from training data, accuracy may be compromised. This problem is particularly relevant in real-world situations where there are variations in image quality because to things like lighting and artifacts. Moreover, our tests showed that these models might

not perform well in situations with complicated backdrops, which could affect their accuracy. Post-processing techniques are frequently needed for binary classification applications, such as fire detection, to refine results and impact model performance. Our study demonstrates the benefits and drawbacks of our group method, emphasizing the necessity of customizing solutions for particular use cases and resolving issues with data preprocessing, model optimization, and post-processing. Since there is not a single, perfect method for classifying images, our research recommends investigating stronger models or hybrid strategies in the future.

5 Results and Discussions

From the confusion matrix Fig. 3a, out of 1280 test cases, the CNN-LSTM hybrid model has correctly detected 611 fire images and 567 non-fire images, thus giving an accuracy of 87%. But it has failed detect 57 fire images and misclassified detected 45 non-fire images which amounts to Type II and Type I error, respectively. We must aim to prevent Type II error as it leads to disastrous events, as compared to Type I error which leads to false alarms. The classification report Table 2 supports the above claim with the performance metrics of the model depicting precision score of 0.87, recall score of 0.88, and f1 score of 0.88 (Fig. 4).

From the confusion matrix Fig. 4, out of 970 test cases, the ensemble model has correctly detected 660 fire images and 310 non-fire images, thus giving an accuracy of 92%. But it has failed detect 62 fire images and misclassified detected 4 non-fire images which amounts to Type II and Type I error, respectively. We must aim to prevent Type II error as it leads to disastrous events, as compared to Type I error which leads to false alarms. The classification report supports the above claim with the performance metrics of the model depicting precision score of 0.93, recall score of 0.91, and f1 score of 0.92. The number of images that are failed to detect is 4 which is less than the CNN-LSTM model; thus, Type II error is also less; hence, the ensemble model is better than the CNN-LSTM model.

The CNN-LSTM hybrid model achieved an impressive accuracy rate of 92%. The model also gave a high F1-score as shown in Table 2. The confusion matrix in Fig. 4 shows a low number of false positives and false negative count, thus reduced false alarms.

Our experimental data shows that our ensemble model outperforms individual models consistently. It improved accuracy while simultaneously lowering the likelihood of false positives, making it a reliable tool for detecting fire in image data. The classification report shows a high F1-score depicting consistent performance of the model (Table 3). The confusion matrix in Fig. 4 shows a low number of false positive and false negative counts implying a reduced false alarm.

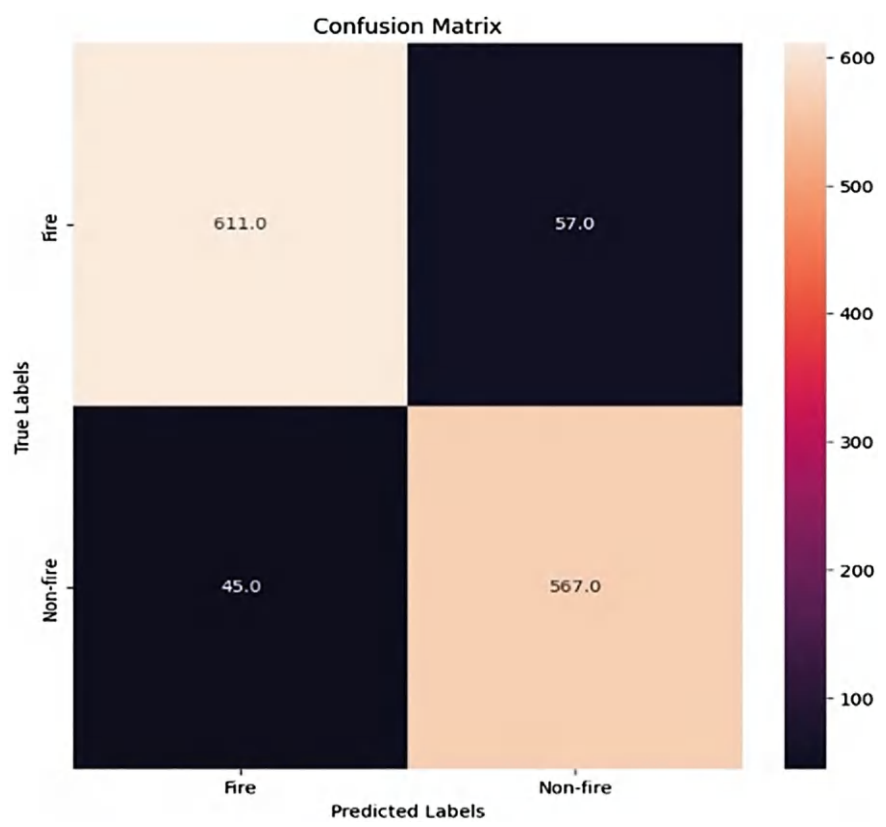


Fig. 3 Confusion matrix for the CNN-LSTM hybrid model

Table 2 Classification report of the CNN-LSTM hybrid model

Type of image	Precision	Recall	F1-score	Support
Fire	0.87	0.88	0.88	666
Non-fire	0.87	0.86	0.86	610

6 Conclusion

This study tackles the crucial problem of fire detection in diverse settings, emphasizing the risk to both human life and expensive assets. We have created learning models for fire detection, including CNN-LSTM hybrid model with good resut. Furthermore, the study supports the creation of an ensemble of deep learning models for fire detection by combining the Xception and MobileNetV2 architectures. Experiments show that the study suggested CNN-LSTM hybrid model has an amazing accuracy rate of 92%, consistently outperforming both state-of-the-art methods and

Fig. 4 Confusion matrix for ensemble of Xception and MobileNetV2

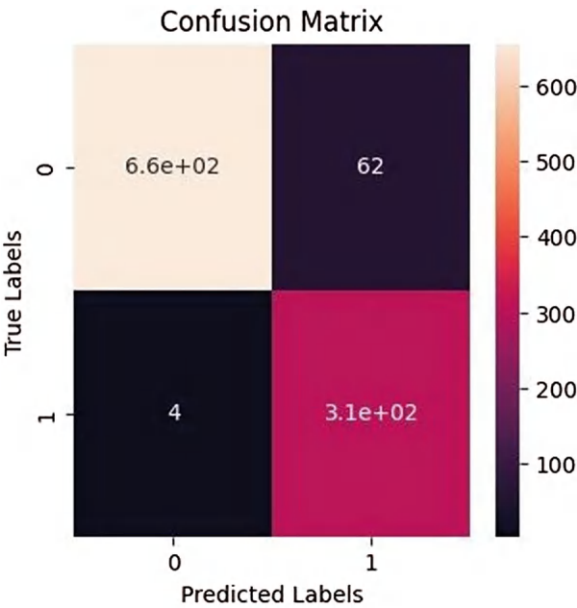


Table 3 Classification report for the ensemble of Xception and MobileNetV2

Type of image	Precision	Recall	F1-score	Support
Fire	0.93	0.91	0.92	668
Non-fire	0.91	0.93	0.92	612

current models. In a variety of contexts, this research helps to improve asset protection and fire safety. An ensemble of Xception, ResNet50, Inception gave a very good accuracy of 94% and prediction among all the models.

People can also try using different models and ensemble them for the good accuracy. Utilizing models equipped with a real-time alarm system will contribute to resolving this significant problem.

References

1. Sathishkumar VE, Cho J, Subramanian M et al (2023) Forest fire and smoke detection using deep learning-based learning without forgetting. *Fire Ecol* 19:9

2. Namozov A, Im Cho Y (2018) An efficient deep learning algorithm for fire and smoke detection with limited data. *Adv Electr Comput Eng* 18(4):121–128

3. Ahmad K, Khan MS, Ahmed F et al (2023) FireXnet: an explainable AI-based tailored deep learning model for wildfire detection on resource-constrained devices. *Fire Ecol* 19:54

4. Seydi ST (2022) Fire-Net: a deep learning framework for active forest fire detection. *J Sens* 2022:8044390, 14

5. Xu R, Lin H, Lu K, Cao L, Liu Y (2021) A forest fire detection system based on ensemble learning. *Forests* 12(2):217
6. Sharma J, Granmo OC, Goodwin M, Fidge JT (2017) Deep convolutional neural networks for fire detection in images. In: Boracchi G, Iliadis L, Jayne C, Likas A (eds) *Engineering applications of neural networks*. EANN 2017. *Communications in computer and information science*, vol 744. Springer, Cham
7. Zhang Q, Xu J, Xu L, Guo H (2016) Deep convolutional neural networks for forest fire detection. *Proceedings of the 2016 international forum on management, education and information technology application PB—Atlantis Press*, pp 568–575
8. Nguyen MD, Vu HN, Pham DC, Choi B, Ro S (2021) Multistage real-time fire detection using convolutional neural networks and long short-term memory networks. *IEEE Access* 9:146667–146679
9. Khudayberdiev O, Zhang J, Elkhailil A, Balde L (2022) Fire detection approach based on vision transformer. In: Sun X, Zhang X, Xia Z, Bertino E (eds) *Artificial intelligence and security*. ICAIS 2022. *Lecture Notes in Computer Science*, vol 13338. Springer, Cham
10. de Almeida Pereira GH, Fusioka AM, Nassu BT, Minetto R (2021) Active fire detection in Landsat-8 imagery: a large-scale dataset and a deep-learning study. *ISPRS J Photogrammetry Remote Sens* 178:171–186
11. Taspinar YS, Koklu M, Altin M (2021) Fire detection in images using framework based on image processing, motion detection and convolutional neural network. *Int J Intell Syst Appl Eng* 9(4):171–177

Anxiety or Depression Prediction of MDD in Children Using Ensemble Learning: A Systematic Review



C. R. Anju and Shiny Duela

Abstract AI and cognitive psychology aims to understand and explain the intelligent behavior of human beings to develop sophisticated AI systems that mimic human cognition. This study examined artificial intelligence with cognitive psychology to detect MDD and its subtype in children of India. MDD is becoming more widely identified as a serious issue affecting children and adolescents. According to the World Health Organization, depression is one of the leading causes of disability. This review's objective is to evaluate MDD and its subtype in children of India. A scoping review of 50 articles after screening and identification is used to categorize mental health problems other than MDD among children and an ensemble model is used to categorize MDD patients with other mental disorders. The data obtained from the model has been cleaned using a pre-processing methodology and recursive feature elimination is used to select the relevant feature. The prevalence reported in the area of children's mental health across research varies greatly, predominantly because of methodological discrepancies. There is a lack of knowledge regarding the symptom profile and risk factors that result in the cause and findings of depression in children and adolescents.

Keywords Artificial intelligence · Cognitive psychology · MDD · Ensemble learning

1 Introduction

Major depressive disorder (MDD) is a common psychological disorder [1] in adults, children, and adolescents, due to symptom variances from the adult criteria, it is frequently misdiagnosed and mistreated in children [2]. This severe psychological

C. R. Anju · S. Duela (✉)

SRM Institute of Science and Technology Ramapuram, Chennai, India

e-mail: shinyduj@srmist.edu.in

C. R. Anju

e-mail: anjucr@gmail.com

condition is described by persistent emotions of desolation and despair together with a loss of passion or enjoyment in previous hobbies. [3]. The symptoms comprise persistent melancholy or anxiety changes in appetite, sleep disorders (hypersomnia or insomnia), fatigue or a decrease in energy experiencing trouble focusing or making decisions, negative self-worth or excessive guilt, and frequent suicidal thoughts [4]. The discrepancy between a low incidence of depressive illnesses and the high prevalence of depressive symptoms on screening measures suggests that a large number of persons do not fit the MDD diagnosis criteria, but show some type of subclinical illness. The connection between this illness (or circumstances) to MDD is not well understood, yet subsyndromal depression is a reliable indicator of what will happen next initial stages of MDD, hence the identification of MDD when combined with other depressive disorders is difficult, MDD improves in confirming the existence of other depressive disorders. The severity of symptoms determines the first treatment plan, which may involve counseling, medical procedures, or both [5]. In 2008, major depressive disorder (MDD) remained the third largest cause of disease burden worldwide, by 2030, estimates from the World Health Organization (WHO) [6]. Cognitive ability, mood, and developmental stages of children are all affected by the psychological illness associated with depression in children [6]. Children might get behind in regular activities, particularly if depression persists for a long time without treatment. It is essential to understand the actual causes of depression, which could impact the growth of kids and mental health. Several research has analyzed the effectiveness and harmful effects of treatment in children and adolescents, while other studies have assessed the possibility of harmful effects including worsening suicidal thoughts and behaviors.

2 Motivation

MDD is a highly prevalent and incapacitating problem among children, but frequently goes undiagnosed, resulting in significant treatment delays. For this reason, understanding and treating children's mental diseases necessitates early identification and diagnosis [7]. This study addresses the severity and subtype of MDD can be estimated using the predictor. There are two primary challenges to effectively treating MDD. To determine who is infected and make sure that appropriate therapy is provided efficiently. MDD frequently remains undiagnosed during which time the illness can exacerbate and affect children's quality of life. Understanding and treating mental disease at the population level requires early diagnosis and identification [4]. The emergence of electronic health records (EHRs) by the Health Information Technology Act Rights (OCR), 2009, has made it possible to use computer analysis of enormous data sets to address clinical concerns that were previously insurmountable [8].

It's uncertain precisely what causes major depressive disorder in children. Most likely an integration of biological, psychological, environmental, and hereditary variables. A child's social, academic, and family functioning are all significantly affected

by a family history of depression [1], trauma, chronic illness, and stressful life events. It could result in relationship issues, academic difficulties, and in extreme situations, suicidal thoughts or deeds [9]. Children diagnosed with major depressive disorder (MDD) sometimes co-occur with other mental health disorders such as anxiety disorders, ADHD, or drug misuse problems. Kids with MDD must have a solid support network. This includes guardians, educators, parents, and mental health specialists who can offer comprehension, direction, and a secure environment for the youngster to communicate their emotions and depression [10].

2.1 Problem Statement

A major depressive in children episode is usually associated with difficulties at home, at school, and in social settings. The research conducted in high-risk, community, and clinical settings has demonstrated that the typical duration of an early-onset MDD episode is six to nine months. There is evidence that the frequency, and severity of risk patients are highly associated with suicidality [11], the stigma related to mental health in India, which has become a serious issue because of the limited access to mental health services, especially in rural regions. The stigma associated with mental health issues can hinder accurate reporting and diagnosis of MDD in children. There are fewer screening questionnaires used in pediatric populations compared to adults [6].

Diagnosis of major depressive disorder (MDD) may be made if five or more of the primary symptoms are listed in Table 1, it is necessary to investigate the efficacy of therapy. The effectiveness of therapeutic approaches in alleviating depression in children and adolescents remains poorly supported by research [6]. Clinical investigation is complicated due in part to financial constraints, disagreements over plausible hypotheses, and methodological difficulties. Several factors, including high study dropout rates, reluctance to accept placebos, and concerns about safety and compliance, have made administering antipsychotic and other therapeutic placebo-controlled or decreased interest or enjoyment [12]. This work intends to create a model using ML algorithms to evaluate gathered data and pre-process it for missing values since these can lead to less accurate predictions following cleaning, only those features from the feature set that have an immediate effect on the target variable are used. Subsequently, an ensemble-based learning strategy is used to raise MDD prediction reliability [13]. Individual classifier algorithms remain beset by problems including over-fitting, lack of robustness, and challenges in choosing the best parameters. However, the difficulties were solved when they were joined by average or majority voting, and uses ensemble method was built by amalgamating the base learners [13]. Children who are diagnosed with major depressive disorder (MDD) usually meet these five criteria listed in the Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition (DSM-5) (Table 2) [14].

Machine learning's appealing feature is that it requires little to no input assumption and is more amazing. According to clinical and epidemiologic research, comorbid

Table 1 Lists the MDD symptoms in adult and pediatric patients

Symptoms for the diagnosis	Symptoms specific for pediatric patients
Depressed mood [7]	Exasperated [8]
Decreased interest or sudden weight change in children [4]	Not gaining weight [8]
Has no normal sleeping pattern [15]	Substantial drop in grades [16]
Altered psychomotor activity[15]	Poor school performance[16]
Drowsiness, feeling worthlessness/inability to concentrate and think[6]	Changes in eating pattern Anorexia Anhedonia, suicidal tendency
Suicidal ideation[14]	Unexplained mood swings and, irritability

mental illnesses are present in 40–93% of children and adolescents with depression [17]. The intensity of MDD symptoms may be adversely affected by co-occurring substance use disorders [9, 18]. It is less evident how comorbidity affects the length of MDD [16].

3 Background and Systematic Literature Review

A literature survey on the prediction of major depressive disorder (MDD) in children involves reviewing existing research, studies, and articles that identify risk factors, early signs, potential tools, and articles for predicting MDD in children’s studies.

Jonathan Flint [3] has drawn on developments in neuroscience and psychology to identify Loci that contribute to the neurovegetative and cognitive changes that are central to MDD, as well as to guide how to best interpret and utilize genetic mapping results. Concepts regarding the nature of the condition, to formulate and respond to appropriate clinical queries. The study should focus on the various forms of the condition, its traits, and the most effective ways to treat each form as it is identified, along with its causes. Athreya et al. the goal of the research is to improve instruments that support clinical practice decision-making by incorporating physiological information into probabilistic models [2]. In the scoping review [4, 5, 7, 8, 13, 19–21] these methods represent a significantly novel, possibly helpful component in symptom tracking and clinical care categorized based on mental health problems such as schizophrenia, bipolar disorder, anxiety and depression, posttraumatic stress disorder, and mental health problems among children [5]. Used 5 machine learning techniques for prediction. Digital methods that can analyze the symptoms of depression and disorders in kids and teenagers were examined. In some studies, various ensemble classifier techniques such as random forest, Xg Boost, Ada boost, and random forest show better performance compared to a single model [16]. Random forest to make a decision system. XgBoost is a gradient-boosted decision tree that aims to increase the speed and performance of the algorithm. Ada boost known

Table 2 Summary of systematic survey on MDD since 2003

No.	Survey title	References	Year	Journal	Content
1	The genetic basis of major depressive disorder[3]	Flint [3]	2023	Molecular Psychiatry	Evaluation of genetic dissection of MDD [3]
2	Evidence for machine learning guided early prediction of acute outcomes in the treatment of depressed children and adolescents with antidepressants [2]	Athreya et al. [2]	2022	Journal of Child Psychology and Psychiatry	Review on treatment of depression in children and adolescents [2]
3	Mental health prediction using machine learning: taxonomy, applications, and challenges [7]	Chung and Teo [7]	2022	Hindawi Applied Computational Intelligence and Soft Computing	Exploration on the prognosis of mental health issues [7]
4	Assessment and prediction of depression and anxiety risk factors in schoolchildren [4]	Qasrawi et al. [4]	2022	JMIR Formative Research	ML methods to forecast the risk factors linked to anxiety and depression in students [4]
5	Predictive modeling of depression and anxiety using electronic health records [8]	Nemesure et al. [8]	2021	Scientific Reports	Predictive performance for the application of machine learning based on EHR data [8]
6	Stress, anxiety and depression prediction using ensemble techniques	Joshi and Jain [13]	2021	International Research Journal of Modernization in Engineering	Review on Ensemble classifier techniques to show which is a better ensemble model [13]
7	Machine Learning Model for Computer-Aided Depression Screening among Young Adults Using Wireless EEG Headset	Sakib et al. [19]	2023	Hindawi Applied Computational Intelligence and Soft Computing	ML-based method for detecting depression among young adults [19]
8	Detection of child depression using machine learning methods	Haque et al. [5]	2021	PLoS One	Evaluation of child depression using machine learning method [5]

(continued)

Table 2 (continued)

No.	Survey title	References	Year	Journal	Content
9	Realizing a stacking generalization model to improve the prediction accuracy of major depressive disorder in adults	Mahendran et al. [16]	2020	IEEE Access	An ensemble-based ML model for prediction of MDD in Adults [16]
10	Assessment of anxiety, depression, and stress using machine learning models	Kumar et al. [22]	2020	Elsevier	Review on psychological disorder using ML models
11	Developing and validating a prediction model of adolescent MDD in the offspring of depressed parents	Stephens et al. [1]	2023	Journal of Child Psychology and Psychiatry	Prediction model of adolescent MDD in the offspring of depressed parents [1]
12	Detecting depression using an ensemble classifier based on quality of life scales	Tao et al. [12]	2020	Brain Informatics	Ensemble binary classifier is proposed to analyze health survey data [12]
13	Improving diagnosis of depression with XGBOOST ML model and a large biomarkers Dutch dataset [23]	Sharma and Verbeke [23]	2020	Open Access	To analyze the detection of depression cases among Dutch citizens dataset [23]
14	Depression recognition using machine learning methods with different feature generation strategies [24]	Li et al. [24]	2019	Elsevier	To recognize depression using the transformation of EEG features and machine learning methods [24]
15	A deep learning ensemble to classify anxiety, depression, and their comorbidity from texts of social networks [25]	Souza et al. [25]	2021	Journal of Information and Data Management	Stacking ensemble techniques to identify depression, anxiety, and comorbidity, using data extracted from Reddit [25]
16	Depression in children and adolescents: A review of Indian studies [6]	Grover et al. [6]	2019	Frontiers	To evaluate the Indian research on depression in children and adolescents [6]

(continued)

Table 2 (continued)

No.	Survey title	References	Year	Journal	Content
17	Predicting symptoms of depression and anxiety using smartphones and wearable data [14]	Moshe et al. [14]	2021	Moshe et al. [14]	To analyze how wearable devices could predict symptoms of depression and anxiety [14]
18	Cognitive science and artificial intelligence: simulating the human mind and its complexity [26]	Naveed Uddin et al. [26]	2019	IET Journals	An evaluation of simulating the human mind and its complexity [26]

as adaptive boosting is a classification meta-algorithm that is also used to improve performance. Here the Accuracy is 98.2, Recall 93.2, and Precision 93.5, and also includes more statistical classification to make the stacked algorithm stronger [1, 12, 16, 22–25]. Nemesure et al. [8] identification of GAD and MDD using EHR data has been investigated in this work. 59 characteristics in all, including continuous, ordinal, and binary variables, were employed. The features were as follows: learning problems, trouble memorization of teachings, gender, age (4 levels: under 18, 18, 19, above 20), topic of study, year of university, and French nationality. AUCs for MDD and GAD were 0.67. One is that the population restricts the generalizability of the results, therefore it’s possible that not all patients within the community were screened for MDD and GAD.

The mental health of children and adolescents in India is not well studied also there are no statistics on the course and results of depression in Indian kids and adolescents [6]. Moshe et al. smartphones and wearable technology may provide a unique source of data to identify changes in risk factors linked to mental diseases on a moment-by-moment basis. The current research is unable to clarify the direction of the association between symptoms of the behavior of stress captured by wearable and smartphone technology (Fig. 1) [14].

Naveed Uddin [26] cognitive science encompasses the interdisciplinary in the subject of artificial intelligence. Research from both the past and the present has been emphasized to guarantee a deeper comprehension of the subject. It emerges as the recent development of macro-modeling has been perceived to be beneficial in capturing a broader perspective or feature of human behavior. Mohammadi and Moradi [27] investigated to determine whether brain activity complexity and functional connectivity can predict depression severity (as measured by Beck Depression Inventory-II scores). Cai et al. [28] this review aims to discover and validate the EEG properties of patients with depression and normal participants. To further elucidate the applications of EEG data processing, future research endeavors ought to concentrate on an array of supplementary feature selection techniques.

Addressing depression in adolescents is still an essential clinical and public health concern. In order to determine the most effective treatments for adolescents and

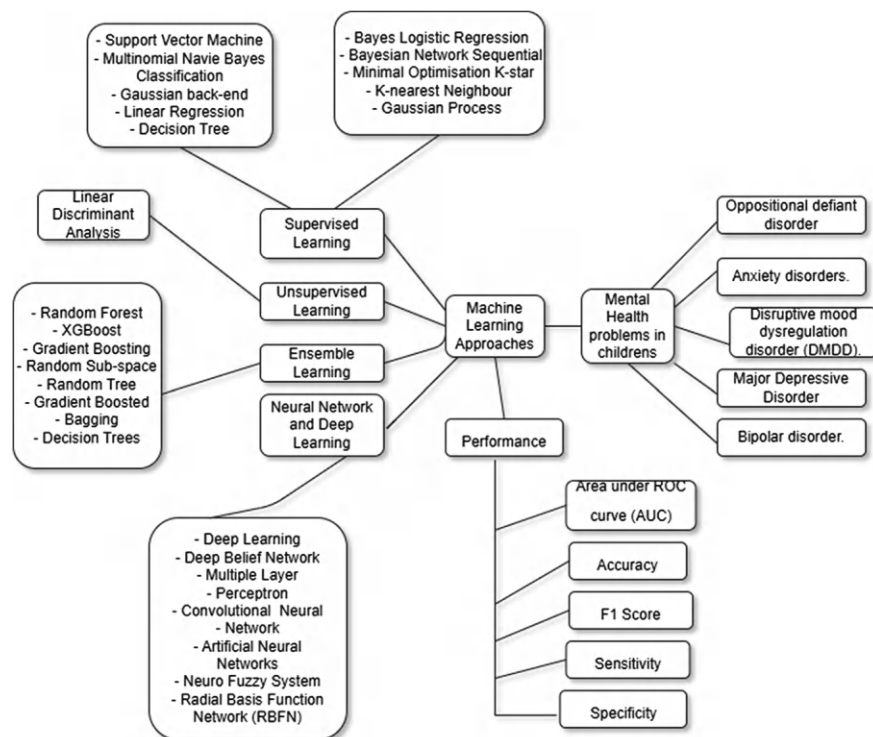


Fig. 1 Taxonomy on the review for MDD

children with major depressive disorder (MDD), an evaluation is necessary due to the significant morbidity and mortality associated with this condition. More research based on severity aspects that affect treatment outcomes may lead to the identification of effective treatments and long-lasting results for adolescents and children with MDD [10, 11, 17, 18, 29–31].

Some studies have explored neurobiological markers such as abnormalities in brain structure, neurotransmitter levels, and functioning of the HPA axis as potential predictors of MDD in children [9, 32, 33]

Ebert et al. [34] evaluating the 1-year incidence of MDD in fresher students and identifying factors that impact MDD frequency in a sample of students deemed representative of the institution. It's possible that methods-based just on students' self-reports won't be enough to identify pupils who are in danger, therefore it's important to take additional steps like staff development and awareness campaigns.

Mullen [35] research has examined treatment in children and adolescents, while other research has examined the possibility of side effects and severity of suicidal thoughts. The best course of treatment frequently combines antidepressant medicine with counseling. This study has not assessed how IPT-A affects suicidality.

Some studies have explored early interventions, such as cognitive-behavioral therapy (CBT) or mindfulness-based approaches, aimed at reducing the risk of MDD in susceptible children [32, 36, 37]. Bittner et al. [38] the relationships between particular childhood anxiety disorders and adolescent mental problems have not been extensively studied in longitudinal investigations of child and adolescent psychopathology examined in this review the predictive specificity of social anxiety, generalized anxiety disorder (GAD), overanxious disorder (OAD), and separation anxiety disorder (SAD).

McElroy et al. [39] the review investigates the developmental relationships between symptoms from early childhood to mid-adolescence, network analysis is used. The study should focus on increasing connectivity and suggest that symptoms could enhance other symptoms as they grow. In terms of clinical symptoms, major depressive disorder (MDD) is a varied illness, and this variation is probably mirrored in genetics. The clinical subgroups of MDD known as “melancholic depression” and “anxious depression” are gaining attention. The metabolome is made up of hundreds of tiny compounds, which can be characterized thanks to metabolomics. Although research is still in its early stages, recent findings indicate the blood metabolome may be able to help with MDD therapy choices. There is a need to examine further biomedical data that shares the same traits and is frequently used in biomedical applications [40–43].

The majority of individuals with bipolar disorder (BD) are addressed, after being initially misdiagnosed with major depressive disorder (MDD). There is uncertainty about whether MDD is an early stage of BD or comorbidity, and there is no consensus on distinct conversion estimates, which hinders the rapid diagnosis and treatment of other depressive disorders. To meet the goals established on non-communicable diseases and mental health issues, researchers from the domains of medicine and computer engineering have to collaborate to determine the variants of mental disorders in children, adolescents, and adults [15, 37, 44–48].

4 Machine Learning Techniques in MDD

Artificial intelligence (AI), creates systems with the capacity to learn from experience and improve without having explicitly developed is applied in machine learning [2]. These days, cognitive science encompasses various branches of research science from the domains of philosophy, education, artificial intelligence (AI), neurology, linguistics, and anthropology [8]. It is sometimes described as the study of every aspect of the mind. Researchers in the cognitive field examine how the brain processes, modifies and represents information in order to better understand behavior as well as intelligence [5]. The goal of cognitive psychology is to understand intelligence principles to enhance learning and cognitive comprehension. Furthermore, artificial cognition (AI) is computer systems simulating human intellect. This made it possible to collect data and apply the relevant rules to reach a firm judgment that would encourage self-correction [26]. In a variety of fields machine learning has

provided significant benefits. Several researchers can establish automated intelligent systems, offer personalized wisdom, and extract useful details from the data using AI [7].

Machine learning uses sophisticated statistical and probabilistic methods to build computers that can learn from experience [16]. It is considered to be a very helpful tool for predicting mental health. It has made several researchers extract valuable information from the data, offer individualized experiences for predicting the occurrences of disease in health care, and categorize these data using prominent techniques [4].

The most popular technique in a variety of research is supervised learning and ensemble learning, particularly when it comes to forecasting disease in the medical domain [26].

4.1 Ensemble Learning

Ensemble learning to boost a model's performance or reduce the likelihood of choosing models with inadequate results [8]. Furthermore, considering neural networks and deep learning can tackle a wide range of issues, including speech, images, and natural language processing, techniques have gained increasing respect recently among machine learning techniques. The methods allow the algorithms to learn from observations because they are based on the brain's neural networks. Machine learning algorithms have been applied to the health sector to diagnose illnesses, forecast epidemic outbreaks, find novel medications, and analyze radiological data. More powerful algorithms were developed and used to address increasingly challenging medical diagnosis issues.

Ensemble classifiers exhibit progress in providing right diagnoses, especially when several features are included. Based on predictions from separate classifiers because of its capacity for generalization, ensemble learning outperforms individual learner algorithms in terms of predictive efficiency.

4.2 Ensemble Method

Ensemble methods generate and merge several models to provide more accurate predictions. They can use various techniques, such as bagging, boosting, or stacking, to reduce variation, reduce bias, or improve accuracy [16]. They are particularly helpful for expanding classifiers and regressors in tree-based models [7]. The voting classifier of ensemble classification is shown in Fig. 2

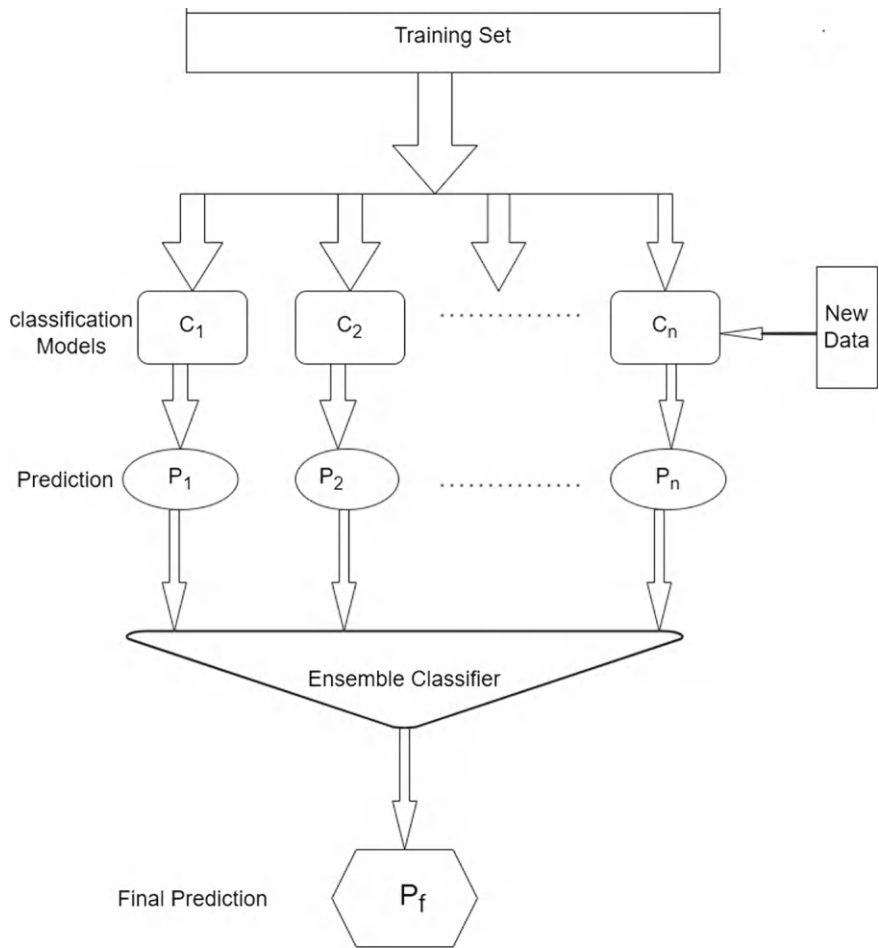


Fig. 2 Ensemble classifier

4.2.1 Bagging

Bootstrap aggregation is a straightforward yet powerful ensemble algorithm. By making copies of the training data for every training set, bagging provides variation. By using replacement sampling, it trains several base learner classifiers [16]. Bagging reduces variance and is appropriate for small datasets.

4.2.2 Boosting

Boosting is an iterative method that minimizes training errors by merging several weak learners. In contrast with bagging, which uses bootstrapped clones of the

training data with the potential for identical cases in each training set [25]. Weighted cases are used to implement the first base learner, and all instances are initially assigned uniform weights. After computing the error, the weights of instances that were incorrectly classified are increased while those of instances that were correctly classified are dropped. Boosting reduces learning process bias [16].

4.2.3 Stacking

To create a stacking generalization model, base learner aggregates all of the predictions from several distinct lower-level base learner models. Stacking intends to increase prediction accuracy. The purpose of stacking is to explore the possibilities of different approaches for a given problem [16]. This strategy depends on the concept that different models, every one of which has limitations in what it can learn, can be utilized for a learning problem. As a result, you can generate a variety of learners and employ them to create an intermediate prediction [13]. A similar target is subsequently predicted by a new model that is added after training from the intermediate predictions. The highest performing model is superior to all intermediate models and is considered to be layered on top of the others to increase total performance.

4.2.4 Random Forest

A random forest integrates the output from multiple decision trees to produce a single outcome [20]. Its widespread acceptance can be attributed to its adaptability, simplicity of usage, and capacity to handle regression as well as classification issues [5]. Random forest algorithm creates an uncorrelated forest of decision trees. The primary distinction between random forests and decision trees is feature randomness or “feature bagging”. The decision trees evaluate every conceivable split, whereas random forests only select a subset of the features that can be divided [12]. Regression and classification issues can then be resolved using the random forest classifier [20]. Each decision tree in the ensemble in the random forest technique. One-third of the training sample is designated as test data; this is referred to as the out-of-bag (OOB) sample [25]. Feature bagging is then used to introduce yet another randomization, increasing dataset variety and decreasing decision tree correlation [25].

4.2.5 Decision Trees

A decision tree, as the name suggests, is a schematic-like tree structure that uses the findings of the algorithm are represented by leaf nodes, features by branches, and features with core nodes [20]. It is an effective supervised machine learning technique that may be utilized to solve both regression as well as classification problems [20]. It's one of the most efficient algorithms out there. It's one of the most

powerful machine learning algorithms since random forest trains on different subsets of training data. It is generated by continually dividing the initial training data into subsets corresponding to the attribute values until a stopping condition—such as the minimum number of samples needed to divide a node or the maximum depth of the tree is satisfied [25].

5 Challenges and Future Implications

In children, anxiety and behavioral problems are usually indicative of mood disorders. The lack of gender differences in the findings of earlier studies implies that environmental variables have a greater frequency of significant effects on genes that are expressed later in life or during adolescence. Early intervention and prevention of mental health disorders can help reduce their incidence in young children.

In research involving children under the age of twelve, the most popular method is the collection of parental reports. Parent reports may also help with prediction judgments and treatment recommendations in the clinical diagnosis of young children. In addition to addressing the disruptive elements of their psychological disorders, the inner life and mental wellness of young children demonstrating externalizing behavior should also be the focus of attention.

In India, between 75 and 85% of depressed people from low- and middle-class backgrounds do not receive treatment for their depressive condition because of a lack of resources for children's physical health and a shortage of qualified health-care professionals. Furthermore, a database of the early causes of neuropsychiatric contributes to the comprehension of mental illnesses broadly and could help improve the forecasting, assessment, and treatment of pediatric psychiatry.

ML techniques has the potential to significantly increase the monitoring and diagnosis capabilities of healthcare systems worldwide. Future directions in child psychology research, are examined, along with future research options and shortcomings, particularly in low-resource regions.

6 Conclusion

Predicting MDD in children is a complex and multifaceted endeavor [20], involving a combination of genetic, environmental, psychological, and social factors [25]. Early detection and intervention are crucial for improving outcomes. Ongoing research in this field may lead to more accurate and effective predictive models, as well as targeted interventions for at-risk children. This paper aims to provide a focused significant summary of the gaps in the literature concerning the applications of machine learning in the field of mental health, as well as to cover recent developments in the field and identify potential directions for further investigation on the current applications of machine learning, particularly in the field of psychological wellness.

Acknowledgements The authors would like to thank the psychologist Dr. Prajeesh Palanthara and the psychologist Dr. Anil Kumar P P at Edhini Ayurvedic Hospital for their assistance in getting this survey published. Their careful recommendations assured the validity and reliability of the study and significantly raised the scope and standard of my research findings.

References

1. Stephens A, Allardyce J, Weavers B, Lennon J, Jones RB, Powell V, Rice F (2023) Developing and validating a prediction model of adolescent major depressive disorder in the offspring of depressed parents. *J Child Psychol Psychiatry* 64(3):367–375
2. Athreya AP, Vande Voort JL, Shekunov J, Rackley SJ, Leffler JM, McKean AJ et al (2022) Evidence for machine learning guided early prediction of acute outcomes in the treatment of depressed children and adolescents with antidepressants. *J Child Psychol Psychiatry* 63(11):1347–1358
3. Flint J (2023) The genetic basis of major depressive disorder. *Mol Psychiatry* 1–12
4. Qasrawi R, Polo SPV, Al-Halawa DA, Hallaq S, Abdeen Z (2022) Assessment and prediction of depression and anxiety risk factors in schoolchildren: machine learning techniques performance analysis. *JMIR Form Res* 6(8):e32736
5. Haque UM, Kabir E, Khanam R (2021) Detection of child depression using machine learning methods. *PLoS One* 16(12):e0261131
6. Grover S, Raju VV, Sharma A, Shah R (2019) Depression in children and adolescents: a review of Indian studies. *Indian J Psychol Med* 41(3):216–227
7. Chung J, Teo J (2022) Mental health prediction using machine learning: taxonomy, applications, and challenges. *Appl Comput Intell Soft Comput* 2022:1–19
8. Nemesure MD, Heinz MV, Huang R, Jacobson NC (2021) Predictive modeling of depression and anxiety using electronic health records and a novel machine learning approach with artificial intelligence. *Sci Rep* 11(1):1980
9. Zhang X, Wang R, Sharma A, Deverajan GG (2021) Artificial intelligence in cognitive psychology—influence of literature based on artificial intelligence on children’s mental disorders. *Aggress Violent Behav* 101590
10. Sequeira L, Perrotta S, LaGrassa J, Merikangas K, Kreindler D, Kundur D et al (2020) Mobile and wearable technology for monitoring depressive symptoms in children and adolescents: a scoping review. *J Affect Disord* 265:314–324
11. Roza SJ, Hofstra MB, Van Der Ende J, Verhulst FC (2003) Stable prediction of mood and anxiety disorders based on behavioral and emotional problems in childhood: a 14-year follow-up during childhood, adolescence, and young adulthood. *Am J Psychiatry* 160(12):2116–2121
12. Tao X, Chi O, Delaney PJ, Li L, Huang J (2021) Detecting depression using an ensemble classifier based on Quality of Life scales. *Brain Inf* 8:1–15
13. Joshi D, Jain M. Stress, anxiety and depression prediction using ensemble techniques
14. Moshe I, Terhorst Y, Opoku Asare K, Sander LB, Ferreira D, Baumeister H et al (2021) Predicting symptoms of depression and anxiety using smartphone and wearable data. *Front Psych* 12:625247
15. Sharaev M, Artemov A, Kondratyeva E, Sushchinskaya S, Burnaev E, Bernstein A et al (2018) MRI-based diagnostics of depression concomitant with epilepsy: in search of the potential biomarkers. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). IEEE, pp 555–564
16. Mahendran N, Vincent PDR, Srinivasan K, Sharma V, Jayakody DK (2020) Realizing a stacking generalization model to improve the prediction accuracy of major depressive disorder in adults. *IEEE Access* 8:49509–49522

17. Deshpande M, Rao V (2017) Depression detection using emotion artificial intelligence. In: 2017 international conference on intelligent sustainable systems (ICISS), pp 858–862. IEEE
18. Zhu T, Jiang J, Hu Y, Zhang W (2022) Individualized prediction of psychiatric readmissions for patients with major depressive disorder: a 10-year retrospective cohort study. *Transl Psychiatry* 12(1):170
19. Sakib N, Islam MK, Faruk T (2023) Machine learning model for computer-aided depression screening among young adults using wireless EEG headset. *Comput Intell Neurosci*
20. Abou-Warda H, Belal NA, El-Sonbaty Y, Darwish S (2017) A random forest model for mental disorders diagnostic systems. In: *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016*, vol 2. Springer International Publishing, , pp 670–680
21. Reddy US, Thota AV, Dharun A (2018) Machine learning techniques for stress prediction in working employees. In: 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp 1–4. IEEE
22. Kumar P, Garg S, Garg A (2020) Assessment of anxiety, depression and stress using machine learning models. *Proc Comput Sci* 171:1989–1998
23. Sharma A, Verbeke WJ (2020) Improving diagnosis of depression with XGBOOST machine learning model and a large biomarkers Dutch dataset (n = 11,081). *Front Big Data* 3:15
24. Li X, Zhang X, Zhu J, Mao W, Sun S, Wang Z et al (2019) Depression recognition using machine learning methods with different feature generation strategies. *Artif Intell Med* 99:101696
25. Souza V, Nobre J, Becker K (2021) A deep learning ensemble to classify anxiety, depression, and their comorbidity from texts of social networks. *J Inf Data Manage* 12(3)
26. Naveed Uddin M (2019) Cognitive science and artificial intelligence: simulating the human mind and its complexity. *Cogn Comput Syst* 1(4):113–116
27. Mohammadi Y, Moradi MH (2021) Prediction of depression severity scores based on functional connectivity and complexity of the EEG signal. *Clin EEG Neurosci* 52(1):52–60
28. Cai H, Chen Y, Han J, Zhang X, Hu B (2018) Study on feature selection methods for depression detection using three-electrode EEG data. *Interdiscip Sci Comput Life Sci* 10:558–565
29. Milin R, Walker S, Chow J (2003) Major depressive disorder in adolescence: a brief review of the recent treatment literature. *Can J Psychiatry* 48(9):600–606
30. Rois R, Ray M, Rahman A, Roy SK (2021) Prevalence and predicting factors of perceived stress among Bangladeshi university students using machine learning algorithms. *J Health Popul Nutr* 40:1–12
31. Uchida M, Fitzgerald M, Woodworth H, Carrellas N, Kelberman C, Biederman J (2018) Subsyndromal manifestations of depression in children predict the development of major depression. *J Pediatr* 201:252–258
32. Gao S, Calhoun VD, Sui J (2018) Machine learning in major depression: from classification to treatment outcome prediction. *CNS Neurosci Ther* 24(11):1037–1052
33. Behzad R, Behzad A (2021) The role of EEG in the diagnosis and management of patients with sleep disorders. *J Behav Brain Sci* 11(10):257–266
34. Ebert DD, Buntrock C, Mortier P, Auerbach R, Weisel KK, Kessler RC et al (2019) Prediction of major depressive disorder onset in college students. *Depress Anxiety* 36(4):294–304
35. Mullen S (2018) Major depressive disorder in children and adolescents. *Mental Health Clin* 8(6):275–283
36. Lowe SR, Quinn JW, Richards CA, Pothen J, Rundle A et al (2016) Childhood trauma and neighborhood-level crime interact in predicting adult posttraumatic stress and major depression symptoms. *Child Abuse Negl* 51:212–222
37. Nestsiarovich A, Reps JM, Matheny ME, DuVall SL, Lynch KE, Beaton M et al (2021) Predictors of diagnostic transition from major depressive disorder to bipolar disorder: a retrospective observational network study. *Transl Psychiatry* 11(1):642
38. Bittner A, Egger HL, Erkanli A, Jane Costello E, Foley DL, Angold A (2007) What do childhood anxiety disorders predict? *J Child Psychol Psychiatry* 48(12):1174–1183
39. McElroy E, Fearon P, Belsky J, Fonagy P, Patalay P (2018) Networks of depression and anxiety symptoms across development. *J Am Acad Child Adolesc Psychiatry* 57(12):964–973

40. Liu Y, Yieh L, Yang T, Drinkenburg W, Peeters P, Steckler T et al (2016) Metabolomic biosignature differentiates melancholic depressive patients from healthy controls. *BMC Genom* 17:1–17
41. Braithwaite EC, O'Connor RM, Degli-Esposti M, Luke N, Bowes L (2017) Modifiable predictors of depression following childhood maltreatment: a systematic review and metaanalysis. *Transl Psychiatry* 7(7):e1162–e1162
42. Pampouchidou A, Simos PG, Marias K, Meriaudeau F, Yang F, Pediaditis M, Tsiknakis M (2017) Automatic assessment of depression based on visual cues: a systematic review. *IEEE Trans Affect Comput* 10(4):445–470
43. Kieling C, Buchweitz C, Caye A, Manfro P, Pereira R, Viduani A et al (2021) The identifying depression early in adolescence risk stratified cohort (IDEA-RiSCo): rationale, methods, and baseline characteristics. *Front Psych* 12:697144
44. Alvaro PK, Roberts RM, Harris JK (2014) The independent relationships between insomnia, depression, subtypes of anxiety, and chronotype during adolescence. *Sleep Med* 15(8):934–941
45. Grządzielewska M (2021) Using machine learning in burnout prediction: a survey. *Child Adolesc Soc Work J* 38(2):175–180
46. Laijawala V, Aachaliya A, Jatta H, Pinjarkar V (2020) Mental health prediction using data mining: a systematic review. In: *Proceedings of the 3rd International Conference on Advances in Science and Technology (ICAST)*
47. Ferrari AJ, Charlson FJ, Norman RE, Flaxman AD, Patten SB, Vos T, Whiteford HA (2013) The epidemiological modelling of major depressive disorder: application for the Global Burden of Disease Study 2010. *PloS One* 8(7):e69637
48. Sau A, Bhakta I (2019) Screening of anxiety and depression among seafarers using machine learning technology. *Inf Med Unlocked* 16:100228

Lie Detection Using Audio Classification



Tejas Ravindra Rote, Shivama, and M. Narayanamoorthi

Abstract This study explores lie detection by focusing on sound. By using real-world courtroom data changed into sound, a Convolutional Neural Network (CNN) model is used to find important features from sound data that help in spotting lies. Unlike traditional methods that need manual work to identify features, the CNN model does this work on its own and shows a high accuracy of 93.5%. This improvement is not only good for finding lies in sound but also helps in improving the performance of systems that use multiple types of data for lie detection. This makes it a strong and reliable tool for real-life courtroom scenarios. The results show that CNN is very good at handling the complex structures of sound data, moving us closer to better lie detection methods. Moreover, the study points out the need for more real and diverse data to help further improve lie detection. This will help in enhancing trust and safety in different areas.

Keywords Deception detection · Speech analysis · Verbal cues · Machine learning · Audio modality · Audio data analysis

1 Introduction

Unveiling the truth has always been at the core of human interaction, particularly in judicial proceedings where the stakes are high and the consequences far-reaching. Traditional mechanisms to detect deception, such as polygraph tests, have been met with skepticism due to their invasive nature, high costs, and potential inaccuracies. Furthermore, these methods necessitate specialized equipment and expert personnel, escalating operational expenses. As technology advances, there emerges

T. R. Rote · Shivama (✉)

Department of Software Systems, Vellore Institute of Technology, Vellore, India

e-mail: shivama.2020@vitstudent.ac.in

M. Narayanamoorthi

Department of IoT, Vellore Institute of Technology, Vellore, India

an opportunity to explore non-invasive, cost-effective, and potentially more accurate alternatives.

The focal point of this research is the examination of speech and verbal cues, a domain relatively uncharted in the context of lie detection. Speech, an intrinsic aspect of human communication, may inadvertently betray underlying intentions or emotions. Through comprehensive analysis of audio data from real-world trials, this study endeavors to uncover speech characteristics that may diverge during deceptive statements. The overarching ambition is to forge a lie detection system adept at distinguishing truth from falsehood-based purely on the uttered words.

The relevance and feasibility of such a venture are manifold. Technologically, the surge in capabilities within natural language processing and machine learning equips us with sophisticated tools for in-depth speech analysis. From an economic perspective, a speech-based system could circumvent the need for pricey apparatus and training, potentially positioning it as a cost-effective alternative. On the societal front, if realized successfully, this system could augment the credibility of judicial procedures, potentially resulting in more precise verdicts and reinforcing public confidence in legal establishments. Environmentally, a software-driven solution aligns with contemporary sustainability objectives, leveraging existing computer infrastructures without necessitating additional resources. The political implications are profound as well; a system with the potential to diminish wrongful convictions might garner widespread support, though it would be imperative to institute robust ethical safeguards. Finally, the adaptability of such technology to a myriad of languages and dialects underscores its global relevance.

2 Literature Review

In the developing field of audio classification, researchers have explored various paradigms to enhance performance and applicability. Camara et al. [1] delved into the domain of lie detection, uncovering that high-stakes lies are considerably more intricate to falsify and discern. However, the constrained sample size and participant diversity pose limitations to the generalizability of their findings. In a parallel vein, Chen et al. [2] developed a novel architecture for audio classification, demonstrating promising outcomes, albeit the study is limited by its exclusive evaluation of the proposed architecture.

The explorative work by Stergiou and Damen [3] brought forth an innovative approach to audio retrieval by employing an iterative attention mechanism. Despite refining predictions and enhancing performance, the approach is marked by its computationally intensive nature, suggesting a tradeoff between accuracy and computational efficacy. Liu et al. [4] proposed a simplified pooling frontend method utilizing max pooling in audio classification. While offering valuable insights into utilizing simple methodologies, the findings may be limited to specific datasets.

Furthermore, Gazneli et al. [5] delved into the architectural facet of audio classification networks, advocating for the utilization of end-to-end methodologies. Nevertheless, the research focused on a limited scope of audio classification tasks, pinpointing a need for expanded research.

Shaar et al. [6] pivoted toward detecting previously fact-checked claims, proposing a method that while robust, leans heavily on the existence of a database of fact-checked claims, thus presenting a limitation in scenarios where such a database is unavailable or incomplete. On a related note, Kiran et al. [7] elucidated the nuances of lie detection through analyzing facial expressions, highlighting the potential in assessing styles and response times. However, the study is bound by its exclusive focus on body expressions and movements, lacking a holistic evaluation of deceptive behavior.

Rodriguez-Diaz et al. [8] deployed a machine learning-based lie detector in a gaming context, an approach that, while innovative, is potentially marred by the introduction of racial bias and lacks generalizability due to its reliance on video data. In a distinct approach, Salunkhe [9] explored the utilization of Attention-based Bidirectional LSTM for deception detection in reviews, uncovering that while classification algorithms such as Random Forest, XGBoost, and LSTM networks showcase efficacy, the study is restrained by its application to a specific dataset and may not be universally applicable. Moreover, Li and Li [10] proposed the ATST model for audio representation learning, demonstrating superior performance against state-of-the-art methods in certain audio classification tasks, albeit within a limited audio type.

Zhao et al. [11] explored the intriguing nexus between audio and text, developing a method that adeptly transfers knowledge from visual modalities to audio modalities. Nonetheless, the dependency on pre-trained visual models serves as a potential limitation, particularly in scenarios where nuanced or domain-specific visual models are requisite. Zhang et al. [12] provided an enlightening survey on machine learning techniques for audio classification, categorizing options for video and audio data, and highlighting the pivotal role of data labeling. Their work underscores the potential enhancement in accuracy achievable through combining active learning with semi-supervised learning, marking a pathway for future exploration.

Phan et al. [13] presented an improved approach for audio scene classification through label tree embedding, though the computational efficiency of the system is compromised due to the embedding. In a realm of audio-visual video parsing, Mo and Tian [14] introduced the Multimodal Grouping Network (MGN), albeit with a low accuracy rate, which restricts its practical applicability in real-world scenarios. Afouras et al. [15] ventured into self-supervised object detection from audio-visual cues, offering a methodology for training robust object detectors. However, the approach is not devoid of limitations, as it is prone to detecting objects that often appear together with the sound-making object, necessitating further refinement.

Zaheer et al. [16] introduced a generative cooperative learning approach for unsupervised anomaly detection, exploiting the infrequent nature of anomalies toward learning normal behavior in videos. However, its efficacy may be compromised in scenarios where abnormal events have not yet occurred, indicating a potential area

for enhancement. Kumari et al. [17] provided a comprehensive review of lie detection techniques, concluding that deep learning-based systems can potentially outperform polygraph tests. Nonetheless, the variability in test results due to factors like emotions, physical condition, and mental stress signifies a critical area that warrants further exploration and mitigation.

Liu et al. [18] explored unsupervised temporal video grounding, proposing the Deep Semantic Clustering Network which, while innovative, brings forth the challenges associated with unsupervised learning, such as unpredictability in results. Krishnamurthy et al. [19] utilized a deep learning approach for multimodal deception detection, amalgamating video, audio, and text modalities. Despite its ingenuity, the approach is noted to be time-intensive and complex, and potentially susceptible to errors in practical, dynamic environments. Mittal et al. [20] delved into an emotion-based method for detecting real and fake emotions, leveraging the similarity between audio-visual modalities for classification. However, the approach is marred by a high misclassification rate and mismatches, pinpointing a crucial need for refinement.

Bareeda et al. [21] delved into lie detection using speech processing techniques, suggesting a novel method for identifying stress in a speaker's voice as an indicator of deceit. Despite achieving an overall classification accuracy of 81%, the method demonstrates a considerable margin for enhancement, especially concerning its precision. Srivastava and Dubey [22] employed artificial neural networks for deception detection, utilizing a variety of physical features like Mel-Frequency Cepstrum Coefficients and pitch. However, obtaining accurate measurements of these features proves challenging, signaling a potential area for further research and development.

Marcoola et al. [23] proposed a lie speech classification approach using voice features, where Model 1 achieved an accuracy of 72.5% but was hindered by low precision levels, indicating a pivotal need for model refinement. AlArfaj and Hosni [24] developed a deep learning model for EEG-based lie detection, achieving a precision of 70% and consequently, a low F1 Score, revealing a necessity for enhancing the model's precision and reliability. Ramadhan et al. [25] focused on sentiment analysis from movie reviews using the SVM classification model, which, while providing insight into public opinion, showcased a tendency toward lower precision, emphasizing the need for improved classification models.

Bhamare et al. [26] introduced a deep learning model for lie detection, utilizing Sklearn's gradient booster to evaluate feature importance, revealing a decrement in model performance when an interviewee's eyes are closed, thus highlighting an avenue for further refinement. Didi et al. [27] embarked on a unique journey of COVID-19 tweet classification, asserting that user sentiments could be analyzed with the aid of a hybrid model based on BERT and SVM. However, their work is bound by its reliance on global English language tweets, suggesting a need for multilingual or language-independent approaches.

Javaid et al. [28] proposed a multimodal lie detection system, combining EEG, audio, and video modalities, which reached its peak performance when all modalities were integrated. The bidirectional LSTM, however, demonstrated suboptimal performance on sequence classification tasks, opening a window for exploring alternative models or approaches. Altalbe [29] explored audio fingerprint analysis for speech

processing through a deep learning method, albeit with a notably small sample size and dataset, indicating a potential lack of generalizability and robustness in the findings.

Venkatesh et al. [30] employed deep recurrent convolutional networks on videos for deception detection. However, the use of video data precipitated issues related to racial and gender bias, necessitating an exploration into more unbiased and ethical methodologies. Lastly, Ali et al. [31] focused on hate speech detection on Twitter, utilizing models like BERT and xlmroberta, and while achieving notable results, they too were constrained by the sole use of English language tweets.

3 Methodology

The methodology, illustrated in Fig. 1, harnesses the potency of both traditional and Convolution Neural Network (CNN) models on trial proceedings videos categorized as “deceptive” or “truthful”. The pathway encompasses a thorough initial data visualization via wave-plots and spectrograms, followed by rigorous data pre-processing, wherein audio, extracted and segmented from videos, and is also augmented to enhance model generalization. Pivotal features such as chroma STFT, ZCR (Zero Crossing Rate), MFCC, Mel spectrogram, and RMS value are extracted to feed into the model training stage, which introduces a pipeline of classical models and a specially crafted 4-layer CNN model. The latter, incorporating a Global Average Pooling layer and batch normalization, is strategically designed to mitigate overfitting and adeptly handle audio classification. A comprehensive model evaluation, utilizing metrics like F1 Score and Accuracy, culminates the methodology, offering a comparative lens between traditional models and our proposed CNN, highlighting the latter’s nuanced applicability and efficacy in the domain of audio-based deception detection.

3.1 Data Pre-processing

The initial dataset comprises 120 videos, out of which 120 videos—60 deceptive and 60 truthful—are selected for analysis, ensuring a balanced dataset to mitigate potential biases in subsequent model training as illustrated in Table 1. The first step in our data pre-processing involves extracting audio from these videos, providing a base for further granular analysis and processing.

To achieve manageability and foster a more detailed examination, the extracted audio clips are segmented into smaller audio chunks, culminating in a total of 1298 clips, comprising 652 truthful and 646 deceptive clips. This segmentation ensures that the models are exposed to varied instances of data, facilitating a detailed learning process.

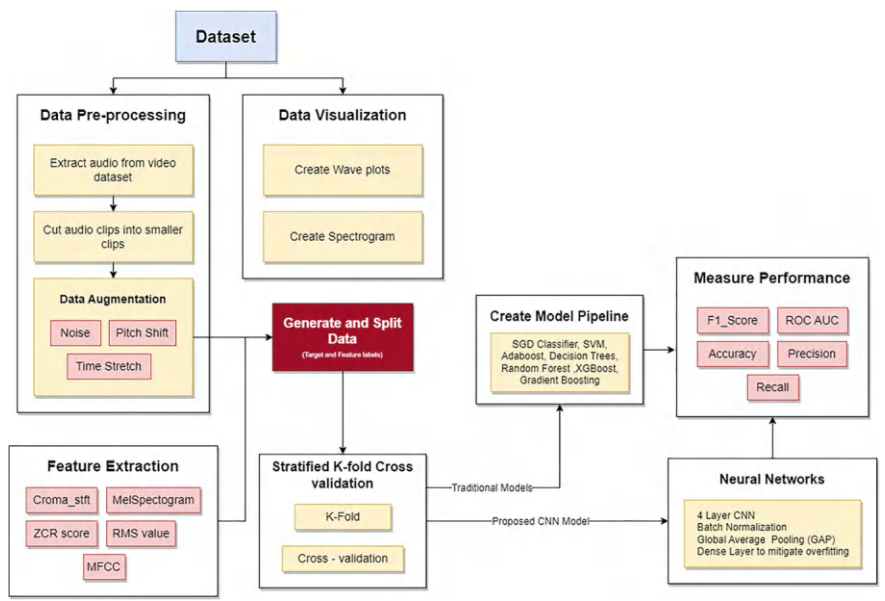


Fig. 1 Overview of proposed model

Table 1 Dataset description

Detail	Deceptive	Truthful	Total Clips
Original dataset	0	60	120
After pre-processing	652	647	1299
After data augmentation	1956	1941	3897

In order to enhance the robustness and generalizability of our models, data augmentation is employed on the audio clips. This step is crucial to ensure that our models are exposed to a varied and rich dataset, which encompasses different possible scenarios and variations within the audio. Each clip in the dataset is subjected to a series of augmentation techniques, namely:

Noise Injection: Artificial noise is added to the original audio clips to simulate possible real-world scenarios where background noise might be present. Time Stretching: The original audio clips are time-stretched, altering the speed without changing the pitch. Pitch Shifting: The pitch of the audio clips is modified, simulating different vocal tonalities and variances that might be encountered. These augmented audio clips are then appended back to the dataset, amplifying our data size to approximately 3200 clips, thereby ensuring a comprehensive and varied dataset for model training. The augmented data not only aids in improving model resilience to varied data scenarios but also aims to prevent over-fitting by providing a more generalized learning environment.

3.2 Feature Extraction

The feature extraction process is important in our methodology, focusing on distilling comprehensive and relevant audio features that are deemed pivotal for identifying deceptive speech. The extracted features encapsulate distinct characteristics of the audio signal, making them crucial indicators for differentiating between deceptive and truthful speech. These features include Zero Crossing Rate (ZCR), Chroma Short Time Fourier Transform (Chroma STFT), Mel-Frequency Cepstral Coefficients (MFCC), Mel Spectrogram, and Root Mean Square (RMS) value.

- **Zero Crossing Rate (ZCR):** ZCR measures the rate at which the signal shifts from positive to negative or back. In the context of speech deception, variations in ZCR could potentially be indicative of stress or nervousness, often subtly altering the speaker's voice pitch and tone.
- **Chroma STFT:** Chroma features, highlighting the twelve different pitch classes, are pivotal for describing harmony in an audio signal. The subtle manipulations in pitch and tone, potentially stemming from the psychological stress of deception, might be effectively captured through Chroma STFT.
- **MFCC:** Representing the short-term power spectrum of a sound, MFCCs can potentially highlight nuanced alterations in the speaker's vocal tract, revealing underlying emotional states or intentional vocal modulation often associated with deceptive speech.
- **Mel Spectrogram:** As it represents the spectrum of frequencies in a sound as they are perceived, the Mel Spectrogram can provide insights into the variability and modulations in speech patterns, which might differ when a person is being deceptive versus speaking truthfully.
- **RMS Value:** Capturing the magnitude of the audio signal, the RMS value could unveil variations in speech intensity and loudness, which might be subconsciously modulated during deceptive speech.

Two functions, *extract_features()* and *get_features()*, meticulously facilitate the feature extraction process. *extract_features()* is designed to compute and return the aforementioned audio features from a specific audio clip. Meanwhile, *get_features()* employs *extract_features()* first on the original audio clips, followed by its application on the augmented audio data, ensuring a consistent and rich feature set across the entirety of the dataset.

Through this rigorous feature extraction, the models are endowed with a rich, informative feature set, enhancing their ability to discern and learn patterns associated with deceptive and truthful speech, potentially unmasking the subtle vocal nuances that characterize deceit.

3.3 Model Training

Once vital features were extracted, our methodology transitioned into a carefully planned phase of data organization, ensuring optimal structure for model training. A meticulously constructed Data-Frame was employed to house the extracted features, providing a systematic and accessible format for model consumption. This Data-Frame scrupulously cataloged the extracted feature values for each audio clip while also designating them as “truthful” (0) or “deceptive” (1), ensuring alignment with their original labels. Thus, every row represented an audio clip through its feature values and corresponding authenticity label, presenting a comprehensive data panorama. In recognizing the potential challenges of model training with constrained data and the looming peril of over-fitting, our methodology wisely incorporated Stratified K-Fold Cross-Validation. This technique ensured each fold was representative of the entirety, maintaining the ratio between deceptive and truthful instances, and safeguarding against model bias toward specific data subsets. This approach, especially crucial in a domain where data is both vital and scarce, ensured a judicious use of available data, making certain that model training and evaluation were robust and strategically attuned to the nuanced characteristics of deceptive speech.

In the training phase, our approach integrates a well-designed model pipeline to compare with the proposed Con-Fig. 2: Overview of Neural network convolutional Neural Network (CNN) model. The pipeline combines several traditional classification models, including Support Vector Machines (SVM), XGBoost, Gradient Boosting, Random Forest, Decision Tree, Stochastic Gradient Descent (SGD) Classifier, and AdaBoost. Rather than utilizing the models in their base form, each is strategically tuned using the Grid Search technique to best navigate the complexities of deceptive speech detection. This isn’t just a quest for accuracy but an effort to unlock parameter configurations that enhance each model’s predictive capabilities. After training, models undergo a thorough evaluation using metrics such as F1 Score, Recall, Precision, Accuracy, and ROC-AUC, providing a comprehensive view of model performance beyond mere accuracy. This stage concludes with a suite of finely tuned traditional models, establishing a benchmark against which the proposed CNN model is compared, enabling a detailed, data-driven comparative analysis in the endeavor to decipher deceptive speech.

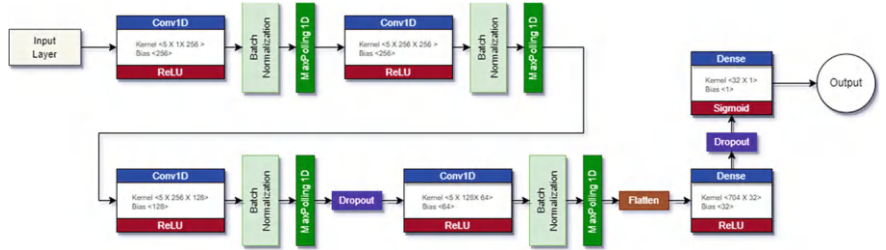


Fig. 2 Overview of neural network

The proposed Convolution Neural Network (CNN) model diverges notably from traditional models, meticulously engineered to decode spatial hierarchies in our data, thereby exposing intricate patterns essential for discerning deceptive speech. As illustrated in Fig. 2, the model unfolds through multiple layers, each systematically designed to extract information from the audio features at various scales and dimensions:

- **Convolution Layers:** Four layers, utilizing a kernel size of 5 and Re-LU activation, scan the input features, extracting patterns potentially indicative of deceptive vocal nuances, progressively transitioning from granular to abstract feature extraction as layer units reduce from 256 to 32.
- **Batch Normalization:** Incorporated within convolution layers, batch normalization stabilizes and hastens training by minimizing internal co-variate shift, ensuring smooth navigation through the parameter space and potentially avoiding suboptimal local minima.
- **Max-Pooling Layers:** These layers sub-sample the extracted feature maps post-convolution, preserving crucial features while reducing computational complexity and minimizing over-fitting by providing translation in-variance.
- **Dropout Layers:** Strategically positioned to combat over-fitting, these layers randomly nullify a fraction of input units during training, preventing excessive co-adaptation of hidden units.
- **Global Average Pooling (GAP) Layer:** Bridging convolution and dense layers, the GAP layer condenses spatial dimensions, converting feature maps to scalars and concurrently mitigating over-fitting by minimizing parameters forwarded to dense layers.
- **Dense Layers:** A 32-unit dense layer with Re-LU activation synthesizes distilled information from preceding layers to formulate higher-level features.
- **Output Layer:** The architecture culminates with an output layer employing sigmoid activation, ensuring bounded output interpreted as the probability of deception presence in the audio clip.

3.4 Model Evaluation

In assessing the performance of our Convolution Neural Network (CNN) model for audio-based deception detection, we employed a suite of evaluation techniques to gauge its effectiveness.

Firstly, accuracy, a fundamental metric, was calculated by dividing the number of correctly classified instances by the total number of instances. This metric offers a broad understanding of overall classification correctness. Secondly, precision, computed as the ratio of true positives to the sum of true positives and false positives, provides insight into the model's ability to make accurate positive predictions.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (1)$$

On the other hand, recall, determined as the ratio of true positives to the sum of true positives and false negatives, assesses the model's capacity to identify all actual positive instances.

$$\text{Recall} = \text{True Positive} / (\text{True Positive} + \text{False Negative}) \quad (2)$$

These two metrics, precision and recall, together form the foundation for the F1 Score, which strikes a balance between the precision and recall, offering a comprehensive assessment of model performance.

$$\text{F1 score} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall}) \quad (3)$$

Additionally, the Receiver Operating Characteristic-Area Under the Curve (ROC-AUC) score was utilized to measure the model's ability to distinguish between classes across various thresholds, thereby shedding light on its discrimination capability. Each of these evaluation techniques plays a distinct role: accuracy provides an overall measure of correctness, precision highlights the model's positive predictive power, recall gauges its ability to uncover true positives, and ROC-AUC signifies its capacity for classification discrimination. Together, they offer a comprehensive assessment of our model's proficiency in audio-based deception detection.

4 Result and Discussion

In an intricate domain such as deception detection, leveraging potent machine learning methodologies is paramount. The comparative analysis between the employed Convolution Neural Network (CNN) model and methodologies from research papers "Deception Detection using Real-life Trial Data" (paper A) and "Bag-of-Lies: A Multimodal Dataset for Deception Detection" (paper B) shines a light on potential enhancements and optimizations in utilizing machine learning for deception detection.

The CNN model's superior performance, evinced by a notable accuracy of 93.5%. As shown in Table 2, unlike traditional classifiers, CNN autonomously discerns and learns critical features through its convolution and pooling layers, eliminating the requisite for elaborate manual feature engineering and ensuring a nuanced encapsulation of underlying hierarchical data patterns. This capacity to autonomously navigate through complex, non-linear relationships in data not only strengthens the model's predictive capability but also often ensures a more generalized and robust performance on unseen data.

While paper by Perez-Rosas et al. [33] utilizes neural network models for audio modality in deception detection, achieving accuracy metrics in the range of 54.55–59.50% for various annotators, the CNN model herein provides a comprehensive methodology with a mean accuracy of 93.5% as shown in Table 3. By leveraging convolution layers, the model autonomously identifies and extracts potent features

Table 2 Performance of traditional classification models

Model	Accuracy (%)	ROC-AUC (%)	Precision (%)	Recall (%)	<i>F1</i> -Score (%)
GradientB	56	61.4	54.4	50.5	50.4
XGBoost	55.8	61.6	53.9	50	49.9
SGDClassifier	52	54.2	50.7	43.4	43
SVM RBF	64	67.4	62.5	69.5	65.4
Random Forest	61.2	65.5	61.4	49.5	52.6
Naïve Bayes	54.8	60.9	63.4	48	47.7
AdaBoost	57.2	59	56.8	43.2	45.8
DT	58.5	59.4	57.6	55.2	54.9

Table 3 Comparison of audio modality by Perez-Rosas et al. [33]

Model/Annotator	Audio (%)
A1	51.24
A2	55.37
A3	59.50
Proposed CNN	93.50

from the audio data, potentially illuminating subtle vocal variations and phonetic nuances crucial in discerning deceptive instances. The integration of this CNN model could enhance the efficaciousness of multimodal systems, both by refining the predictive accuracy of the audio modality component and potentially bolstering the overall integrated output of a multimodal system, ensuring a more robust and dependable deception detection mechanism, especially in real-world, trial data scenarios.

These differences in employment patterns are likely influenced by a variety of factors, including social stigma and discrimination against individuals with disabilities, as well as access to education and job training opportunities. Addressing these disparities will require a multifaceted approach, including policy changes to improve access to education and training programs, as well as efforts to combat negative attitudes and stereotypes toward individuals with disabilities in the workplace.

The model proposed by Gupta et al. [32], where Random Forest and KNN were employed for the audio modality and achieved accuracies of 53.24 and 53.22%, respectively, the CNN model unveils its proficiency in handling sequential audio data as shown in Table 4. The model’s adeptness at managing non-linear, hierarchical data structures and discerning complex relationships within audio data provides it with a distinctive edge, especially when compared to Random Forest and KNN which might necessitate manual feature engineering and potentially lack in representing intricate, hierarchical data patterns. Thus, the integration of a CNN model, with its inherent feature learning and pattern recognition capabilities, could significantly amplify the reliability and effectiveness of deception detection systems, particularly in a multimodal context.

Table 4 Comparison of audio modality by Gupta et al. [32]

Model	Set A	Set B
Random Forest	53.24%	54.89%
KNN	53.22%	56.22%
proposed CNN	93.5%	

This comparative lens underscores the instrumental role of methodological selection in optimizing deception detection, with the CNN model showcasing potent capabilities in managing the nuanced, complex structures of audio data.

5 Conclusion

In conclusion, this study delved into the realm of deception detection, a crucial task in various applications, such as security and fraud detection, with a particular focus on the audio modality. Through extensive experimentation and comparative analysis, we have demonstrated that the application of Convolutional Neural Networks (CNNs) for audio classification yields notable advantages over traditional classification algorithms.

Our CNN model exhibited a remarkable Mean AUC of 0.9350, showcasing its ability to effectively capture intricate patterns in audio data, which are often elusive to traditional classifiers. This heightened predictive performance can be attributed to CNNs’ innate capacity to handle sequential data and autonomously learn essential features through their convolutional layers, eliminating the need for manual feature engineering. Moreover, CNNs excel in modeling intricate non-linear relationships and interactions among audio features, a key advantage in the multifaceted domain of deception detection.

It is important to acknowledge that one of the limitations of this study, and a broader challenge in the field, is the scarcity of authentic and diverse datasets for deception detection. This limitation impedes the comprehensive evaluation of deception detection models. Addressing this issue through the creation and sharing of more realistic and representative datasets is imperative for advancing the field and enhancing the reliability of deception detection systems.

In light of our findings, we propose that CNNs have the potential to play a pivotal role in the development of robust and effective multimodal deception detection systems. By integrating CNN-based models, such as the one presented in this study, into multimodal architectures, we can harness the power of deep learning to further enhance the predictive capabilities of audio modality components. This holistic approach to deception detection, incorporating diverse modalities, offers promising avenues for future research and real-world applications. Our model’s exceptional performance in audio classification positions it as a valuable asset in the construction of such multimodal systems.

In summary, our research underscores the promise of CNNs in audio-based deception detection and underscores the need for comprehensive, authentic datasets to propel the field forward. As we continue to refine and expand the capabilities of machine learning and deep learning techniques, the potential for deception detection in practical scenarios becomes increasingly attainable, ultimately contributing to enhanced security and trust in various domains. Our model, with its remarkable accuracy, is poised to support multimodal models in the proper classification of deception, offering a robust foundation for future endeavors in this domain.

References

1. Camara MK, Postal A, Maul TH, Paetzold GH (2024) Can lies be faked? Comparing low-stakes and high-stakes deception video datasets from a machine learning perspective. *Expert Syst Appl* 249:123684. <https://doi.org/10.48550/arXiv.2211.13035>.
2. Chen Y, Zhu Y, Yan Z, Ren Z, Huang Y, Shen J, Chen L (2023) Effective audio classification network based on paired inverse pyramid structure and dense MLP block. https://doi.org/10.1007/978-981-99-4742-3_6
3. Stergiou A, Damen D (2023) Play it back: iterative attention for audio recognition. In: *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp 1–5
4. Liu X, Liu H, Kong Q, Mei X, Plumbley MD, Wang W (2023) Simple pooling front-ends for efficient audio classification. In: *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp 1–5
5. Gazneli A, Zimmerman G, Ridnik T, Sharir G, Noy A (2022) End-to-end audio strikes back: Boosting augmentations towards an efficient audio classification network. *arXiv preprint arXiv:2204.11479*
6. Shaar S, Martino GDS, Babulkov N, Nakov P (2020) That is a known lie: detecting previously fact-checked claims. *arXiv preprint arXiv:2005.06058*
7. Kiran KO, Mahatmanand AS, Manu V, Nithin Krishnan RS, Vijayan M (2022) A review on lie detection using artificial intelligence and machine learning. *IJCRT* 10(4). <https://ijcrt.org/papers/IJCRT2204149.pdf>
8. Rodriguez-Diaz N, Aspandi D, Sukno FM, Binefa X (2021) Machine learning-based lie detector applied to a novel annotated game dataset. *Fut Internet* 14(1):2
9. Salunkhe A (2021) Attention-based bidirectional LSTM for deceptive opinion spam classification. *arXiv preprint arXiv:2112.14789*
10. Li X, Li X (2022) ATST: audio representation learning with teacher-student transformer. *arXiv preprint arXiv:2204.12076*
11. Zhao Y, Hessel J, Yu Y, Lu X, Zellers R, Choi Y (2021) Connecting the dots between audio and text without parallel data through visual knowledge transfer. *arXiv preprint arXiv:2112.08995*
12. Zhang S, Jafari O, Nagarkar P (2021) A survey on machine learning techniques for auto labeling of video, audio, and text data. *arXiv preprint arXiv:2109.03784*
13. Phan H, Hertel L, Maass M, Koch P, Mazur R, Mertins A (2017) Improved audio scene classification based on label-tree embeddings and convolutional neural networks. *IEEE/ACM Trans Audio Speech Lang Process* 25(6):1278–1290
14. Mo S, Tian Y (2022) Multi-modal grouping network for weakly-supervised audio-visual video parsing. *Adv Neural Inf Process Syst* 35:34722–34733
15. Afouras T, Asano YM, Fagan F, Vedaldi A, Metze F (2022) Self-supervised object detection from audio-visual correspondence. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 10575–10586

16. Zaheer MZ, Mahmood A, Khan MH, Segu M, Yu F, Lee SI (2022) Generative cooperative learning for unsupervised video anomaly detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14744–14754
17. Sreeja Kumari S, Arya JN, Sandhya S (2022) A review of lie detection techniques. *Int Res J Eng Technol (IRJET)* 9(8):1101. <https://www.irjet.net/archives/V9/i8/IRJET-V9I8180.pdf>
18. Liu D, Qu X, Wang Y, Di X, Zou K, Cheng Y et al. (2022) Unsupervised temporal video grounding with deep semantic clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 36(2), pp 1683–1691
19. Krishnamurthy G, Majumder N, Poria S, Cambria E (2018) A deep learning approach for multimodal deception detection. In: International Conference on Computational Linguistics and Intelligent Text Processing, pp 87–96. Springer Nature Switzerland, Cham
20. Mittal T, Bhattacharya U, Chandra R, Bera A, Manocha D (2020) Emotions don't lie: an audio-visual deepfake detection method using affective cues. In: Proceedings of the 28th ACM international conference on multimedia, pp 2823–2832
21. Bareeda EF, Mohan BS, Muneer KA (2021) Lie detection using speech processing techniques. *J Phys Conf Ser* 1921(1):012028
22. Srivastava N, Dubey S (2018) Deception detection using artificial neural network and support vector machine. In: 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, pp 1205–1208
23. Marcolla FM, de Santiago R, Dazzi RL (2020) Novel lie speech classification by using voice stress. In: ICAART (2), pp 742–749
24. AlArfaj AA, Mahmoud HAH (2022) A deep learning model for EEG-based lie detection test using spatial and temporal aspects. *Comput Mater Continua* 73(3)
25. Ramadhan NG, Ramadhan TI (2022) Analysis sentiment based on IMDB aspects from movie reviews using SVM. *Sinkron* 7(1):39–45
26. Bhamare AR, Katharguppe S, Nancy JS (2020) Deep neural networks for lie detection with attention on bio-signals. In: 2020 7th International Conference on Soft Computing and Machine Intelligence (ISCM). IEEE, pp 143–147
27. Didi Y, Walha A, Wali A (2022) COVID-19 tweets classification based on a hybrid word embedding method. *Big Data Cogn Comput* 6(2):58
28. Javaid H, Dilawari A, Khan UG, Wajid B (2022) EEG guided multimodal lie detection with audio-visual cues. In: 2022 2nd International Conference on Artificial Intelligence (ICAI). IEEE, pp 71–78
29. Altalbe A (2021) Audio fingerprint analysis for speech processing using deep learning method. *Int J Speech Technol*
30. Venkatesh S, Ramachandra R, Bours P (2020) Video based deception detection using deep recurrent convolutional neural network. In: Computer Vision and Image Processing: 4th International Conference, CVIP 2019, Jaipur, India, September 27–29, 2019, Revised Selected Papers, Part II 4. Springer Singapore, pp 163–169
31. Ali R, Farooq U, Arshad U, Shahzad W, Beg MO (2022) Hate speech detection on Twitter using transfer learning. *Comput Speech Lang* 74:101365
32. Gupta V, Agarwal M, Arora M, Chakraborty T, Singh R, Vatsa M (2019) Bag-of-lies: a multimodal dataset for deception detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops
33. Pérez-Rosas V, Abouelenien M, Mihalcea R, Burzo M (2015) Deception detection using real-life trial data. In: Proceedings of the 2015 ACM on international conference on multimodal interaction, pp 59–66

Comparative Study of NLP Tools for the Detection of Cyber Bullying



Jayati Bhadra, Navaneeth Kashyap, Kunkum Poovamma, and S. Dutta

Abstract Social networking sites have increased in popularity in recent years because they provide users with a platform where they can connect worldwide and discuss their interests. However, people will benefit from a good environment thanks to the social network. Random cyberbullying is wild, and it poses a serious threat to the physical and mental health of the victim. It is important to find the right protection to detect and prevent this. The proposed method aims to detect the presence of cyberbullying activities on social networks such as Twitter where the tweets are scrapped and made it into a clean dataset and preprocessed the data by removing the “stopwords,” “lemmatization,” “tokenization” and more which is explained further. The dataset is then compared using the “NLP models” (e.g., ‘stanza’, ‘Roberta’, ‘NLTK’). In this article, we also compare the performance of three different NLP tools in detecting cyberbullying on Twitter profiles. We evaluated the tool using data from recorded tweets. We found that NLTK outperforms “Stanza” and “RoBERTa” in detecting cyberbullying. “NLTK” achieved an accuracy of 94%, while “Stanza” and “RoBERTa” together achieved an accuracy of 19%.

Keywords NLTK · RoBERTa · Stanza · Natural language processing

J. Bhadra · N. Kashyap · S. Dutta (✉)
Department of Advanced Computing, St. Joseph’s University, Bangalore, India
e-mail: duttasanjay098@gmail.com

J. Bhadra
e-mail: jayatibhadra@sju.edu.in

K. Poovamma
Aditya Birla Management Corporation Pvt Ltd, Bangalore, India

1 Introduction

Cyberbullying is defined as “The use of a computer, mobile device, or other technology to send or transmit messages or images that hurt or embarrass others. They abuse online security to our benefit” by The National Crime Prevention Commission. Web 2.0 which allows easy, interactive and 24/7 access to online platforms, and opens the door to cyber crimes such as cyber bullying. The consequences can also be long-lasting and affect the victim mentally, physically and emotionally. Discrimination, cheating, slander, anti-bullying, cyber harassment, racism, and gender bias are just some of the other symptoms of cyberbullying. They often appear on mobile devices, gaming platforms, and social services. It is an action that threatens, angers or harms the victim. As social media use increases, cyberbullying also spreads rapidly. Although research on the effects of bullying continues, little progress has been made in identifying cyberbullying. Therefore, it is important to control and prevent all these problems by handling them carefully.

Various “machine learning” classification algorithms can be used to identify cyberbullying. In cyberbullying, “natural language processing (NLP)” comes into play to identify the nature of bullying. This tool analyzes the content of the message. Find specific terms and articles regarding cyberbullying.

In this study, we compare the effectiveness of our NLP tools in detecting the nature of bullying in Twitter profiles. We use a tweet log to measure its effectiveness.

“Natural Language Toolkit” (‘NLTK’) is a well-known “open source Python” package for “natural language processing” (‘NLP’). It is a popular framework for creating Python programs that work with human data. With powerful NLP tools which provides easy access to more than 50 relational and lexical databases (such as “WordNet”), in addition to libraries for “classification,” “tokenization,” “stemming,” “tagging,” “parsing,” and “semantic reasoning.” The process of identifying and extracting ideas from the text is called idea analysis. This is useful for determining whether a text is good, bad or average.

Another python library for “natural language processing” called “Stanza” includes tools that convert human speech sequences into sentences and word lists. It also, create phrases, pronunciations, and objects from these words and provides analysis-based explanations, and annotate websites. The tool uses traditional dependency strategies which is designed to work seamlessly in more than 70 languages. Stanza features real neural networks that allow you to quickly train and evaluate using your own recordings. The model is based on the “PyTorch” library. The software runs faster on a GPU-enabled computer. “Stanza” also includes a Python interface to the “Core NLP” Java package that provides additional functionality such as polling, importance analysis, language comparison, and derivatives.

In addition to above two packages, “RoBERTa” is a self-supervised transformer model based on various English datasets before training. This means that it simply trains the raw text without human annotations (so it can use publicly available data) and then uses automated services to create information and products from those articles.

2 Literature Survey

Nandhini et al. [4] used the Levenshtein algorithm to identify cyberbullying and used “Naive Bayes classifier” to classify cyberbullying activities. This classifier assumes that all predictors (or features) are independent and rarely occurs in real life.

Bozyiğit et al. [5] uses the equivalent data of 5000 saved multiple social media contents. Analyze the relationship between cyberbullying and social media characteristics using “chi-square tests.”

López-Vizcaíno et al. [2] called for early detection of “cyberbullying” in social networks. Two groups of features and two early detection methods are used here. The results show an improvement of up to 42% for basic detection methods (“Random Forest,” “Ada boost,” “Support Vector Classifier”).

Kumari et al. [6] proposed a pre-learned “VGG-16 network” and a “convolutional neural network” to extract features from images and texts. They are optimized by genetic algorithms to get efficiency. The proposed model was validated using data containing text and images and achieved an *F1* score of 78%.

Giumetti et al. [7] reported various predictors of cyberbullying via social media, revealed some limitations of existing research, and made some recommendations.

Vimala Balakrishnan et al. [1] reported that the cyberbullying detection system used in this study used psychological characteristics of Twitter users, such as behavior, thoughts, and emotions. “Machine learning” was used to classify tweets into one of four groups: “bullies,” “attackers,” “spammers,” and “humans.” This algorithm uses various Twitter-based features, like text, user, and network information. The results show that the use of personality and behavior in the tweet can increase the detection of cyberbullying.

Rahul Ramesh Dalvi et al. [9] this article, bullying tweets, posts, etc., suggested to show how to use the software to identify cyberbullying on Twitter. “SVM” and “Naive Bayes” are two models used to identify and test cyber bullying content on social media. The accuracy rate of “Naive Bayes” and “SVM (Support Vector Machine)” is 71.25% and 52.70%, respectively, and can detect real bullying content. However, “SVM” outperformed “Naive Bayes” on similar tasks on the same dataset. In addition, tweets are received via the Twitter API and given to a model to determine whether they constitute harassment.

López-Vizcaíno et al. [2] studied various methods to determine how long it takes to identify cyberbullying on social media. It uses the training tracking system with two early detection models, initial and dual early detection models. While the latter requires two “machine learning” methods, the former follows a simpler approach. Two specific groups and two early research methods were adapted to this problem. They perform a qualitative evaluation using real data and show that they see the model’s detection (Extra Tree) accuracy increased by up to 42%, compared to time-sensitive evaluation that leads to delayed decision making.

Varun Jain et al. [3] reported that in the following study, a model based on the detection of cyberbullying behavior in the data was developed using “natural language” and ‘machine learning’ with two different types of data. Cyberbullying like hate speech

on Twitter, and personal attack comments on Wikipedia forums. In this paper, three extraction methods and four classification methods were examined to determine the most effective method. The model provides more than 90% accuracy for information in tweets and more than 80% accuracy for information in Wikipedia.

Yuvaraj et al. [10] create an automatic classification system that can identify cyberbullying letters without having to place them in a high place. On the other hand, the overfitting problem of the classifier prevents it from giving limited answers. Overcoming these limitations, they developed a text classification system that prioritizes tweets, filters out background noise and other irrelevant information, removes selected features, and then classifies them without too much information. This study developed a deep decision tree that uses the hidden layer of a “deep neural network (DNN)” as tree nodes to serve as input.

3 Methodology

Twitter data collected from tagged tweets and while preparing the material, we remove numbers, extra spaces, emoticons, abbreviations, symbols, symbols, and lower case information. And processed the data through 3 different NLP models to detect cyber bullying.

3.1 Approach

Tokenization

Tokenization is a method of breaking text or sentences into smaller parts called tokens. The token here can be in the form of a word, a character, or even a sub-word. Therefore, tokenization can be divided into three categories: character, single word, and sub-word tokenization. Tokenization is a must when analyzing data. There are many layers to examine and syntax to consider. We need to simplify our language so that our machines can easily understand it. This is where we need to use “NLP.” If the data is not tokenized, we cannot use it. This is an important step in both “natural language processing” and advanced “deep learning.” “Tokens are the building blocks of natural language.”

Lemmatization

“Stemming” and “lemmatization” are literal normalization techniques. They can also be called modeling techniques that processes text, words, and additional information. Rooting and lemmatization have been studied and algorithms have been developed for English grammar, conjugation, nouns, genders, adjectives, verbs, adverbs, moods, persons, numbers, gender, etc. It is the changing of certain words to express different sentences, such as, inflection which refers to one or more grammatical groups through

prefixes or suffixes or other internal changes (such as vowel changes). Stemming and lemmatization are widely used in tagging systems, information retrieval, and Web search results. “Stemming” is a way of converting words to their letters. Basics like marking a group of words on the same root, even if the root itself is not a valid word in the language. To compare the effectiveness of three “NLP” tools in detecting cyberbullying on Twitter, we followed the following steps:

- We collected registered tweets containing 31,963 entries.
- Each of our NLP tools is used to extract features from tweets.
- Then we train machine learning models to classify tweets as cyberbullying or non-cyberbullying.
- Finally, we evaluate the model’s performance by comparing the predictions with previously collected data.

3.2 Experimented Results

A. Dataset

The data set for this paper consists of data collected from social media “Twitter.” The data set consists of tweets from people, and the data are classified as positive or neutral sentences. It consists of 31,962 rows and 3 columns.

B. Data Preprocessing Technology

Tokenization: This is the process of breaking objects into smaller parts, such as a word or sentence.

Do not remove words: Remove words that are frequently used in the text, such as “the,” “if,” and “of.”

Stemming and lemmatization: Reduce words to their simplest forms through radicalization and lemmatization.

Part of Speech (POS) tag: Identifying the “part of speech” of each word in a sentence.

Named Entity Recognition (NER): Identification and classification of names in text, such as people, places, and organizations.

Grammar parsing: Creating the parse tree of a sentence and showing the communication between words.

Semantic reason: Extracting meaning from the text.

C. Tools Used

“NLTK”

“Natural Language Toolkit” is a powerful “NLP library,” a function table, and various text processing tools for “classification,” “tokenization,” “stemming,” “tagging,” “parsing,” and “semantic reasoning” which has complete API documentation and easy-to-access online books.

The following are some important features of “NLTK”:

1. Various tools and resources for traditional NLP tasks
2. Open source and modifiable
3. Easy to install and use
4. It has a very wide body and language equipment.
5. Rich training and information available

“NLTK” can be used for many different purposes such as:

Classification of documents—Name recognition—“Machine translation”—Q&A—Chat bots.

“Stanza”

“Stanza” supports NLP tasks, including name recognition, dependency analysis, “part of speech tagging,” “lemmatization,” and “sentence segmentation.” “Stanza” is known for its completely language-independent neural pipeline that provides high performance when processing text in more than 70 languages.

Here is a brief overview of ‘Stanza’s’ capabilities:

- Language independent
- Read ‘NLP’ Projects
- Easy to use
- Accuracy and efficiency
- Community support

“RoBERTa”

The best “BERT” optimization method is a large language model trained on large text and numerical data. It is a transformer-based model, meaning it learns remote dependencies in the text using a self-listening process.

“RoBERTa” has been studied using a variety of methods designed to improve its robustness and performance compared to “BERT.” These methods include:

1. Larger training data from “BERT”
2. Longer training time
3. More Byte-level BPE use tokens
4. Eliminate the goal of predicting the next sentence
5. Leverage dynamic masks

3.3 Experimented Results

All non-harassment related messages are identified from non-harassment related reports.

NLP tools	Accuracy (%)
1. “NLTK”	93.69

(continued)

(continued)

NLP tools	Accuracy (%)
2. “Stanza” for sentiment analysis and “RoBERTa” for sentiment score	19.37

After comparing the performance of three different NLP tools in detecting cyberbullying on twitter data. We find that “NLTK” outperforms “Stanza” and “RoBERTa” in detecting cyberbullying. “NLTK” achieved an accuracy of 94%, while ‘Stanza’ and “RoBERTa” together achieved accuracy of 19%.

4 Conclusion

In this paper, data were analyzed from Twitter and were found that “NLTK” outperforms “Stanza” and “RoBERTa” in detecting cyberbullying. NLTK achieved 94% accuracy, while “Stanza” and “RoBERTa” achieved 19% accuracy.

Further research is required to answer the question why “Stanza’ and ‘RoBERTa” did not performed well. One of the possibilities may be that “Stanza” and “RoBERTa” are not good at detecting cyberbullying. However, our results show that “NLTK” is a useful tool for detecting cyberbullying on Twitter.

References

1. Balakrishnan V, Khan S, Fernandez T, Arabia H (2019) Cyberbullying detection on twitter using big five and dark triad features. *Personal Indiv Diff* 141:252–257
2. Fernández López-Vizcaíno M, Nóvoa FJ, Cacheda F (2021) Early detection of cyberbullying on social media networks. *Futur Gener Comput Syst* 118:219–229. <https://doi.org/10.1016/j.future.2021.01.006>
3. Jain V, Kumar V, Pal V, Vishwakarma DK (2021) Detection of cyberbullying on social media using machine learning. In: Panigrahi PK, Bhateja V, Joshi A (eds) *Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, *Advances in Intelligent Systems and Computing*. Springer, Singapore, vol 1266, pp 1091–1096
4. Nandhini B, Immanuelraj Kumar S (2015) Cyberbullying detection and classification using information retrieval algorithm. In: Goyal D, Mehrotra R, Joshi G, Kumar A (eds) *Proceedings of the International Conference on Advances in Computing, Communications and Data Engineering*. ACM, pp 1–5
5. Bozyiğit A, Utku S, Nasibov E (2021) Cyberbullying detection: utilizing social media features. *Expert Syst Appl* 179:115001. <https://doi.org/10.1016/j.eswa.2021.115001>
6. Kumari K, Singh J (2021) Identification of cyberbullying on multi-modal social media posts using genetic algorithms. *Trans Emerg Telecommun Technol* 32:e3907. <https://doi.org/10.1002/ett.3907>
7. Giumetti GW, Kowalski RM (2022) Cyberbullying via social media and well-being. *Curr Opin Psychol* 45:101314. <https://doi.org/10.1016/j.copsyc.2022.101314>
8. Natarajan Y, Chang V, Gobinathan B, Pinagapani A, Srihari K, Dhiman G, Raja RA (2020) Automatic detection of cyberbullying using multi-feature based artificial intelligence with deep decision tree classification. *Comput Electr Eng*. <https://doi.org/10.1016/j.compeleceng.2021.107186>

9. Dalvi RR, Chavan S, Halbe A (2020) Detecting a twitter cyberbullying using machine learning. In: Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS). Springer, pp 297–301. <https://doi.org/10.1109/ICICCS48265.2020.9120893>
10. Yuvraj et al. (2021) Automatic detection of cyberbullying using multi-feature based artificial intelligence with deep decision tree classification. Comput Elect Eng 92, 107186. <https://doi.org/10.1016/j.compeleceng.2021.107186>

Simulated Intelligence-Based Risk Expectation Models for Diabetic Retinopathy



Balluru Adishankar Nithya, Md. Sameeruddin Khan, and R. Chinnaiyan

Abstract Diabetic retinopathy (DR) is the most well-known and weakening confusion of diabetes, influencing innumerable individuals around the world. The tricky idea of DR permits it to advance without clear side effects until it causes abrupt vision misfortune. This features the significance of grasping the hidden infection instruments and advancing successful clinical mediations. AI (ML) has shown guarantee in foreseeing the probability of DR movement, a region that has been investigated in many examinations. In any case, a huge piece of these examinations are restricted to cross-sectional examinations, catching a general image of the illness at a solitary moment. Furthermore, the range of calculations utilized in these examinations convolutes the undertaking of looking at their outcomes. To address these significant holes, we directed an extensive deliberate survey of important examinations distributed between January 2017 and April 2023. Our pursuit traversed numerous electronic data sets, including IEEE Xplore, PubMed, Springer Connection, Google Researcher, and Science Direct, and eventually brought about thirteen examinations that met our foreordained rules. Our assessment shows that ML-based expectation models for DR movement show promising outcomes. Be that as it may, it is basic to more readily figure out the movement of this illness over the long haul through additional longitudinal examinations. Simultaneously, the survey features various significant examination holes, to be specific the desperation for normalized datasets, the requirement for more vigorous appraisal measures, and support for expanded straightforwardness in creating ML models. Moreover, six key regions were the subject of specialized conversations, spinning around the methodologies and approaches applied in these examinations. These incorporate the requirement for reasonable ML models, which can empower medical services experts to figure out

B. A. Nithya (✉)

Presidency University, Bengaluru, India

e-mail: nithya.20223cse0020@presidencyuniversity.in

Md. Sameeruddin Khan

School of CSE and IS, Presidency University, Bengaluru, India

e-mail: deancse@presidencyuniversity.in

R. Chinnaiyan

School of CSE, Alliance University, Bengaluru, India

the fundamental highlights hidden expectations, as well as the significance of careful information pre-handling and sensible determination of elements. In rundown, our audit gives significant experiences into the ongoing scene of ML-based risk expectation models for DR progress and the related difficulties. It gives off an impression of being a significant asset for specialists endeavoring to make more exact, solid, and powerful models to anticipate the movement of DR and consequently give data for clinical independent direction.

Keywords Diabetic retinopathy · Diabetes · Vision

1 Introduction

Vision is the act or power of seeing of a singular's life and absence of vision, frequently alluded to as visual debilitation and its general construction, absence of vision, can restrict the capacity to perform errands day-to-day undertakings, influences individual fulfillment, diminishes free versatility, and can prompt aversion [1, 2]. Around the world, there are around 217 million individuals with moderate or extreme vision weakness, 188.5 million individuals with gentle vision impedance and 36 million individuals with serious vision debilitation (World Government assistance Affiliation orientation). Much really disturbing, without new and compulsory mediations, these numbers will quadruple before long, with an expected 38.5 million people experiencing visual hindrance by 2020 what's more, 115 million individuals by 2050 because of proceeded with advancement and maturing population [3]. Diabetic retinopathy (DR) is the main source of new preventable vision impedance in non-industrialized count-attempts [4–7]. DR is a moderate microvascular degenerative difficulty brought about by long haul diabetes (DM). The movement of DR can be incredibly quick, and it is estimated that 12–27% of moderate non-proliferative diabetic retinopathy (NPDR) may advance to diabetic retinopathy multiplication (PDR) in no less than one year or less. 52% of outrageous NPDR types progress to PDR inside about a year [8]. In the meantime, a few patients with gentle to extreme NPDR hold vision for a while and progress gradually in the long haul, while different patients progress quickly to PDR [9]. Regardless, it is indistinct which patients fall into the quick or sluggish development class of DR. Forestalling DR relocation is maybe perhaps of the most troublesome undertaking ophthalmologists face each day when treating patients [10]. Along these lines, they frequently expect to distinguish DR right on time before it advances to irreversible vision misfortune, since preliminary clinical investigations have demonstrated the way that early treatment can slow down movement [1, 7, 11–14]. Understanding DR development is a significant issue in clinical examination and practice [15]. Risk factors assume a basic part in the pathogenesis of DR and the movement to the visual disability period of PDR, and these game factors might be liable for the uncommon fluctuation that characterizes the course and musicality of DR in various patients [9, 16, 17].

Screening to recognize people with elevated degrees of betting-related DR movement from the antiproliferative stage to the proliferative stage, and early mediation is essential to alleviate the expanded gamble of visual impedance and reduce costs related with checking further developed phases of DR [1, 10, 18]. A total comprehension of the development of Preprint submitted to Diabetic Retinopathy Exploration Overview October 13, 2023 DR because of these impacting factors is the reason for thinking about counteraction and proper medicine [19]. Along these lines, it is critical to precisely anticipate the advancement of DR and utilize significant gamble factors as information components to fabricate expectation models [20–23]. Betting anticipation models incorporate man-made consciousness computations or mathematical conditions that can be utilized to appraise the probability of specific antagonistic occasions happening from now on. The utilization of counterfeit intelligence, particularly profound convolutional cerebrum association, to make risk expectation models is unavoidable, as it can identify complex designs in a lot of patient information what's more, figure out the intricate connection between the human components of the game-limited instances of the past. As of late, models have been proposed utilizing man-made brainpower methodologies, taking into account, three significant sorts of pointers or hazard factors, including fundus pictures, social division, and hereditary personistics. A tests join different game components to impact model show. Along these lines, investigators have put forth different attempts to advance powerful assumption models for early debacle recuperation purposes. Past models depended on measurable pursuit procedures and hand-tuned remarkable quality plans, however, they were not strong, and their prejudicial power was extremely poor contrasted with utilizes a simulated intelligence process, so they show generally disappointing forecast execution. The motivation behind this audit is to extensively gather, assess, and present the consequences of different existing investigations on ML-based DR movement expectation models. It expects to team up with different specialists in portable pursuit replies to fabricate ML-based risk avoidance models for the DR development and its embodiments; Join the consequences of these examinations and take apart the difficulties, limits, and yet again search holes to illuminate choices about computer-based intelligence systems and measurements to construct new prescient models.

2 Related Work

Progressing assessments have basically examined both assessment strategies and man-made insight examination to expect the development of diabetes-related issues with by and large unique center interests. For example, Tan et al. [24] executed a SLR focusing in on looking at ML techniques to predict improvement toward diabetic issues, while excepting DR. In this article, 32 tests were picked and evaluated. For each test, the producers portrayed the ML pipelines and executions, and found that non-standard woods had the best expected show. Of the 32 tests surveyed, 31

slumped models were at high bet of somberness, for which external underwriting was controlled to allow them to be completed in clinical practice.

Moreover, Xu et al. [25] coordinated an assessment moreover focusing in on ML hunch models to evaluate their capacity to provoke diabetic issues like diabetic retinopathy, nephropathy, and neuropathy in diabetic patients. The PubMed information base and Google Investigator crawler were searched for fundamental reviews between January 1, 2016 and May 31, 2019, and only six outlines met the actions qualifying standards. For each investigated study, the kind of ML process, key estimations, datasets used to drive the models, as well as the estimations presented, endorsement strategy, and external supports were irrefutably seen. The creators prescribe that future examinations should use clinical information assembled in uncontrolled-remotely supported conditions.

Considering this, Van der Heijden et al. [26] fairly supported risk expectation models to additionally foster DR in patients with type 2 diabetes. Two informational collections were broke down from inception to December 2019. 12 surveys a review involving 16 models predicting an extensive variety of DR up to visual incapacity was picked. The survey focused in on the distant endorsement of pal related models in a clinical setting in the Netherlands. The effect of help shows that most models mistakenly anticipate the patient's waged that produces DR.

Finally, Haider et al. [27] coordinated an outline of 22 articles from 2014 to Walk 2017 that included 14 prognostic models for the DR improvement. External endorsement, inward endorsement, biased power, slanting gamble, people, test size, results, quantifiable model, and supported model execution/appraisal in each article and discussion. It is suggested that the evaluated models require reviving and external underwriting while at the same time spreading out a clinical office for application in clinical practice. Regardless, this assessment revolves around confirmation-based prognostic models made for the DR improvement.

Normally, the continuous sagacious composing essentially revolves around risk estimate assessments that produce DR and other diabetes-related issues, while attempting to disregard focuses on that really convey mimicked insight based assumption models to advance DR beginning with one stage then onto the following level of DR earnestness steps using vertical markers. Besides, each survey doesn't give a total assessment or association between the philosophies used in the picked assessments. Additionally, they didn't explain assessments that solidified imaging modalities with other clinical signs to drive insightful models of DR improvement. In any case, each article highlights the necessity for external endorsement of emergency focus assumption models delivered for application to clinical practice. As required, this study intends to fill the openings perceived in current assessments and give encounters into state-of-the-art unlawful duty aversion risk presumption models for the DR advancement (Fig. 1).

Diabetic Retinopathy

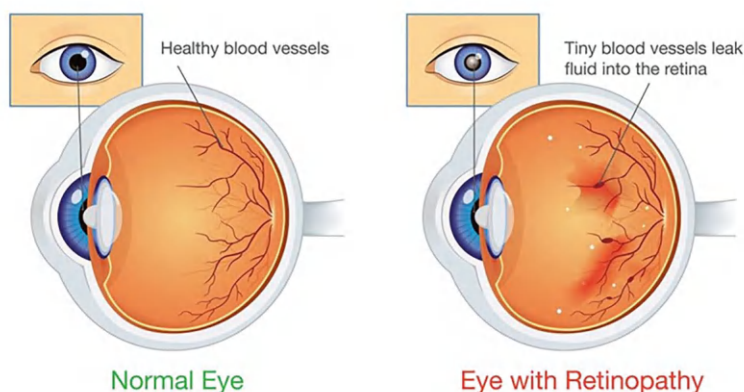


Fig. 1 Diabetic retinopathy

3 Material and Strategies

This article follows the Scholarly Study (SLR) approach and follows the Specifying Promoters for Successful Overviews and Engineered Appraisals show ([28]; PRISMA 2020) which expresses the principles guarantee that SLR is executed really. The PRISMA show was applied to this study since it is the norm for leading reviews in science, which is the principal hindrance. PRISMA shows were additionally followed to design the testing questions, research strategies, consider and keep away from frameworks, survey chosen examinations, and consolidate the data introduced in this article.

3.1 Research Questions

The examining questions (Q1–Q3) are taken from the targets of this SLR and are as per the following:

Question 1: What applications are available as indicated by the singular assumptions for the DR development? Question 2: What are the pipelines, pointers, datasets, qualities, and related imperatives of current ML-based assumption models for DR developments? Question 3: What are the hardships in investigation and examination that are recognized here?

3.2 Scan Methodology for Related Examinations

This search system comprehensively describes the methodology for collecting search terms, search techniques for relevant surveys, survey documents, and selection of the data sets examined. The search terms were chosen based on scientists' information about the field of knowledge. The main search term was "Diabetic Retinopathy". Other survey terms used were classified into two categories: motion recognition strategies and descriptions. The tagline "Diabetic Retinopathy" was combined with a term from the Forecast section and another term from the Motion Description Methods segment to shape the dataset's search strings. For example: "Diabetic retinopathy + Movement + Waiting + AI." That's a total of 330 quest chains per information base for reference.

Incorporation and Avoidance Measures

Review standards include any reviews created, evaluating the visibility and effectiveness of a model from a practical perspective, and approving an AI model that includes risk factors to create the model expected shape for the DR movement. Articles that focus on testing, controlling, exploring, or concluding DR and forecasting DR movement using non-AI-based strategies are prohibited. Book articles, collection procedures, and book entries were also rejected. There are no restrictions on delivery date, age, and type of diabetes. The references of the selected surveys were also reviewed to find relevant articles and the authors' distribution lists were also reviewed for additional assessments applicable to the review check this out.

3.3 Concentrate on Choice and Information Extraction Methods

Three cycles are utilized to pick which readings to consider: diary title, plan, and assessing the article by perusing. Utilization of titles of studies, tests, and unfit duplicates in each unique data base is dismissed. Furthermore, distributed fixings and references were inspected to distinguish concentrates that met the overview models. The last articles were chosen in light of the six examination questions (Q1–Q3) underneath. Included information, where available, were removed from the included reviews and arranged:

(1) General qualities: author(s), year of distribution and (2) Simulated intelligence model properties: model plan, measurements, model execution assessment (prescient capacity), model assessment, endorsement, qualities, and limits, (3) Informational indexes: size, populace, age, direction, type and term of diabetes, missing data, data handling techniques, balance/unusual data, and the board. The Center Evaluation and Data Extraction for Purposely Checking Hope Supporting Examinations (CHARMS) conspire [29] was utilized to make an essential appraisal of the inclining hazard and suitability of the examination. Three (3) independent experts are likewise used to

recognize audits, remove data, and evaluate hazard of inclination, and resolve clashes through exchange and casting a ballot.

3.4 Profound Artificial Intelligence Related Specialized and Basic Conversation of the Took on Procedures and Approaches

In-depth and fundamental conversation related to artificial intelligence about the processes and approaches taken.

This segment discusses the six (6) in-depth simulated intelligence-related conversations and processes performed.

3.4.1 Model Determination and Assessment

The community and relationships of DR were analyzed by the creators in [30] among 757 people aged 40 and over who participated in the 2005–2006 Public Health and Dietary Assessment Overview (NHANES). The LR and KNN models were used by the creators to evaluate the presence of DR among members, using 15 indicator factors. The external approval data set is used to think about the area under the beneficiary's work mark (AUROC) and the mean square error (MSE) of each calculation. This was done using the companion copy from NHANES 2007 to 2008. The results of the calculated recurrence test showed little association with the risk of diabetic retinopathy. Specifically, women (OR = 4.130, 95% CI: 1.820–9.380; $P < 0.05$), had higher HbA1c levels (OR = 1.665, 95% CI: 1.197–2.317; $P < 0.05$), serum creatine (OR = 2.952, 95% CI: 1.274–6.851; $P < 0.05$), and eGFR (OR = 1.009, 95% CI: 1.000–1.014, $P < 0.05$) are serious risk factors for people with diabetes retinal disease. In-house validation results show that the typical presentation across all models is almost identical (AUROC 0.945), with k-nearest neighbors (KNN) showing the highest value at 0.984 AUROC. After external approval, the strengths of the model were still revealed, with only slight reductions in discrimination. The area under the receiver's work mark inflection point (AUROC) remained at or above 0.902, demonstrating excellent predictive ability. Among the models, the K-nearest neighbor (KNN) calculation has the best performance, with an AUROC of 0.982. Both the calculated recurrence model and the AI model are demonstrated strengths in the clinical analysis of diabetic retinopathy.

The goal of this review [31] is to predict the incidence of diabetic retinopathy while performing component extraction to distinguish important factors. The current review includes assorted information from the UCI AI archive. Various AI (ML) techniques were used to analyze the dataset and decide on ideal presentation metrics including response likelihood, selectivity, true positive rate (tp), false negative rate (fn), and collector work mark curvature (ROC). The current evaluation uses various

AI calculations including Innocent Bay, Sequential Negligible Streamlining (SMO), Strategic Recurrence, Stochastic Angle Plunge (SGD), Bag Classification, J48 Classification, Selective Tree Classification, and Arbitrary Forest Classification. Calculated recurrence showed the most positive results regarding the implementation in general.

3.4.2 Highlight Design and Choice

In their review, the creators [32] presented a deep brain network ensemble and imagination element identification technique consisting of four stages. In the baseline stage, the preprocessed entropy images effectively improve the quality type of the retina. In this way, quality is determined through a deep group model that wraps the InceptionV3, ResNet101 and Vgg19 structures, applied to the fundus image. In this way, the previously mentioned qualities are coordinated to create an open compositional space. To smooth the component space, we propose a set of four-step strengths identification strategies, especially Least Repetitive, Highest Relevance, Chi-square Test, ReliefF, and F, to the purpose is to differentiate between the best elements. Additionally, reasonable qualities are selected from most election philosophies with the aim of easing multifaceted computer design. The use of a help vector machine, a common AI classifier, was eventually used to characterize diabetic retinopathy. This possible technique has been tested on freely available datasets such as Kaggle, MESSIDOR-2, and IDRiD. Advanced calculations show an exact speed of 97.78%, a response speed of 97.6%, and a specific speed of 99.3% when using 300 floating points turn on the top. These results are better than other current driving techniques. The paper titled Dayana and Emmanuel [33] presents a strategy using deep learning procedures to improve highlight identification with the ultimate goal of accurately staging the severity of diabetic retinopathy lines in fundus images. First, a preprocessing step is performed to remove all unnecessary signals and improve the separation between reflections of the retinal fundus image. Therefore, the candidate's wound-related concerns are recognized through the use of a combined consideration-based organization (AFU-Net). Finally, a meaningful convolutional brain organization is used to classify the different stages of diabetic retinopathy and thus the weights of the model are modified using an indistinguishable calculation Okay. The proposed strategy shows exceptional performance in two freely available benchmark datasets compared to the current state-of-the-hand methods, evaluated by metrics, e.g., F1 score, precision, responsiveness, and distinctiveness.

3.4.3 Information Preprocessing and Irregularity Handling

In their distribution [34], the creators introduced a strategy for preprocessing information from a variety of Fundus Images (CFPs). Then, the scientists directed the extraction of highlights using the head test (PCA) method. The present review

presents an initial method for multi-brand salient feature extraction and characterization (ML-FEC) using a prefabricated convolutional brain (CNN) design within the framework deep learning. In this way, a motion learning strategy was used to target the preparation of a specific visual arrangement. This is achieved using three state-of-the-art CNN architectures, namely ResNet50, ResNet152 and SqueezeNet1, which involve boundary adjustment to accurately recognize and align lesions. The test results demonstrate that ResNet 50 achieves 93.67% accuracy with a Hamming deficit of 0.0603, Squeezenet1 achieves 91.94% accuracy with a Hamming deficit of 0.0805 and ResNet 152 achieve an accuracy of 94.40% with a Hamming deficit of 0.0560. These findings suggest that ResNet 152 is a feasible model for use in daily clinical practice and to work with large-scale diabetic retinopathy screening programs.

During the basic research phase [35], various AI computations, such as strategy recurrence, Backup Vector Machine, k-nearest neighbors, corner support, Reliable Bays, and Irregular Forests were introduced used to study the accuracy of patient-related hunches. Prerequisites. After investigation, it was resolved that the Arbitrary Woods calculation yielded a most no-table accuracy of 80.7% after implementing a destructive oversampling method to account for the loss balance in the data set. In the next step, democratic accounting is used to employ three models that show widespread implementation. These findings are promising as the model achieved an accuracy of 82.0% using the default dataset and an accuracy of 81.7% when using the appropriate dataset fit.

3.4.4 Move Learning and Generalizability

The review [36] analyzes the use of state-of-the-art deep learning and motion models in relation to clinical imaging investigations, with clear reference to recognizable evidence of diabetic retinopathy. This study provides an in-depth investigation of open-source datasets containing images of diabetic retinopathy. Kaggle dataset is used to prepare and evaluate the proposed model.

The review [20, 21] presents an intelligent method using deep brain gating (DNN) linked with recursive element elimination (RFE) to enable ideal localization of diabetic retinopathy (DR) using individual risk factors. The feasibility model uses recursive element elimination (RFE) to remove irrelevant reflections and deep brain organization (DNN) to characterize infection. This review used a directly available dataset to evaluate early-stage diabetic retinopathy (DR), using both the proposed model and various state-of-the-art models. The feasibility model achieved a forecast accuracy of 82.033%, which is a clear improvement over the current model. Effective extraction of important risk factors for retinal disease can be achieved through the use of recursive component elimination (RFE).

3.4.5 Interpretability and Reasonableness

In Ref. [35], a reasonable calculation was developed using gradient-weighted layer drive programming. The motivation behind this calculation is to provide a visualization of the reflections used by the model to classify retinal images as characteristic of diabetic retinopathy. The proposed technique involves screening a deep convolutional brain network prior through deep learning computation, which is determined to promote its viability in identifying diabetic retinopathy. The model preparation targets a specific subset of the EyePACS dataset, using cosine quenching to address the learning rate while consolidating the warm-up phase. This technique was performed to improve the accuracy of the preparative interaction. Experiments were performed on all nine datasets. The proposed organization yielded an expanded group rate, confirmed by an area under curvature (AUC) value of 0.986, responsiveness of 0.958, and specificity of 0.971 for EyePACS. The area under the curve (AUC) values for MESSIDOR, MESSIDOR-2, DIARETDB0, DIARETDB1, Gaze, IDRID, E-opththa, and UoADR were 0.963, 0.979, 0.986, 0.988, 0.964, 0.957, 0.984, and 0.990 single.

In their distribution, the creators [37] present another army model intended for deep exploration to provide interpretability. This model merges loadings from different individual models to create a concordance method to distinguish important elements of the retinal injury available in fundus images. In this way, the extracted components are fed into a classifier specifically designed for the specific task of deciding the severity of diabetic retinopathy, ultimately leading to a definitive result. The evaluation used a periodic learning rate approach, using a programmed learning rate locator to improve model accuracy. The model was prepared on a dataset of fundus images with varying degrees of diabetic retinopathy (DR), obtained from the APTOS dataset.

3.4.6 Moral Contemplations and Inclination

AI-based risk expectation models for diabetic retinopathy (DR) movement involve different moral contemplations and inclination suggestions. Reasonableness and predisposition concerns emerge from potential inclinations present in the datasets utilized and the gamble of algorithmic inclination in expectations. Guaranteeing value and decency is critical to forestall the intensification of aberrations and give equivalent admittance to determination and therapy. Straightforwardness and logic are essential to lay out trust and responsibility, with endeavors to foster models that can be deciphered and made sense of. Safeguarding patient protection and information security is foremost, expecting adherence to protection guidelines and execution of powerful safety efforts. Human oversight and clinical aptitude ought to be kept up with, with models seen as choice help devices as opposed to substitutions for clinicians. Clinical approval and outside assessment are important to guarantee precision, security, and clinical utility. Informed assent ought to be gotten from patients, and they ought to be instructed about the reason, advantages, dangers, and impediments

of utilizing prescient models. Patient strengthening and contribution in dynamic cycles are critical. Inclination location and moderation strategies, alongside reasonableness assessments, ought to be utilized to limit oppressive predispositions and guarantee impartial results. Continuously checking and evaluating tilt is essential to ensure the reliability of model developments. Overall, ethical reflection and bias reduction promote equitable access, patient safety, simplicity, and the use of reliable computational information in predicting DR's movement.

4 Results and Examination of List Items

Projects of five (5) datasets using the keywords ((diabetic retinopathy prediction AND) AND AI) yielded 2026 records, of which 557 were duplicates. After the works were selected and the titles edited, 30 reports were reviewed for quality assessment and 13 reports met the standards of review and assessment questions and were finally selected for audit. Figure 2 demonstrates the appropriation of concentrates by year of distribution.

4.1 Investigation of List Items

Q1: Open applications for people anticipating DR movement A monetarily open essential program for planned DR movement assumptions relies upon planning accessible qualities by noticing retinal wounds and their movement after some time long time. The Retmarker, made by Cunha-Vaz, looks at numerous CFPs gathered

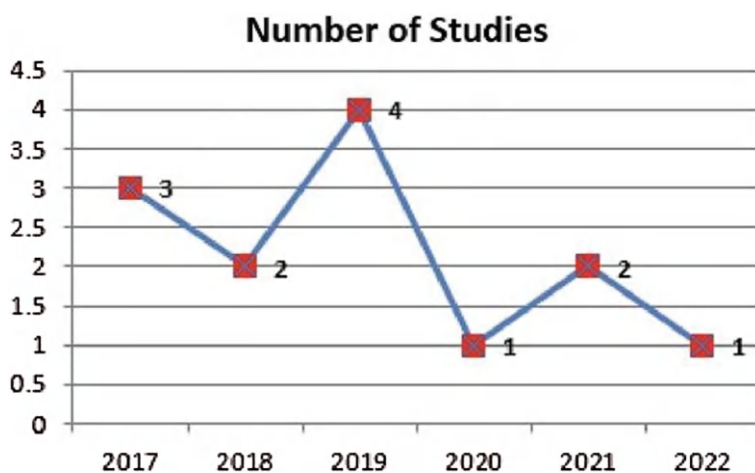


Fig. 2 Distribution of studies by year of publication

at various times from the meter, and in this, manner perceives the progressions happening in the retinal injury. Assessment shows that the quantity of microaneurysms (Mom) acquired from field 2 CFP can be utilized to foresee DR movement. These remarkable improvements in Mother's injury and income gauges are utilized to lay out a gamble profile in view of optical and non-optical gamble factors, e.g., Mom's development rate, exudate expansion, beat level, and HgA1c. Be that as it may, this model, because of its painstakingly collected remarkable plan, isn't a lot and is computationally costly because of its high dimensionality. The separation capacity is likewise incredibly low, dissimilar to utilizing profound learning feature extraction. Fundamentally, the RetinaRisk application made by Aspelund et al. A mathematical forecast model was utilized to catch the portability hazard of individuals with sight-harming retinopathy (STR) and timing of screening in view of every patient's gamble profile. Pointers for this stochastic assumption model were gotten from clinical epidemiological data on risk factors for the rate and movement of diabetic retinopathy from the Icelandic eye screening dataset.

Through the site <http://www.risk.is>, the DR movement forecast model recovers clinical data, coordinating diabetes type, diabetes term, direction, HbA1c or glucose level normal, circulatory pressure, as well as presence and seriousness from DR. This data is utilized to decide every individual's gamble of most exceedingly terrible visual STR in one year or less. The computation was performed by noticing the white Danish populace from the Nordic populace and had the option to foresee the DR development chances with an exactness of 76.

In any case, the estimation will for the most part misjudge the bet level, particularly for high bet levels, which can be because of different elements, for example, non-opportunity, non-linearity of pointers and credibility that a piece of patients with high-risk qualities might endure be blessed to receive foster extra gamble factors, for example, glucose and heartbeat. In like manner, many as of late found risk factors for vision-harming DR, for example, a past filled with other diabetes issues and weight were dropped from the informational index to give need to wagering calculators. Counting these wagering factors can demonstrate helpful while concentrating on forecast precision. At long last, the RetinaRisk estimation isn't pertinent to a large number of patients of various races and ethnicities.

The DAPHNE mechanized programming for retinal picture appraisal made by the College of Surrey in the UK can assess retinal pictures and really channel and show DR movement between pictures from early assessments advance and follow. In any case, the first item was not intended for this capacity; subsequently, there is a requirement for complete correspondence and coordination in clinical practice. Progressively, these applications are more savvy than manual strategies. Question 2: ML assumption model for DR development ML is reasonable for imagining DR developments due to its capacity to effectively find complex models concealed in the clinical records of endless patients and catch the confounding associations between them throughout a drawn out series. Subsequently, numerous tests have utilized simulated intelligence to additionally foster the satisfaction of assumptions. For instance, Tsao et al. fabricated a prescient model to recognize DR patients at high gamble of type 2 diabetes utilizing traditional simulated intelligence procedures including

support vector machines, determination trees choice, association of mind prosthesis and determined repeat. The discoveries showed that help vector machines (SVMs) accomplished better outcomes contrasted with other artificial intelligence calculations. As of now, Al-Sari et al., utilized the Erratic Woods classifier to foresee movement to diabetic kidney sickness (DKD) or DR following 5 years. Mikano et al., anticipated the development of the DKD stage a half year after the fact utilizing a repeat procedure. Zhou et al. combine customary ML strategies [Processed Overt repetitiveness (LR), Slope Backing Trees (GBT)] and DL to plan artificial intelligence models. In basically the same manner, a few researchers, Arcadu et al., Adal et al. and Bora et al. utilized the CNN Starting v33 design to decide how to drive prophet models.

Arcadu et al. made DL models to foresee patient seriousness and future movement of DR as at least 2 phases of DR deteriorate on the ETDRS inside a 2-year range following 6, 12 and 2 years from the standard visit. The Start v338 design with movement learning was utilized to drive the models. Model result is bunched by pair (0 = no ETDRS DRSS crumbling occasions at least two periods in months 6, 12 or 24 and 1 = ETDRS DRSS disintegration occasions at least two periods up in months 6, 12 or 24 or 24). Attribution maps are made to show regions where the model spotlights on picking how to portray explicit inquiry pictures.

Bora et al. constructed two kinds of profound learning structures (DLS) in light of Beginning v3 models, to foresee the advancement of “gentle or more extreme (“gentle+”)” DR in 2 years or less. This study has impediments, for example, the detachment of certain things to wander things, for example, cyclic pressure in the informational collections, that could be utilized for examination. Contrast DLS with these gamble factors or to adjust to it multivariably test. Also, there have been situations where the model misdiagnosed a few pictures as “DR-one-sided” with high exactness; the patient might have created DR. The audit recommended replies to these issues that could be investigated in future work, for example, working on motorized DR appraisal estimations joined with outline hardware risk. They likewise suggest exploring specific fields at a more significant level to distinguish wounds past the three fields that might stay undetected.

Adal et al. elevated a technique to investigate and quantify world-space longitudinal changes in microaneurysms and macular hemorrhages because of contrasts in brilliance between caught pictures. Taken at various distances. Fundus pictures are preprocessed utilizing enlightenment normalization and enlistment, helping CNN models rapidly learn visual highlights that intently look like retinal changes related with DR and consolidated rapidly. The outcomes show that CNNs arranged utilizing normalized enlightenment fundus pictures beat CNNs arranged on discrete complex and blue channel pictures.

Zhao et al. likewise draw in repetitive mind associations and long haul brief memory (RNN-LSTM) to advance hunch models. Leontidis et al. recommended computations by Boruta et al. and Takahashi et al. combined SVM classifier and DL techniques to additionally foster the DR association. In light of this, Hua et al. proposed a shrewd profound bimodal strategy to pick up utilizing a triple profound detour affiliation (Tri-SDN) to concentrate on recognizable proof of engine risk in

illness diabetic retinopathy. In like manner, a few researchers have likewise made other new ML computations, for example, Yan et al. pushed a model utilizing four (4) changed profound convolutional mind convolutions (DCNN). In this survey, CNN Starting v3 was utilized to isolate features from fundus pictures. The removed picture vector is taken care of into a completely connected layer to portray the ongoing seriousness of AMD. This seriousness joined with 52 free hereditary variations is a commitment to a more complete affiliation layer to foresee the probability of late improvement of AMD past the expected explicit years. Include maps were likewise made to show areas that frequently have added advantageous characteristics.

Furthermore, Chu et al. proposed a multi-cell CNN performing different errands (M 2 CNN) to assess diabetic retinopathy. The multi-cell configuration part is applied to bit by bit extend the profundity of DNN and further foster the motivation behind data representation. Consequently, lessen planning time and further develop portrayal precision. Perform multiple tasks learning configuration learns the connection among pictures and various moves toward anticipate names. M2CNN comprises of three modules. The primary module utilizes the Resnet-v2 initiator module. The following module chooses different mind network techniques utilizing the multi-cellular designing module. The third module performs picture design and features repeating organizations while applying the performing various tasks learning module. To additionally foster the enlightening imaging objective (lessening the variety of retinal fundus pictures), least pooling was applied to standardize the pictures.

Tehran et al. likewise pushed a two-level, multi-input profound conviction association (M2DBN) model to normally eliminate proper components from the retinal picture to conclude the degree of DR movement. The association can possibly get numerous commitments at two levels inside a profound association. Three portrayals, particularly useful pictures about the area, region and the entire world are extricated and given to the learning association. The proposed network relies upon tackling three significant issues connected with profound association. The first is the foundation of a worm association equipped for looking over divided pictures with independent commitments to taking care of the issue of handling immense pictures and lacking named data. It likewise resolves the issue of concentrating on the capacity of profound conviction organizations (DBNs) to utilize a two-layer profound conviction organization to oversee information from every one of the three unique layers for various pictures. To put it plainly, the inquiry is to construct the information design ahead of time while setting up the profound association. The outcomes of the uncovered association recommend that the proposed structure is probably going to be a strong grouping instrument. Sadly, the execution of the association isn't quick because of the picture parsing cycle, time and memory intricacy engaged with profound associations. This leaves a hole for future work on the quick execution of this association.

Length et al. proposed a non-selective prognostic model to foresee contamination relocation with non-balance timescales utilizing a start to finish profound learning procedure integrating point extraction grasped saliency (from 'picture') and autoregressive model. While past models anticipated future results, this model

predicts features that address what might be on the horizon, which are then delegated advancing or not advancing.

In the chose studies, different man-made intelligence cycles can be gathered into two classes, all the more explicitly traditional simulated intelligence and profound mind association, which are utilized to drive prescient models for DR development. 71% of these models utilized profound learning procedures that give special openness to traditional ML strategies, as displayed in Table 1. A gathering of three profound detour subsidiaries accomplished the best execution with 90.6% precision, reliability (96.5%), exactness (88.7%), lucidity (82.1%), and AUROC (88.8%). Just traditional ML models are fit for surveying DR movement. This might be because of Profound Learning's authoritative revelation issue. Concerning penetrability of estimating models, Yan et al. and Takahashi et al. models have produced remarkable guidelines for addressing areas that are frequently added to the normal characteristics by deciding the skewness of the subsequent layer for the data picture. Besides, outer endorsement was conceded for only one of these audits.

5 Datasets Available

To create a reliable model, a combination of private and publicly available datasets were utilized. Impressively, a significant 56% of the models were based on private datasets sourced from various countries around the world. While public datasets boasted the largest dataset size, both private and public datasets were utilized to ensure the model's accuracy and consistency. For a better understanding of the evaluation process, below is a list of the different datasets used in the assessments.

6 Conclusion

Diabetic Retinopathy (DR) is a moderate microvascular degenerative illness that prompts visual impairment or vision misfortune which is of most extreme concern internationally. Risk factors assume a huge part in the movement of DR and might have the option to make sense of the various paces of movement in various patients. Therefore, AI models have been proposed to actually comprehend and foresee DR movement in light of these gamble factors. This paper utilizes the SLR way to deal with select, orchestrate sum up and distinguish research difficulties and holes in momentum research for risk expectation for DR movement. Hence, this paper will be helpful for analysts with interest in the utilization of AI in DR guess in light of various modalities and utilizing on the holes recognized to concoct groundbreaking thoughts, advancements and innovation in this exploration region. It is suggested that another orderly survey can be done following a couple of years with additional investigations.

Table 1 List of datasets available

Paper	Dataset				Missing data handling	Data preprocessing
	Name	Sector	Description	Size		
1	Pilgrim hospital in Boston, UK	Private	127 subjects with diabetes that had progressed to DR 27 subjects with diabetes that had not progressed were collected for four (4) years	154	None	Image registration + image enhancement
2	Grampian Diabetes Database, UK	Private	Grampian Diabetes Database collected from within 4 periods of six (6) months intervals	156	None	N/A
3	Kyung Hee University, Medical Center, Seoul, South Korea	Private	Historical fundus photography EMR based record of DM patients	96	None	Color normalization, down sampling of image resolution, center-cropping of fundus images
4	Jichi Medical University, Japan	Private	9939 retrospective photographs of patients between May 2011 and June 2015	9936	None	None
5	RIDE and RISE studies	Public	Retrospective 7-field CFPs	N/A	None	None
6	EyePACS and 3 medical centers in Thailand	Public and Private	CFPs from 2 longitudinal datasets from EyePACS and 3 medical centres in Thailand 575,431 eyes, 7976 for internal validation and 4762 for external validation	575,431	None	Data augmentation

(continued)

Table 1 (continued)

Paper	Dataset				Missing data handling	Data preprocessing
	Name	Sector	Description	Size		
7	AREDS	Public	Genotype data, longitudinal color fundus photographs, and disease severity assessment over a period of 12 years from AREDS dataset	31,262	None	None
8	Rotterdam Eye Hospital	Private	Baseline and follow-up retinal images	51	None	None

7 Statement of Contending Interest

All creators have taken part in (a) origination and plan or examination and translation of the information; (b) drafting the article or changing it fundamentally for significant scholarly substance; what’s more (c) endorsement of the last variant. This composition has not been submitted to, nor is under audit at, another diary or other distributing setting. The creators have no connection with any association with an immediate or circuitous monetary interest in the topic examined in the composition. The creators have affiliations with associations with an immediate or circuitous monetary interest in the topic examined in the composition.

References

1. Amedo AO, Adade S, Koomson NY, Osae EA (2016) Impact of visual weakness on the personal satisfaction: a review of patients revealing at the low vision focus of the eastern provincial clinic of Ghana. *J Ophthalmic Sci* 1:01
2. Welp A, Woodbury RB, McCoy MA, Teutsch SM (2016) Making eye wellbeing a populace wellbeing basic: vision for tomorrow
3. Bourne RR, Flaxman SR, Braithwaite T, Cicinelli MV, Das A, Jonas JB et al (2017) Size, transient patterns, and projections of the worldwide commonness of visual deficiency and distance and close to vision debilitation: a precise survey and meta-examination. *Lancet Glob Wellbeing* 5:e888–e897
4. Maniadakis N, Konstantakopoulou E (2019) Cost viability of medicines for diabetic retinopathy: an orderly writing survey. *Pharmacoeconomics* 37:995–1010
5. Rodriguez NM, Aguilar S (2016) Pervasiveness of diabetic retinopathy in a center populace from Puerto Rico. *Optom Vis Sci* 93:750–753
6. Simó-Servat O, Hernández C, Simó R (2019) Diabetic retinopathy with regards to patients with diabetes. *Ophthalmic Res* 62:211–217

7. Chime DSW, Cheung GCM, Wong TY (2016) Diabetic retinopathy: worldwide predominance, significant gamble factors, screening practices and general wellbeing challenges: a survey. *Clin Investig Ophthalmol* 44:260–277
8. Koetting. The four phases of diabetic retinopathy. <https://modernod.com/articles/2019-june/the-four-phases-of-diabeticretinopathy?c4src=article=infinite-scroll>
9. Cunha-Vaz J, Ribeiro L, Costa M, Simó R (2017) Diabetic retinopathy aggregates of movement to macular edema: pooled investigation from autonomous longitudinal investigations of as long as 2 years' term. *Contribute Ophthalmol Vis Sci* 58:BIO206–BIO210
10. Jacoba CMP, Celi LA, Silva PS (2021) Biomarkers for movement in diabetic retinopathy: growing customized medication through coordination of computer based intelligence with electronic wellbeing records. *Courses in Ophthalmol* 36(4):250–257
11. Lin IC, Wang YH, Lin CL, Chang YJ, Lee SH, Wang IJ (2015) Diabetic polyneuropathy and the gamble of creating diabetic retinopathy: a cross country, populace based study. *Acta Ophthalmol* 93:713–718
12. Mohamed Q, Gillies MC, Wong TY (2007) The board of diabetic retinopathy: an orderly survey. *JAMA* 298:902–916
13. Pao SI, Lin HZ, Chien KH, Tai MC, Chen JT, Lin GM (2020) Recognition of diabetic retinopathy utilizing bichannel convolutional brain organization. *J Ophthalmol* 2020:1–7
14. Wong TY, Sabanayagam C (2019) The conflict on diabetic retinopathy: where could we presently be? *Asia-Pac J Ophthalmol* 8:448
15. Hemingway H, Croft P, Perel P, Hayden JA, Abrams K, Timmis A, Riley RD (2013) Forecast research procedure (PROGRESS) 1: a structure for investigating clinical results. *BMJ* 346:e5595
16. Lee CC, Perkins BA, Kayaniyl S, Harris SB, Retnakaran R, Gerstein HC et al (2015) Fringe neuropathy and nerve brokenness in people at high gamble for type 2 diabetes: the commitment associate. *Diab Care* 38:793–800
17. Schreur V, Ng H, Stefánsson Nijpels G, E, Tack CJ, Klevering BJ, (2021) Approval of a model for the expectation of retinopathy in people with type 1 diabetes. *Br J Ophthalmol* 105:1286–1288
18. Klein R, Klein BE, Greenery SE (2001) What number of steps of movement of diabetic retinopathy are significant? The Wisconsin Epidemiologic Investigation of Diabetic Retinopathy. *Curve Ophthalmol* 119:547–553
19. Schmalz G, Ziebolz D (2020) Changing the concentration to the entire patient rather than one oral infection: the idea of individualized counteraction. *Adv Prev Drug* 8(9):1620
20. Alfian G, Syafrudin M, Fitriyani NL, Anshari M, Stasa P, Svub J et al (2020) Profound brain network for anticipating diabetic retinopathy from risk factors. *Science* 8:1–19
21. Alfian G, Syafrudin M, Fitriyani NL, Anshari M, Stasa P, Svub J et al (2020) Profound brain network for anticipating diabetic retinopathy from risk factors. *Science* 8:1620
22. Hou Y, Cai Y, Jia Z, Shi S (2020) Risk variables and commonness of diabetic retinopathy: a convention for metainvestigation. *Medication* 99:e22695
23. Yousefi S, Pasquale LR, Boland MV (2020) Computerized reasoning and glaucoma: enlightening the black box. *Ophthalmol Glaucoma* 3:311–313
24. Tan KR, Seng JJB, Kwan YH, Chen YJ, Zainudin SB, Loh DHF et al (2021) Assessment of AI strategies created for expectation of diabetes difficulties: an orderly survey. *J Diabetes Sci Technol* 2021:19322968211056916
25. Xu Q, Wang L, Sansgiry S (2020) A precise writing survey of foreseeing diabetic retinopathy, nephropathy and neuropathy in patients with type 1 diabetes utilizing AI. *J Prescription Artif Intell* 3:6
26. Van der Heijden AA, Nijpels G, Badloe F, Lovejoy HL, Peelen LM, Feenstra TL et al (2020) Expectation models for improvement of retinopathy in individuals with type 2 diabetes: orderly survey and outside approval in a Dutch essential consideration setting. *Diabetologia* 63:1110–1119. Haider S, Sadiq SN, Moore D, Value MJ, Nirantharakumar K (2019) Prognostic expectation models for diabetic retinopathy movement: an orderly survey. *Eye* 33:702–713

27. Moher D, Liberati A, Tetzlaff J, Altman DG (2009) Favored detailing things for efficient surveys and meta-investigations: the prisma proclamation. *PLoS Drug* 6:e1000097
28. Moons KG, de Groot JA, Bouwmeester W, Vergouwe Y, Mallett S, Altman DG et al (2014) Basic evaluation and information extraction for methodical audits of expectation displaying studies: the CHARMS agenda. *PLoS Drug* 11:e1001744
29. Li H, Dong L, Zhou W (2023) Advancement and approval of clinical record-based calculated relapse and AI models to analyze diabetic retinopathy
30. Emon MU, Zannat R, Khatun T, Rahman M, Keya MS, Ohidujjaman (2021) Execution examination of diabetic retinopathy forecast utilizing AI models. In: *Procedures of the sixth global meeting on innovative calculation advances, ICICT 2021*, pp 1048–1052
31. Jagan MN, Murugan R, Goel T (2021) An original fourstep include determination procedure for diabetic retinopathy reviewing. *Phys Eng Sci Drug* 44:1351–1366
32. Dayana AM, Emmanuel WS (2022) An upgraded swarm advancement based profound brain network for diabetic retinopathy order in fundus pictures. *Multimed Instrum Appl* 81:20611–20642
33. Usman TM, Saheed YK, Ignace D, Nsang A (2023) Diabetic retinopathy recognition utilizing head part investigation multi-mark include extraction and order. *Int J Cogn Comput Eng* 4:78–88
34. Mushtaq Z, Ramzan MF, Ali S, Baseer S, Samad A, Husnain M (2022) Casting a ballot characterization based diabetes mellitus expectation utilizing hypertuned AI procedures. *Crowd Inf Syst* 2022:1–16
35. Tariq M, Palade V, Mama Y, Altahhan A (2023) Diabetic retinopathy identification utilizing move and support learning with compelling picture preprocessing and information expansion strategies. Hatzilygeroudis IK, Tsihrintzis GA. Jain LC (2023) *Fusion of AI paradigms. Intelligent Frameworks Reference Library*.
36. Mohammad Shorfuzzaman M, Shamim Hossain AES (2021) A logical profound learning troupe model for vigorous finding of diabetic retinopathy evaluating. *ACM Trans Interact Media Comput Commun Appl* 17:1–24
37. Chime DSW, Peng L, Varadarajan AV, Keane PA, Burlina PM, Chiang MF et al (2019) Profound learning in ophthalmology: the specialized and clinical contemplations. *Prog Retin Eye Res* 72:100759

Sentimental Analysis Using Machine-Learning Models—A Comparative Study



Veda Spandana Tungala, Aarthi Hasitha Yepuri, Sai Sindhuja Anaparthi, Aasritha Sai Vemulapalli, and S. R. Reeja

Abstract E-commerce is integral today, shaped by technology. Reviews on platforms like Myntra and Flipkart are vital, which are classified as positive or negative. Technology-driven online commerce exceeds traditional retail, using sentiment analysis with advanced models for client feedback. Sentiment analysis through NLP proves valuable in classifying customer feedback as positive or negative. To improve classification efficiency SVM, Naive Bayesian, and Decision Tree are frequently utilized. Improvement of the analysis of sentiment can be done in various ways for example, by using a technique that considers word semantics at the sentence level as well as within the specific product domain. The discussed methods enable the designer to quickly determine the crucial design standards for future product development by evaluating user-generated information. This paper discusses several methods and approaches for sentiment analysis in great detail.

Keywords E-commerce · Sentiment analysis · NLP · SVM · Naive Bayesian · Decision Tree

1 Introduction

Sentiment is the subjective or emotional attitude that is conveyed in a speech, text, or other form of communication. It expresses the authors or speaker's thoughts, beliefs, or attitudes regarding a certain issue, circumstance, or topic.. Sentiment can be broadly classified into three classes: neutral, negative, and positive. Sentiment analysis is a technique of automatically determining if text data contains either a positive or a negative emotional sentiment. The text must be prepared, turned into numerical features, a suitable machine-learning model must be chosen, and the model must be trained on labeled data in order to identify sentiment patterns. After being

V. S. Tungala · A. H. Yepuri · S. S. Anaparthi · A. S. Vemulapalli · S. R. Reeja (✉)
VIT-AP University, Andhra Pradesh, India
e-mail: reeja.sr@vitap.ac.in

trained, the model can assess the sentiment of text, providing insightful information for uses like analyzing consumer comments, keeping an eye on brands, and evaluating public opinion. For the model to be accurate and useful in real-world situations, it must be continuously improved and adjusted to changing language patterns. Researchers have explored sentiment analysis using an incredible range of techniques and technology by utilizing the data-rich e-commerce environment. This toolkit includes advanced Natural Language Processing (NLP) models such as BERT, Bi-LSTM, and SVM, as well as traditional Machine-Learning techniques like Naive Bayes and Support Vector Machines. Sentiment analysis must balance data-driven insights with the obligation to protect privacy and ethical integrity, serving not only the interests of organizations but also the greater good.

2 Literature Survey

Many e-commerce websites are flourishing and becoming an integral part of our lives in the modern world. Earlier, people would visit a store to make purchases, but today's technology makes purchases more convenient. Evaluations of people play a significant role and affect purchase behavior as technology advances. People use websites like Myntra, Meesho, and Flipkart to display reviews that customers provide based on their interactions with the products and their overall satisfaction. Because a lot of individuals have written reviews about the product. So, from a big collection of reviews, we will separate the positive and negative comments.. Classifying the reviews into positive and negative categories is done through the use of different Machine-Learning Models in the study [1]. Sentimental analysis has shown to be a useful tool for categorizing evaluations in recent years, whether they are favorable or unfavorable depending on the contentment of the customer. Here, the sentimental analysis of the customer valuations is done using NLP (Natural Language Processing). In-depth explanations of the procedure are provided along with a suggested general method for categorizing sentiment polarity. Data for the study were acquired from Amazon.com's online customer reviews of various goods. Both review-level and sentence-level categorization trials are carried out with good outcomes [2]. In recent years, e-commerce has grown remarkably. The amount of product reviews written online has increased as a result of more consumers making purchases online. The viewpoints of other customers influence the consumer's assessment of the products; implicit views in customer evaluations play a major role in the customer's decision to purchase. This study examines sentiment categorization using several machine-learning approaches utilizing the Amazon reviews dataset. We chose the model that performed best and retrained it for the binary classification after running the experiment on multiclass classifications [3]. Classification can be used as a means for sentimental analysis and the crucial tasks are categorization of text, recognition of tone, and classification of image, etc. The majority of the currently used supervised classification techniques are based on conventional statistics, which can yield optimal outcomes. The primary goals are to improve accuracy and inform

the manufacturer of any problems with the product. The division of sentiment is the major challenge with sentiment assessment. When trying to categorize the reviews in accordance with their unique rating in relation to star-scale, it becomes difficult. Additionally, categorization issues in implicit-level sentiment analysis exist at the review-level. SVM, Decision Tree and Naive Bayesian are mostly used to increase classification efficiency. The suggested system uses the Amazon Dataset as a dataset to increase the Turney algorithm's precision. In comparison to other classification techniques, Semantic Orientation (SO) using Pointwise Mutual Information produces satisfactory results. In the case of a positive average SO, the review level is considered an excellent value. However, when a negative average SO is achieved, the review level becomes negative [4]. The process of mining data, for the aim of estimating the emotion of the comment using natural language processing is known as sentiment evaluation. The text is divided into positive, neutral and negative by the sentiment evaluation. The sentiment assessment examines the data, identifying the "greater" and "lesser" sentiment as either positive or negative, respectively. As a result, over the past few decades, the internet has evolved into an essential repository of unprocessed information. The growing raw data offers an extremely abundant reservoir of information for any kind of decision-making process. Sentiment analysis is a discipline that has grown automated methods for analyzing such enormous amounts of information. The main intent of sentiment evaluation is to categorize data by identifying its degree of polarity. Despite the fact that sentiment analysis is a based-on text procedure, it may be difficult to determine a phrase's precise sentiment. It indicates that in order to surpass any contemporary method utilized to evaluate sentence polarity, an improved approach must be established [5]. An essential task for Natural Language Processing (NLP) is opinion extraction and sentiment examination. It tracks each user's opinions, viewpoints about the product. This provides an opportunity for companies to implement the appropriate product changes to improve satisfaction among consumers. A great deal of the sentiment review methods that are currently utilized for such online commodities are extensively mistaken and require quite a bit of training. With the aid of Deep Learning Modified Neural Networks (DLMNN), an approach has been suggested for the sentiment extraction of internet product reviews. Additionally, a technique for forecasting e-commerce products is recommended implementing the Improved Adaptive Neuro-Fuzzy Inferences System (IANFIS) [6]. Consumers these days mostly rely on E-commerce platform reviews and ratings when buying a product. Popular e-commerce businesses like Amazon, Flipkart, and Myntra etc. enable customers to share their experience with the product, and also customers can share the proof of product received. To understand the insightful information from a large number of reviews, first, the data needs to be classified as both positive and negative reviews. Naïve Bayes, SVM, and LSTM were chosen to categorize the reviews. The present SA methods used for these customer product reviews have low accuracy and models take a long time to train [7]. During the COVID-19 lockdown, electronic commerce has played a significant role in India. One of the necessities that everyone needed during lockdown was groceries and other food items. During the pandemic, online grocery store sales increased drastically, and businesses had many problems when delivering groceries to customers.

Their capacity to deliver products on time while maintaining consumer happiness becomes quite challenging. A lot of consumer reviews had been posted online at that time on many different sites. One of the most significant methods of analyzing internet reviews is sentiment analysis. This study mainly focused on analyzing those customer reviews. The methods used to classify the reviews are TF-IDF followed by naïve Bayes classification. The model developed helps grocery store websites handle the large volume of reviews given to products and classify them based on their feedback [7]. The usage of social networking sites like Twitter has increased during the past years. These websites allow users to share their ideas and opinions. To assess the public's view on a specific issue of interest, sentiment analysis of such data, which includes opinions from individuals, is crucial. For fast data retrieval, Twitter also provides developer-friendly streaming API which enables the search of real-time tweets from multiple individuals. Twitter only allows a word limit of 140 characters for its users. On Twitter tweets, sentiment analysis is employed for Lexical, Machine-learning-based analysis, and Hybrid analysis. The studies that are provided are divided into groups based on the methodology they use. The disciplines of sentiment analysis on Twitter, such as extracting opinions, tracking sentiments and emotions, and detecting irony and non-irony tweets are also covered. These tasks have lately drawn more attention [8]. This study's goal is to evaluate the degree of customer satisfaction regarding the services offered by Traveloka by analyzing the number of satisfied and dissatisfied individuals. The analysis process was conducted using the Scikit-learn library in Python. In this research, six classification techniques were employed as the base for comparing: Naive Bayes, Logistic Regression, and Support Vector Model (SVM). Retrieving data, transforming data, training classification models, forecasting test data, and finally analyzing the outcomes were all part of the research process. The findings imply that SVM performs exceptionally well at accurately interpreting the sentiment of tweets about Traveloka [9]. E-commerce has a global worth of \$13 trillion and is projected to grow to \$55.6 trillion by 2027, according to Business Wire research. Reviews of products and services play a major role in influencing our purchasing decisions in this quickly growing industry. But it can be challenging to manually evaluate reviews in order to study business models and come to wise judgements. Based on customer reviews and evaluations, Natural Language Processing (NLP) is used to classify these products. Predicting paid and unpaid reviews is possible with NLP and machine learning. Differentiating between "Positive," "Neutral," "Negative," and "Indifferent" phrases is the main focus of the suggested module. Following the data analysis, the assumptions categorized as "better" are labeled as positive, while those categorized as "worse" are classified as negative. The automation process yielded an impressive validation accuracy by employing Fast Text for word embedding in combination with a Multi-channel Convolutional Neural Network [10]. The paper performs a thorough review of sentiment classification techniques, emphasizing lexicon-based and supervised machine-learning algorithms that use word frequency-inverse document frequency vectorization. The goal of the study is to ascertain which strategy works best for which kinds of problems. SVM, Gradient Boosting and Logistic Regression are the three machine-learning techniques that are investigated. Conversely, SentiWordNet,

VADER Pattern, and Sent WordNet are the three lexicon-based algorithms investigated. Our tests unequivocally demonstrate that, across all evaluated criteria, the three machine-learning models’ accuracy consistently dominates the lexicon-based models. Our machine-learning results demonstrate an improvement when compared to previous efforts on text sentiment analysis [11].

The summary is shown in Table 2.

3 Performance Analysis

In Table 1, the performance of various machine-learning models is assessed based on accuracy scores. Notably, both Support Vector Machine (SVM) and Logistic Regression emerge as top performers, each achieving an impressive accuracy rate of 92%. This signifies a robust predictive capability exhibited by both models. The results suggest that SVM and Logistic Regression outshine their counterparts, including Naive Bayes, Decision Tree, Random Forest, and KNN, which exhibit accuracy scores ranging from 87 to 91%. Such consistent high accuracy rates for SVM and Logistic Regression underscore their reliability and effectiveness in handling the underlying dataset.

In Fig. 1, the graph illustrates the comparative performance of various machine-learning models across key metrics, encompassing accuracy, precision, recall, and F1 score. These models, as detailed in Table 1, were subjected to a comprehensive analysis. Examining the outcomes of these performance metrics, including accuracy, precision, recall, and F1 score, reveals a notable similarity in the models’ effectiveness.

Determining the superior model is not solely reliant on direct result analysis, as observed performance trends exhibit resemblances among the models. Instead, the choice of a model hinges on specific objectives and constraints. Notably, the initial data preprocessing step involves the removal of missing rows, a fundamental practice that can contribute to higher-than-expected accuracy values. For future investigations, it is recommended to enhance data preprocessing with advanced techniques like Text Tokenization, Stopword Removal, Stemming, and the incorporation of Embedding Techniques such as Word2Vec and BERT. Tailoring model selection to

Table 1 Table depicting performance analysis of machine-learning Models

Machine learning Models	Accuracy
Naive Bayes	90
SVM	92
Decision Tree	87
Logistic Regression	92
Random Forest	91
KNN	87

Table 2 An overview of sentimental analysis comparing different machine-learning models

S. no.	Data set	Features	Methods	Performance matrix	Advantages	Disadvantages
1	The online store Amazon.com: :400,000 reviews in the data set	Name of Product, Brand, Price, Rating, Reviews, and Examine the votes	Naive's Bayes SVM Decision Tree	Accuracy	Sentimental analysis helps in classifying reviews into positive and negative. So, it provides a better understanding of customer choices	While online review sentiment analysis may increase consumer trust, it has drawbacks such as accuracy issues, oversimplified binary classifications, data bias, and ethical issues
				66.95		
				81.77		
				74.75		
2	The online retailer Amazon.com: Reviews of 20,062 products	ID of Reviewer and Product, Ratings of Product, Review Time, Helpfulness factor, Text Review	Naive's Bayes SVM Random Forest	Averaged F1-Score	Uses sentiment evaluation with the NLP approach, demonstrating the incorporation of cutting-edge technologies	This approach might not be effective for reviews that just include implicit feelings
				Level of Sentence Classification: 80%		
				Level of review classification: 73%		
				CrossLoss, A, P, R, F1		
3	The online store Amazon.com.: 413,840 reviews and 6 features	Name of Product, Brand, Price, Rating, Reviews and Examine the votes	Logistic Regression Random-Forest	59.6, 85.3, 86.3, 85.3, 85.7	Proposed new types of algorithms like BERT and Bi-LSTM	There are some Limitations in this study due to Limited Resources in the Laptop
				40.3, 89.9, 90.1, 89.9, 89.7		
				11.59, 78.4, 78.5, 78.4, 77.9		
			Naive's Bayes BERT	18.9, 94.7, 90.8, 94.7, 94.6		
				23.4, 93.3, 92.9, 93.3, 92.9		
			Bi-LSTM			

(continued)

Table 2 (continued)

S. no.	Data set	Features	Methods	Performance matrix	Advantages	Disadvantages
4	Amazon data set: 2500 positive reviews and 2500 negative reviews	Text, ID of reviewer, product rating, rating of a product, date and time, Usefulness	SVM	Accuracy 69	Proposed new algorithm based on turnery algorithm which can increase the accuracy	A large data set is required
			Naive Bayes	69		
			Decision tree	69		
			Turnery	78		
5	50,000 movie reviews (Stanford data set), half positive and half negative	Review text is the main feature and later categorized into adverbs, adjectives, and verbs	Naive Bayes	Accuracy 89.855	This new method takes less time to train and test opinions than other existing methods, and increases the accuracy	Gives good accuracy only big data sets are taken. Accuracy is not good when a data set with 5000 entries is taken
			Logistic Regression	89.575		
			Linear support vector classifier	89.36		
			Decision Tree	87.785		
6	Food Review data set (5 lakhs records)	Product and User Id, Name of Profile, Helpfulness of Denominator and Numerator, Score Time Summary of Text	DLMNN: CB	Accuracy 89.7764	When compared to the current ANFIS, the suggested IANFIS achieves better results	Main-word processing only determines the feeling expressed in a specific word; it cannot give data required to understand the complete meaning
			GB	96.8543		
			CLB	95.7764		
			IANFIS	93.7734		
7	Amazon product dataset which comprises of 24,352 entries	reviewerId, reviewer name, review text, Overall Summary, review time		A, P, R, F1	The performance metrics of the BERT analysis used produced better results than the other ML Algorithms	The accuracy is good but the algorithms like LSTM have a few disadvantages which is it takes a long time to train. They are also not completely bidirectional
			NB	80.12, 79.32, 73.57, 69.32		
			SVM	81.33, 82.68, 84.31, 81.26		
			LSTM	83.97, 80.57, 79.28, 82.34		

(continued)

Table 2 (continued)

S. no.	Data set	Features	Methods	Performance matrix	Advantages	Disadvantages
8	A total of 1000 data has been collected from BigBasket, Grofers, Spencers, and Amazon from Jan 2020–Dec 2020	Text reviews of the consumers And Star ratings (1-star or 5-star)	BER0054	88.48, 88.09, 86.22, 89.42	The model that was developed is a classification model that uses ratings of 1 and 5. Sentiment analysis is performed using TF-IDF values	In this study, which was based on internet user ratings, only 1 and 5-star ratings were employed 2, 3, and 4-star ratings were not considered
			Naïve classification Data (Training)	A, P, R, F1 99, 91.2, 93.9, 92.5		
			Data (Testing)	86, 76.7, 76.2, 76.4		
				Accuracy (%) 57.7 75–77		
9	Data to be analyzed is collected from Twitter posts	Total no of positive and negative terms, Length of a tweet, SVMs		Accuracy (%)	In terms of performance, the hybrid approach has so far demonstrated positive sentiment. Which encourages researchers to explore the area	The Comparison of all methods showed that machine learning methods yielded the best outcomes, whereas lexical methods produced the worst
			Lexical	57.7		
			Machine Learning(SVM, Naive Bayes)	75–77		
			Hybrid	84		
10	1200 Tweets about Traveloka which consists of 690 positive tweets and 510 negative tweets	Text Review Review Id Name Product Name Product Price		Accuracy	This study establishes a groundwork for future research in this field, potentially paving the way for even more precise and efficient sentiment analysis techniques	Small size of the dataset used and study only focuses on one specific context and language
			Logistic Regression	The SVM has highest models, with an accuracy of 84.88%		
			Native Bayes			

(continued)

Table 2 (continued)

S. no.	Data set	Features	Methods	Performance matrix	Advantages	Disadvantages
11	Amazon's product review dataset consists of 34,660	reviewerId reviewer Product Name Brand Name Categories Reviews Texts Rating	SVM	84.88%	The study examined four well-known supervised classifiers and analyzed crucial grammatical sections using a sizable dataset of Amazon product reviews	The study did not consider neutrality, which may limit the generalizability of the findings to reviews that are not overtly positive or negative
			Glove	Accuracy BERT RMDL model: accuracy = 97.07% and validation accuracy = 78.84%, while the ELMo LSTN model: accuracy = 76.46%		
			BERT			
			Elmo			
			Fast text			
12	43,620 reviews for a variety of 1,000 goods on Amazon	ReviewerId ReviewerName Ratings of Product Review Text	SVM	Accuracy (%) 89	Machine Learning approaches can handle complex relationships between features and labels, and can be used for a wide range of tasks beyond sentiment analysis. Lexicon-based approaches can be used for languages and domains where labeled data is scarce	Large amounts of labeled data are required for training in machine learning approaches for sentiment analysis, and there is a risk of overfitting or underfitting as well as challenges in interpreting the results
			Gradient boosting	87		
			Linear regression	90		
			Pattern	69		
			VADER	83		
			SentiWordNet	80		

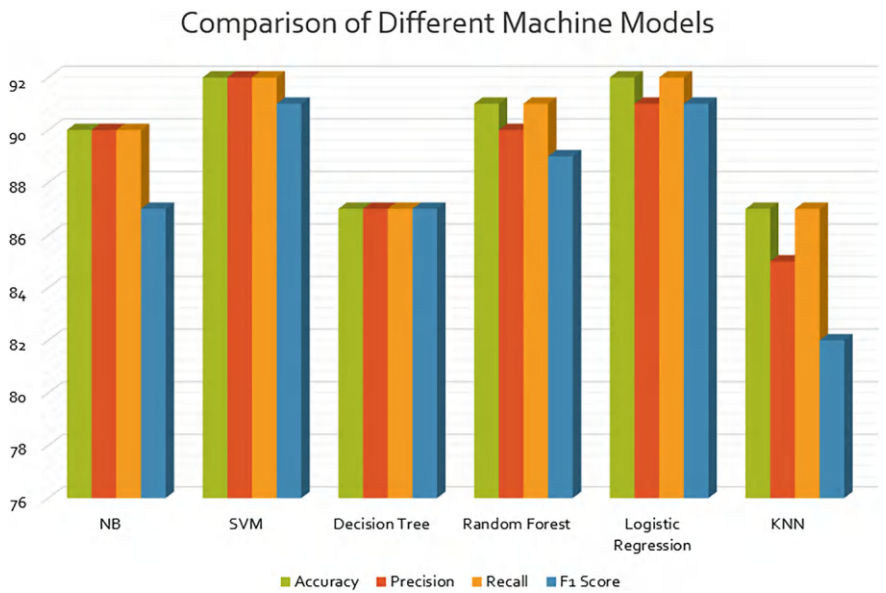


Fig. 1 Performance analysis of machine-learning models

specific goals and constraints is paramount. If interpretability is a priority, Decision Tree emerges as a favorable option. Conversely, for a balance of robustness and generalization, Random Forest proves advantageous. When considering a model with a focus on interpretability, robustness, generalization, and minimized training time, Support Vector Machines (SVM) and Logistic Regression stand out as prudent choices (Table 2).

4 Conclusion and Future Scope

This paper encompasses a wide array of studies that go deep into sentiment analysis across diverse domains and datasets. Each of these studies employs various domains and distinct features, methods, and performance metrics, offering unique insights that collectively shape the sentiment analysis landscape. The studies leverage information from industry leaders like Amazon.com and dangdang.com, emphasizing the importance of this area in understanding consumer preferences, improving user satisfaction, and appropriately categorizing evaluations into positive and negative sentiments. The difficulties faced during the analysis are worsened by the growing integration of sentiment analysis methods into online review websites, demanding ongoing research and improvement. The efficiency of sentiment analysis is mostly dependent on the type of algorithm, whether classical ML-based or lexicon-based along with features, including review text, product names, and brand names. Based

on the above comparative study we can conclude that machine learning models like SVM and Logistic Regression (with accuracy of 92) perform better compared to other models. Utilizing advanced models like BERT, LSTM etc. are recommended for enhancing the efficiency of sentimental analysis. In conclusion, this strategic approach to model selection underscores the significance of aligning machine-learning choices with project goals and priorities, advocating for a nuanced consideration of both traditional and advanced preprocessing techniques for comprehensive and informed decision-making.

In future sentiment analysis endeavors, there is a need to prioritize robust model development capable of navigating diverse language expressions within customer reviews. Strategic exploration of techniques for handling imbalanced datasets is imperative to ensure fair and accurate model training. Furthermore, integrating advanced preprocessing methods, including Stemming, Tokenization, and Word Embeddings, becomes pivotal for refining contextual understanding in sentiment analysis, enabling a more nuanced and accurate interpretation of sentiments in a continually evolving research landscape.

References

1. Singla Z, Randhawa S, Jain S (2017) Sentiment analysis of customer product reviews using machine learning. In: 2017 International conference on intelligent computing and control (I2C2), Coimbatore, India, pp 1–5. <https://doi.org/10.1109/I2C2.2017.8321910>
2. Fang X, Zhan J (2015) Sentiment analysis using product review data. *J Big Data* 2:5. <https://doi.org/10.1186/s40537-015-0015-2>
3. AlQahtani ASM (2021) Product sentiment analysis for amazon reviews (2021). *Int J Comp Sci Inf Technol (IJCSIT)* 13(3). Available at SSRN: <https://ssrn.com/abstract=3886135>
4. Rajesh Kanna P, Pandiaraja P (2019) An efficient sentiment analysis approach for product review using Turney Algorithm. *Procedia Computer Science* 165:356–362. ISSN. 1877–0509. <https://doi.org/10.1016/j.procs.2020.01.038>
5. Sultana N, Kumar P, Patra M, Chandra S, Alam Sk (2019) Sentiment analysis for product review. *Int J Soft Comput* 09(7). <https://doi.org/10.21917/ijsc.2019.0266>.
6. Sasikala P, Mary Immaculate Sheela L (2020) Sentiment analysis of online product reviews using DLMNN and future prediction of online product using IANFIS. *J Big Data*. 7(33). <https://doi.org/10.1186/s40537-020-00308-7>
7. Panda L, Sri KB, SR R (2023) Real time pothole detection system—an application facilitating public safety. In: 2023 International conference on intelligent and innovative technologies in computing, electrical and electronics (IITCEE), Bengaluru, India, pp 389–395, <https://doi.org/10.1109/IITCEE57236.2023.10091000>
8. Geetha MP, Karthika Renuka D (2021) Improving the performance of aspect based sentiment analysis using fine-tuned Bert base uncased model. *Int J Intell Net* 2. <https://doi.org/10.1016/j.ijin.2021.06.005>
9. Ng CY, Law K, Ip WH (2021) Assessing public opinions of products through sentiment analysis: product satisfaction assessment by sentiment analysis. *J Organizat End User Comput* 33:125–141. <https://doi.org/10.4018/JOEUC20210701.oa6>
10. Dickson ZA, Prakoso MRB, Putra MSQ, Syaputra MSAF, Achmad S, Sutoyo R (2023) Sentiment analysis for customer review: case study of Traveloka. *Procedia Comp Sci* 216:682–690. <https://doi.org/10.1016/j.procs.2022.12.184>

11. Singh U, Saraswat A, Azad HK et al (2022) Towards improving e-commerce customer review analysis for sentiment detection. *Sci Rep* 12:21983. <https://doi.org/10.1038/s41598-022-26432-3>
12. Majid A, Nugraha D, Adhinata FD (2023) Sentiment analysis on TikTok application reviews using natural language processing approach. *J Embedded Syst Security Intell Syst*

Real-Time Object Detection and Classification Using Deep Learning Models



Piyush Kumar Pareek , B. R. Shobha Rani , and S. Bharathi

Abstract Real-time object detection is examined to be one of the hardest and complex technologies of computer vision field. It detects objects from the live videos, classifies frames and identifies the detected frames from the input. In past few years many models have been designed which had few limitations like specific object detection, slow processing, low level accuracy and many more. The present Single Shot Multibox Detector (SSD) model overcomes all the limitations. The model is fast and generates accurate outputs with minimum loss in detection. TensorFlow object detection API framework is built above SSD algorithm trained on COCO dataset. SSD model uses MobileNet as backbone to extract feature maps from the live input frames. CNN classifiers predict objects from the extracted feature maps with detected object class name and bounding box with class probability. The model has obtained an accuracy of 97.6% trained at 100 epochs when compared with other object detection models such as YOLO and R-CNN. SSD model is identified as good model with height accuracy and real-time speed in detecting even the smaller objects.

Keywords Object detection · SSD · CNN · Loss function · Python-Keras

1 Introduction

Real-time object detection is driven by the growing need for intelligent systems that can swiftly and precisely interpret visual data for dynamic decision-making. Because real-time object detection gives blind people information about their surroundings, it can help them tremendously in their daily activities. Real-time object detection can be incorporated into assistive technologies to help blind people access information

P. K. Pareek
Nitte Meenakshi Institute of Technology, Bengaluru, India

B. R. Shobha Rani (✉) · S. Bharathi
Dr. Ambedkar Institute of Technology, Bengaluru, India
e-mail: shobhakrishna8@gmail.com

about their surroundings that would otherwise be difficult or impossible to identify, giving them more independence and a higher quality of life overall.

In the field of real-time object detection, deep learning algorithms have become game-changing instruments that have completely changed our capacity to recognise and locate objects automatically in a stream of data, such as pictures or video frames. These algorithms have built-in features that help explain why they work so well in real-time situations, especially Convolutional Neural Networks (CNNs). Deep learning algorithms [1] and improved hardware performance have led to notable advancements in object detection in recent years. The performance of object detection has significantly improved with the advent of Convolutional Neural Networks (CNNs).

Deep learning is useful in a wide range of fields and applications [2] such as autonomous vehicles, medical image analysis, generative modelling, reinforcement learning, human activity recognition with image classification, object recognition, segmentation and object detection and tracking.

Single Shot Multibox Detector (SSD) and You Only Look Once (YOLO) are two examples of deep learning models built for real-time processing that balance speed and accuracy. In situations where quick decision-making is crucial, these models maximise the trade-off and guarantee prompt reactions.

The main goal of employing the SSD model for real-time object detection is to offer a reliable and effective solution that maintains a balance between speed and accuracy. Consequently, it is well-suited for a range of applications, such as robotics, autonomous vehicles, surveillance, and more, that call for the quick and accurate identification of objects in pictures or video streams. It is designed to handle objects at different scales and aspect ratios within an image. This adaptability is essential for detecting objects of various sizes and orientations in diverse environments. SSD is widely used for object detection due to its real-time capabilities, multi-class detection, accurate localisation, and adaptability to various scenarios, making it a valuable tool in computer vision applications.

This paper firstly discusses introduction section on object detection which brings a combination of accuracy, adaptability, scalability, and innovation, making it a fundamental technology in addressing complex computer vision tasks across a wide range of applications. The next section is about related works which lists different algorithms and architectures used for object detection and comparison of their performance is shown. This is followed by methodology section about implementing SSD model and its flow of detecting objects. Further, section on practical implementation of SSD model using Python libraries is discussed finally with results and conclusion section.

2 Related Work

The goal of this work [3] is to provide an overview and analysis of every idea that went into creating neural network and its models. Additionally, they have given greater attention to the GoogleNet models and the comprehensive Inception modules. The intriguing uses of deep learning and the debate about its essential elements have been discussed and shown. Students and academics interested in deep learning and related models will undoubtedly take notice of this publication.

In order to extract and fuse multi-scale features [4] and produce high-level features with rich semantic information, the feature extraction network is first constructed by integrating the hourglass structure network with the attention mechanism layer. Secondly, the system does multi-task learning by using the semantic segmentation task as an auxiliary task. Lastly, the object's position and categorisation are predicted using multi-scale characteristics. According to the testing results, the algorithm performs consistently better than the other three comparison algorithms and significantly improves object detection performance. Moreover, the detection speed may exceed real-time, making it suitable for real-time detection.

In [5] work in corporation of the extracted features and bounding boxes. SSD allows more aspect ratios for generating default bounding boxes around the objects detected. SSD boxes can wrap around the objects in a tighter and more accurate manner. With the help of this paper, they present the analysis and implementation of real-time object detection using SSD which is one of the fastest object detection algorithms. It explains the performance of real-time object detection using SSD with MobileNet v1 which is faster and efficient than traditional methods of object detection.

The paper [6] discussed about SSD model performs faster than YOLO model. The produced space of bounding boxes has been divided in this research into a collection of default boxes with varying aspect ratios and sizes for each feature map point. When it comes to prediction time, the network creates values for each class type that is included in each default box and adjusts the box to better fit the shape of the object. In a process to identify items of smaller scale and to generate more predictions, they blend default boxes with multi-scale feature maps.

The suggested study [7] uses three distinct object identification algorithms (You Only Look Once, or YOLO), Single Stage Detector (SSD), and Faster Region-Based Convolutional Neural Networks (R-CNN)) to try and tackle these problems. This research seeks to determine the optimal trade-off between feature and accuracy by comparing three distinct deep learning object identification techniques. In terms of accuracy, recall, precision, and loss, R-CNN object identification algorithms outperform single stage detectors such as You Only Look Once (YOLO) and Single Shot Detector (SSD).

Below table gives the information on objective and gaps identified in different neural network models.

Aspect	YOLO	R-CNN	SSD
Object Detection Objective	Real-time object detection	Sequential data processing,	Real-time object detection
Bounding box prediction	Grid-based prediction	Sequential prediction	Multi-scale feature maps
Strengths	High accuracy and real-time performance	Captures temporal dependencies in sequential data	Effective multi-scale detection
Weaknesses	May struggle with small objects	May have difficulty learning long-term dependencies	Sensitive to object aspect ratios
Gaps identified	May struggle with small and densely packed objects	Difficulty with long-term dependencies in training	Sensitive to object aspect ratios and scales

3 Methodology

Prior to being analysed by the model, the input image is first preprocessed. Resize, normalise, and data augmentation are a few of the tasks that could be included in this. A high-level abstraction of the imagery is then represented by feature maps that are generated by processing the preprocessed image through the model.

The workflow of real-time object detection using anSingle Shot MultiBox Detector (SSD) model can be summarised as follows:

Input Image: The first step is to provide an input image or a video frame to the SSD model. This image was captured from a camera or extracted from a pre-recorded video.

Preprocessing: The input images were preprocessed to make compatible with the SSD model. These images were resized and normalised with a fixed size of pixels.

Model Inference: The preprocessed images are then fed to the SSD model. The SSD model was trained on a large annotated dataset to identify different objects. It consists of a MobileNet as backbone network followed by a series of convolutional layers.

Feature Extraction: The backbone network further processes the image and extracts a set of high-level feature maps. These feature maps capture relevant information about objects present in the image.

Anchor Generation: The SSD model uses a set of predefined anchor boxes at different scales and aspect ratios. These anchor boxes act as reference boxes that the model will use to detect objects. They are generated based on the feature maps output from the backbone network.

Predictions: The feature maps are passed into additional convolutional layers, which are responsible for predicting the class probabilities and bounding box coordinates for each anchor box. This is done using a combination of classification and regression layer.

Non-maximum Suppression: The model generates a large number of bounding box predictions for various anchor boxes across the objects in the image. To eliminate duplicates and keep only the most confident detections, a non-maximum suppression algorithm is applied. This algorithm selects the boxes with the highest probabilities and discards overlapping boxes on identified object.

Post-processing: The final output of the model consists of a set of bounding boxes, each labelled with a class probability. These bounding boxes are post-processed to filter out low-confidence detections and adjust them for the original image size and aspect ratio.

Visualising Results: The bounding boxes are overlaid on the original image to visualise the detected objects. Each bounding box is labelled with the predicted class and its corresponding confidence score.

An image is taken as input and Single Shot Multibox Detector (SSD) algorithm is applied. The image is divided into grids. The size and complexity of the image determine what number of grids are used. Once the image is divided into grids, each grid undergoes through many layers of MobileNet model to extract feature maps. Multiple 3*3 convolution filters are used to these feature maps for further analysis. These feature maps are further processed through many 3*3 convolution filters. Each grid contains confidence score and bounding box offset. Confidence score is 1 if grid contains some object and 0 if grid contains no object. Prediction of bounding box is explained in Fig. 1

The anchor boxes, also known as default boxes, are applied to the feature maps after the image has been processed run through the model in order to generate a set of possible bounding boxes for the objects in the image. In order to ascertain the class of the object attached in each bounding box, a classifier is applied to each box. Furthermore, non-maximal suppression is used to eliminate overlapping boxes and improve the final set of bounding boxes. Finally, the process generates an array of bounding boxes along with the class probabilities that correspond to them.

3.1 *Choosing Default Bounding Box*

In a Single Shot Multibox Detector (SSD) model, selecting default bounding boxes involves setting up a set of anchor boxes with particular scales and aspect ratios. The formulas used to determine the default bounding boxes' height and width depend on scales, aspect ratios, and the anchor box's base dimensions.

The height (h_i) and width (w_i) of the default bounding box for each combination of aspect ratio and scale can be calculated as follows in Eqs. (1) and (2):

$$w_i = \text{basewidth} * \text{scale} * \sqrt{\text{aspect ratio}} \quad (1)$$

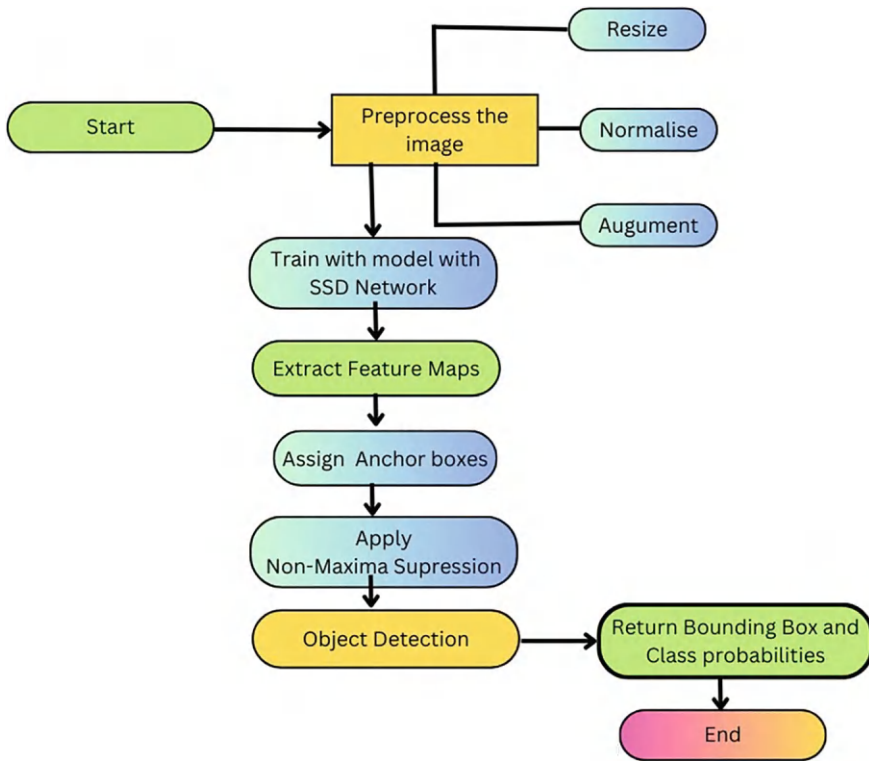


Fig. 1 Flow chart of SSD model for object detection

$$h_i = \text{baseheight} * \frac{\text{scale}}{\sqrt{\text{aspectratio}}} \quad (2)$$

where base height and base width are dimensions of anchor box, aspect ratio is aspect ratio of default boxes and scale is list of scale factors.

3.2 Bounding Box Prediction

SSD has 4 or 6 default boxes across each grid of the feature maps of different scales and aspect ratio. The bounding boxes are pre-selected boxes from the training set to match with ground-truth bounding boxes.

SSD passes a 3*3 convolution filters over feature maps and for each default box 4 bounding box coordinates ($\Delta(cx, cy, w, h)$), are predicted to match with the ground-truth bounding box. At the end the filter outputs confidence score for all object categories ($c_1, c_2, \dots, c_p$) along with the bounding box offsets.

3.3 Matching Strategy

Compare the default boxes with the ground-truth box and select default box which overlap most with the ground-truth box and has highest IoU value or $\text{IoU} > 0.5$.

3.4 Loss Function

SSD algorithm has two loss functions. They are:

3.4.1 Localisation Loss Function

The variation between ground-truth boxes and default bounding boxes is known as localisation loss. The loss is computed by only considering positive match predictions and ignoring negative matches.

The localisation loss between the predicted box l and the ground-truth box g is defined as the smooth L1 loss with cx, cy as the offset to the default bounding box d of width w and height h .

$$L_{\text{loc}}(x, l, g) = \sum_{i \in \text{Pos}}^N \sum_{m \in (cx, cy, w, h)} x_{ij}^k \text{smooth L1} \left(l_i^m - g_j^m \right)$$

x_{ij} takes value as 1 when $\text{IoU} > 0.5$ between default box i and ground-truth box j on class p and 0 otherwise.

3.4.2 Confidence Loss Function

It is the loss in making a class prediction. Only the class score of predictions for positive matches is used to compute the loss. Since the negative matches have no class score(0), they are disregarded. It is evaluated as the SoftMax loss with various classes confidences c (class score).

Finally, loss function is computed as:

$$L(x, c, l, g) = \frac{1}{N} (L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g))$$

where N is the number of positive match and α is the weight for the localisation loss.

3.5 *Hard Negative Mining*

SSD mines unmatched default boxes as negative matches. There are more negative matches than positive matches. The negative matches are background images and forward them to train the classifier making more robust. This creates an imbalance in the classifier as more negative matches are trained. So, to avoid this only those negative matches are trained which has high confidence loss. The ratio for each training batch is 3 negative match for 1 positive match.

3.6 *Data Augmentation*

By using various transformations, this method can expand the dataset and balance the classes. Furthermore, data augmentation enables the CNN model to capture all relevant patterns and characteristics by simulating a variety of scenarios and situations that it would experience in the actual world.

Data augmentation techniques play a crucial role in enhancing the performance, generalisation, and robustness of CNN models by providing a more varied and extensive training dataset. The important data augmentation strategies are:

- (i) Flipping.
- (ii) Cropping and expansion.
- (iii) Resizing with fixed size.
- (iv) Photo distortions.

Data augmentation can learn to recognise patterns and features in various orientations, positions, and scales also allows for the generation of additional training samples from existing data, making the training process more cost-effective.

Deep learning model construction and training is made possible via Keras, a high-level API that also offers the `tf.keras.preprocessing.image`. Keras offers a variety of image data augmentation techniques through its 'ImageDataGenerator' class.

'ImageDataGenerator' is used to define various augmentation parameters such as rotation, shifting, shearing, zooming, and flipping. The 'fit' method is then used to apply data augmentation to the training data. Finally, the model is trained using the augmented data by passing the augmented data generator to the 'fit' method.

3.7 *Inference Time*

SSD makes more predictions (8732) to find correct location, size and aspect ratios than any other detection method. Any predictions with class confidence score less than 0.01 is eliminated as they don't contain any objects. Then SSD proceeds to non-maximum suppression to remove duplicate predictions having same object. Predictions are sorted in ascending order by their confidence score. SSD looks for bounding

box predictions where the Intersection over Union (IoU) between those predictions and the acquired prediction for the same class is larger than 0.45. If found then the existing prediction will be ignored. At the end, only 200 predictions per image are considered.

4 Implementation

Following libraries have to be downloaded and installed via pip:

i. TensorFlow

It is a symbolic math library which is freely available. It processes the data, build model based on processed data and finally the model is trained. It takes input in multidimensional array format. Make use of graph framework as it performs numerical computations.

Types of TensorFlow

1. TensorFlow CPU
2. TensorFlow GPU

Command: `Pip install TensorFlow CPU.`

Steps to install TensorFlow GPU.

1. Download and install Visual Studio Professional 2015 or later version.
2. Download and install CUDA Toolkit, i.e. CUDA 9.0 from CUDA Toolkit Archive.
3. Download and install CUDA patches.
4. Download and install cuDNN.
5. Install TensorFlow-GPU via pip

Command: `Pip install TensorFlow GPU.`

ii. NumPy

NumPy is a freely available library and it is the basic library to perform any ML applications. It is used mainly to computations and calculations in math's as well as statistics. It makes complicated computations simple. It can work on large quantity of data. Multidimensional arrays and matrix multiplications can easily be performed on this.

Command: `Pip install NumPy.`

iii. SciPy

It is an advanced version of NumPy package. It contains many sub-packages which help to resolve mathematics and scientific issues. It contains data science features to perform analysis. It is simple to understand and user-friendly.

Command: `Pip install SciPy.`

iv. OpenCV

It is a cost-free library available on all operating systems. Used in computer vision, machine learning and image processing. It is used in reading and writing data from the images or videos. It can be utilised in all top programming languages. It can be used to detect objects by processing image, shapes can be detected, digits detection.

Command: Pip install OpenCV.

v. Pillow

It's the easiest library to use and comprehend. It serves to modify an image file's format. It works with every operating system and performs operations like saving and opening image files.

Command: Pip install Pillow.

vi. Matplotlib

It is used to create graphical user interfaces. It helps to view datasets in the form of graphs. Also called as plotting tool. It contains features such as zoom-out, zoom-in, drag and drop.

Command: Pip install Matplotlib.

vii. Keras

It is a cost-free library. It runs on topmost layer of several frameworks like TensorFlow, Microsoft Cognitive Toolkit. It is created to enable quick experiments with neural networks, it is user-friendly, extensible and modular.

Command: Pip install Keras.

viii. ImageAI

ImageAI allows API to identify thousand unique objects from a captured frame making use of pre-trained models which are trained on ImageNet-1000 dataset. Squeeze Net, ResNet, InceptionV3 and DenseNet are the provided model implementations.

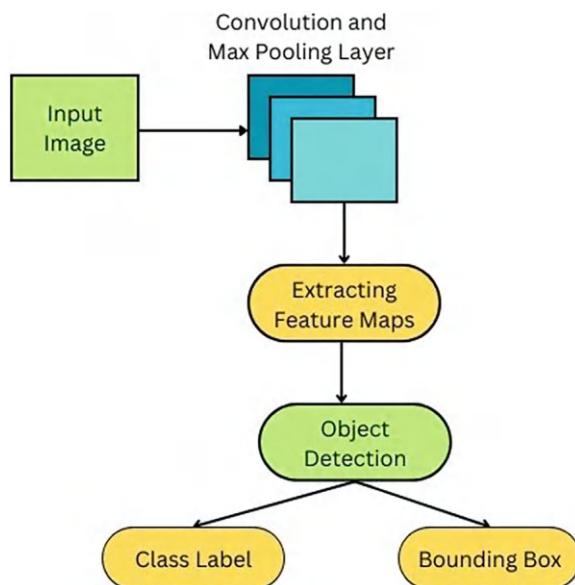
Command: Pip3 install ImageAI.

5 Results and Discussions

Object detection is implemented with Single Shot Multibox Detector (SSD) for real-time object detection. General framework of object detection is shown in Fig. 2.

- First load an image for object detection, deep learning models or algorithms generates huge set of bounding boxes passing over the complete image. This process is called image localisation.
- Visual features are taken out from the bounding boxes. These features are processed and examined to see what objects are shown in the bounding boxes based upon visual features. This process is called as image classification.
- Overlapping bounding boxes are joined to create a single bounding box in the terminal process.

Fig. 2 Training flow of CNN model



5.1 Functional Requirements

TensorFlow Object Detection API

It is a most popular and frequently used framework for generating a deep learning network that resolves object detection issues. Pre-trained models are present in the framework. These models are named as Model Zoo.

This Model Zoo comprised of group of pre-trained models which are trained various datasets like the COCO dataset, the KITTI dataset, and the Open Images Dataset.

5.2 Tensorflow Model Installation

- Step1: Browse to TensorFlow models webpage on GitHub website.
- Step2: Click on download zip file and extract files from it.
- Step3: Rename models-master folder to model.

5.3 Protobuf

Protobuf is used by the TensorFlow object detection API to arrange the training model parameters. Also used for communication purpose between developing programmes.

Step 1: Browse to Protobuf release GitHub.

Step 2: Download the latest version of zip file and extract it.

Step3: Open cmd, then open research directory of models folder.

Step4: To compile Protobuf type 'c:\Program files\protoc-3.4.0-win64\bin\protoc.exe' object_detection/protos/*.proto-python_out =

Step5: Go to object detection folder of research folder, open protos folder and py files will be generated by now.

5.4 COCO API

It contains huge amount of dataset which are used in computer vision applications. It has 330 k images and more than 200 k images are labelled. It contains 80 object categories, 91 stuff categories. Each image has 5 titles and 250,000 people with key points. Its dataset and metrics are used for accuracy evaluation, majorly in mAP.

Objects like person, skis, bicycle, snowboard, tie, backpack, sports ball, bird, baseball glove, carrot, surfboard, giraffe, tennis racket, stop sign, bottle, sheep, wine glass, cat, cup, dog, fork, bus, knife, train, spoon, elephant, bowl, orange, banana, suitcase, and the list continues (Fig. 3).

Placed three objects Cell phone, Motorcycle, and Tiger for detection. Here, camera captured and detected five images. It detected background objects too. Labelling of all detected objects are right except one which is Tiger but detected and labelled has Dog (Fig. 4).

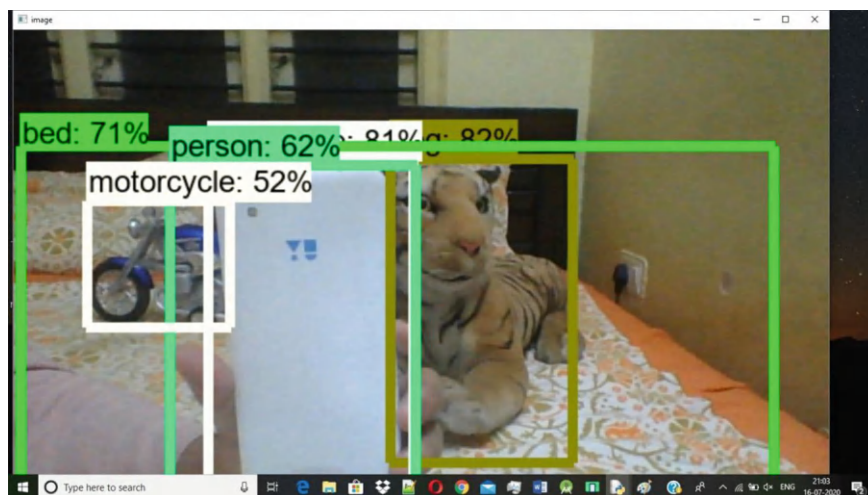


Fig. 3 Object detection with SSD

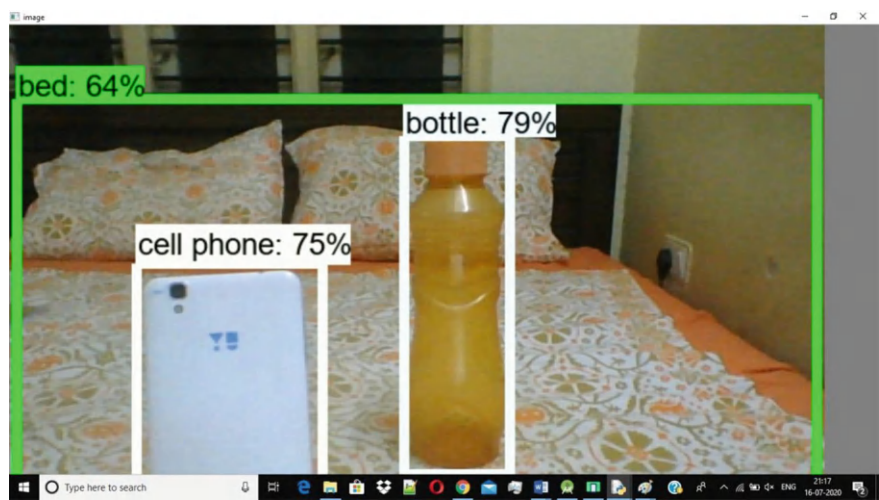


Fig. 4 Object detection

Here all the detected objects in an image is correct, there is no wrong labelling of the object (Fig. 5).

Detected images are not all correct. Actual object is Tiger but detected as Dog which is incorrect labelling (Fig. 6).

In this image, application couldn't come up with one solution. It provided two solutions, i.e. Elephant and Bear. As per the training dataset, the object looks like Elephant and Bear, but the actual object is Elephant. Application provided correct

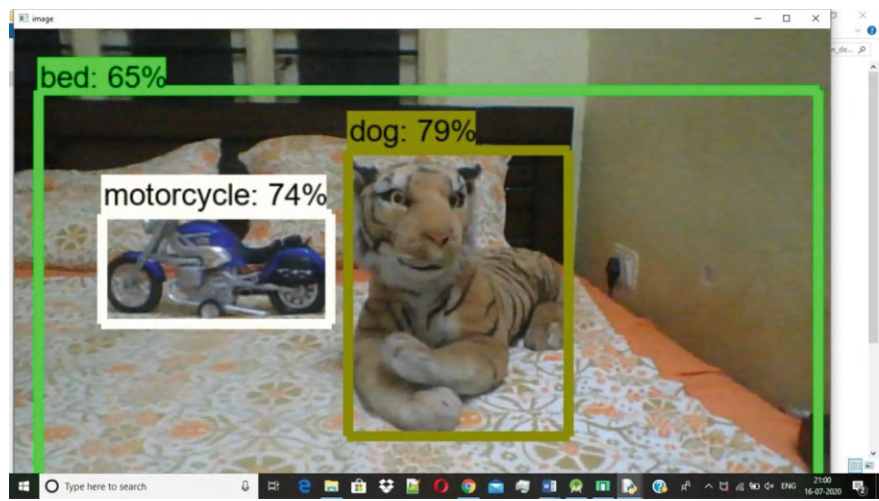


Fig. 5 Object detection with wrong labelling

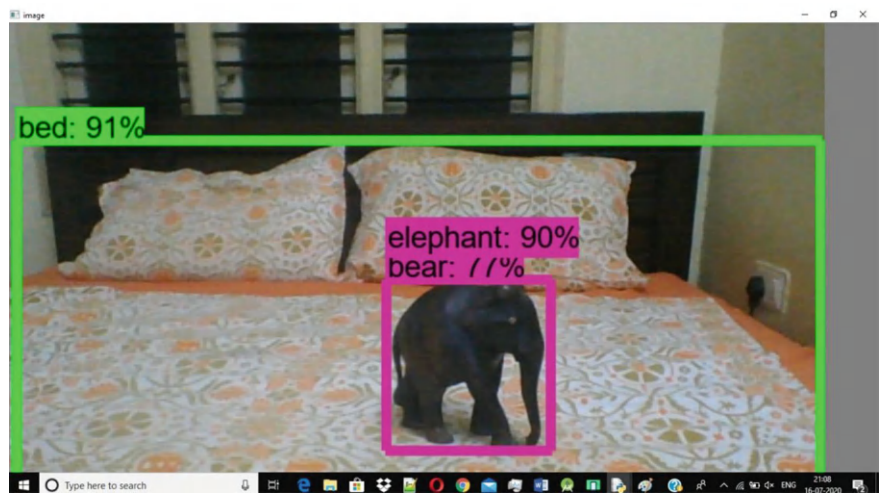


Fig. 6 Object detection with 2 solutions

solution along with the incorrect solution. The following graph shows the graph with accuracy with 97.6% obtained at 100 epochs (Fig. 7).

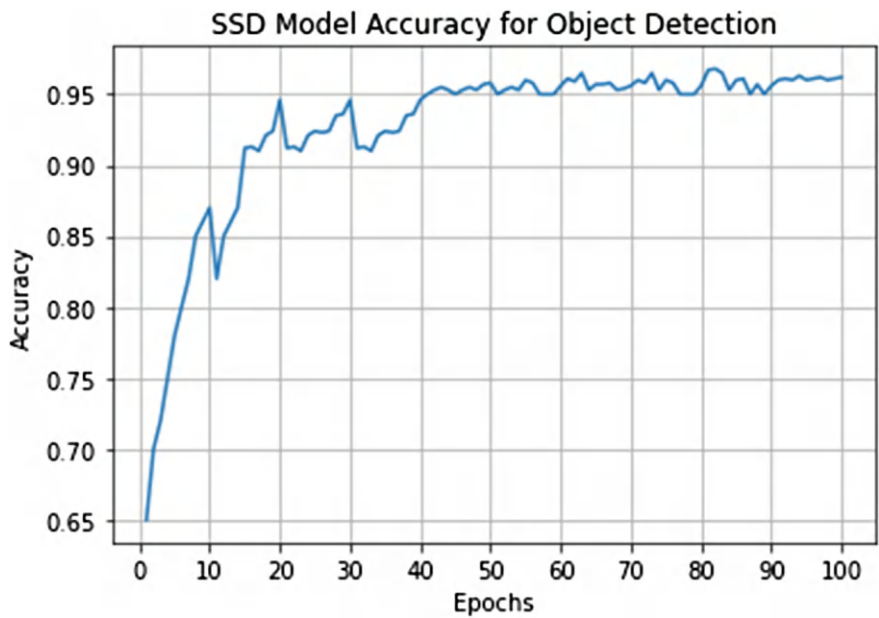


Fig. 7 Accuracy against epochs

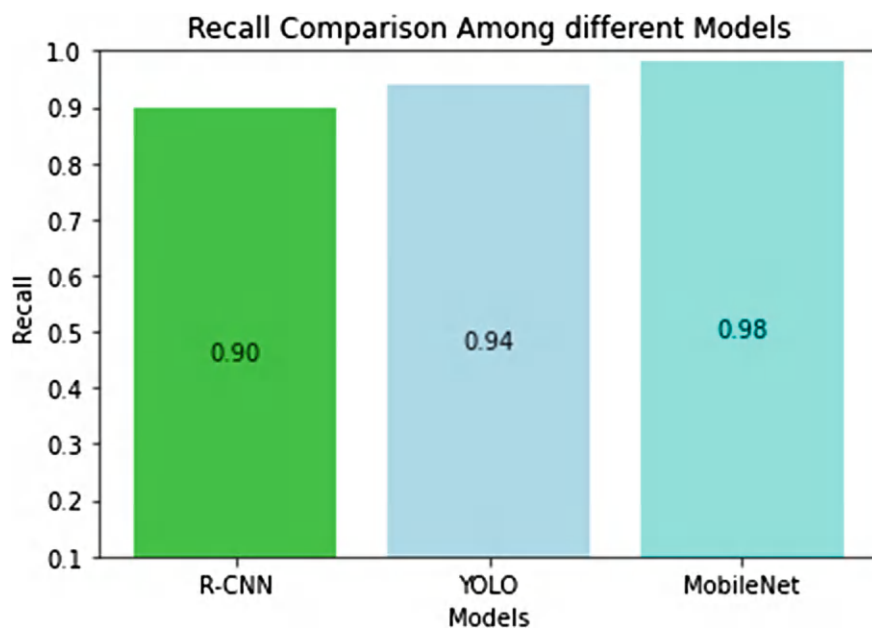


Fig. 8 Comparison of models

Different models performance is compared and analysed that SSD with MobileNet architecture has resulted with high recall with 98% as shown in Fig. 4 (Fig. 8).

6 Conclusion

In this paper, we proposed SSD algorithm to perform real-time object detection. SSD algorithm is integrated with MobileNet CNN architecture so there is reduction in number of parameters and less computational time is required. SSD makes use of small-scale convolution filter to predict classes and bounding boxes offsets. It lowers the input image resolution to 300*300 to increase the accuracy rate. SSD is more efficient and faster comparative to other object detection algorithms with an accuracy of 96.6% at 100 epochs. Detection performance can be improved with other CNN models such as Faster R-CNN and YOLO, EfficientNet.

Enhancement of the application can be done by object identification work. For visual tracking in the night, night vision mode should be available for better detections. There should be a movement in the camera when there is any object movement around it by tracking and recognising.

References

1. Zhou Y (2024) A yolo-nl object detector for real-time detection. *Expert Syst Appl* 238:122256
2. Manakitsa N, Maraslidis GS, Moysis L, Fragulis GF (2024) A review of machine learning and deep learning for object detection, semantic segmentation, and human action recognition in machine and robotic vision. *Technologies* 12:15. <https://doi.org/10.3390/technologies12020015>
3. Jena B, Nayak GK, Saxena S (2022) Convolutional neural network and its pretrained models for image classification and object detection: a survey. *Concurr Comput: Pract Exp* 34(6):e6767
4. Qiang B et al (2020) Convolutional neural networks-based object detection algorithm by jointing semantic segmentation for images. *Sensors* 20(18):5080
5. Shuai Q, Wu X (2020) Object detection system based on SSD algorithm. In: 2020 International conference on culture-oriented science & technology (ICCST). IEEE, 2020. Jonathan Hui, "SSD object detection: Single Shot MultiBox Detector for real-time processing"
6. Wadhwa K (2020) Accurate real-time object detection using SSD. *Int Res J Eng Technol (IRJET)* e-ISSN: 2395-0056 vol 07(05):2650-2654
7. Yadav SP, Jindal M, Rani P et al (2023) An improved deep learning-based optimal object detection system from images. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-023-16736-5>
8. Sabina N, Aneesa MP, Haseena PV (2022) Object detection using YOLO And Mobilenet SSD: a comparative study. *Int J Eng Res Technol (IJERT)* 11(06)
9. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu Alexander C, Berg C-Y (n.d.) SSD: single shot multibox detector
10. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (n.d.) MobileNets: efficient convolution neural networks for mobile vision applications
11. Younis A et al (2020) Real-time object detection using pre-trained deep learning models MobileNet-SSD. In: *Proceedings of 2020 the 6th international conference on computing and data engineering*
12. Feroz MdA et al (2022) Object detection and classification from a real-time video using SSD and YOLO models. *Computational intelligence in pattern recognition: proceedings of CIPR 2021*. Springer Singapore

Multidimensional Object Detection in Real Time Using Deep Learning Techniques



Piyush Kumar Pareek , S. A. Shreyanshu , S. Bharathi ,
and B. R. Shobha Rani

Abstract Multidimensional object recognition plays a very major role in the field of computer vision. The research work in the field of self-driving cars, robotics, video surveillance system requires object detection, localization and classification. Due to the remarkable advancement in the field of neural network, visual recognition system has achieved significant performance. In this work, RCNN and YOLO models are used to face the challenges during multidimensional object detection and classification in real time. The major goal of this work is to recognize the real-time objects using deep learning models to help visually impaired people to make them independent and sign language detection for vocally/hearing impaired people. This work will be extended to object detection in video also.

Keywords Multidimensional · Object detection · RCNN · Faster RCNN · YOLO

1 Introduction

In earlier days processing the images was limited. Now, the scenario is significantly changed with the advancement technologies. However, the latest image processing techniques are yet improve finding the solution to the typical problems in face recognition, numbers in vehicles, signs on roads, analysis of satellite images and medical images, etc. Object recognition is a very important part of image processing involve

S. A. Shreyanshu

Department of AI & ML, Nitte Meenakshi Institute of Technology, Bengaluru, India

P. K. Pareek

Department of MCA, Dr. Ambedkar Institute of Technology, Bengaluru, India

S. Bharathi (✉) · B. R. Shobha Rani

Dr. Ambedkar Institute of Technology, Bengaluru, India

e-mail: bharathi.mca@drait.edu.in

S. A. Shreyanshu

PES University, Bengaluru, India

the task like objects recognition in digital photos. Image classification contains activities like predicting the classes in the objects from given the image. Localization of objects is one of the important features which refer to identifying the region in an image and draw the box around the image. object detection will involve the two tasks such as object localization and classification of more objects in an image. Object recognition and categorization is a challenging task to differentiate between different related images in the field of computer vision.

The major task in the computer vision is segmenting the object, this is will be called as “object instance segmentation” or “semantic segmentation,” where identified objects highlight the specific pixels of the object instead of a rough bounding box. The object recognition is very difficult task in computer vision.

Human beings are capable of detecting and identifying objects available in the image. The visual system of human being is very fast and accurate in performing difficult tasks like recognizing multiple, multidimensional objects and detecting the barriers. The large data sets, faster GPU system and efficient algorithms are required to train the computers for multiple object recognition and classification with high accuracy.

YOLO algorithm will be used for object recognition, localization of objects, loss function for object identification and localization.

Classification of images includes assigning a tag to an image, whereas localization of objects draws a bounding box on all side's multiple objects in the image. Object recognition is really challenging work, which is a combination of two work such as draw a bounding box all the sides of each object of interest in the image and give them a class label.

Region-Based Convolutional Neural Networks, or RCNNs, is a technique used to solve the problems with object localization and object identification tasks. YOLO is another efficient technique used for object recognition in real time.

2 Related Work

Vaishnavi et al., presented the solution to the object tracking and object identification using deep learning technique. The single shot detector model is the fastest model for object tracking and object identification using single layer of a CNN model. The main aim of this work is to optimize the accuracy of single shot detector model [1]. The Mohammad Ali et al., all proposes wavelet compression technique, which is used to reduce the non-dynamic prediction errors. This technique is used to predict images on the sea surface [2]. The Meimetis et al., worked on challenges arises in tracking multiple objects and its resemblance with other objects and exorbitant density of the detected objects. The deep sort technique is incorporated with YOLO object detection method. The performance of the framework for multidimensional object detection analysis for real-time objects are analyses and shown. The combination of the different techniques gives the optimized result [3].

Jadhav et al., discussed the challenges and limitations in object detection with classical method of deep learning algorithms such as RCNN and YOLO [4]. Zhao et al., provide overview of the object recognition with deep learning has become a hot topic in research because of the efficient learning ability and superiority while working with scale transformation, obstruction and background switches. The paper shows the detailed information on deep learning-based object identification frameworks which handle different problems, such as low resolution obstruction, jumbled with changes on RCNN [5].

Pathaka et al., focus on deep learning technology has become more popular in current years due to the pioneering results during the classification of images, object recognition, natural language processing. The main idea behind the importance of using deep learning are because two reasons, viz. existence of huge datasets and more powerful Graphics Processing Units. To perform training, deep learning needs huge datasets and more dominant resources, the above requirements have already been satisfied in the recent years [6].

Galvez et al., focus on the accurate object detection and classification is possible with the help of deep learning techniques. Faster RCNN is one best technique to detect the object in real time with InceptionV2 [7]. Deng et al., worked on the challenges and limitations with object detection in computer vision. This complexity is reduced by using deep learning algorithms. But these algorithms explored with the following problems, decreases the dependency on data, to accomplish accurate detection of small objects, realization of multidimensional object recognition [8].

Devaki et al., uses open CV libraries to identify real-time objects captured by cameras. In this work different methods for object detection is discussed along with advantages and disadvantages [9]. Zaidi et al., discuss the recent object detection techniques using deep learning approach. This survey paper covers recent lightweight techniques for object classification and comparative analysis of different architecture [10]. YOLOv1 model is suggested by Tanvir Ahmad et al. for identifying an object from the image by altering loss function and including inception model and pooling layer of convolution layer of kernel of 1×1 . Then the results obtained are compared with RCNN model to identify the efficiency of the proposed model [11].

3 Objective

- Detecting 3D geometrical shapes
- Identifying overlapping images
- Identifying and classifying multidimensional and multiple objects from an image.

4 Methodology

The main objective of this work is to develop a framework for detecting the objects in the image. In the proposed system the real-time images are used as an input and applies different algorithms such as Hough transformers, deep learning algorithms to identify and recognize the object from the given data. Hough transform model will extract isolated part from the image of any shape, because it needs the required features to be in a specified form. This technique is mainly used to identify the curves like ellipse, circle, complex lines and the parameters if the characteristics is known. This technique can be applied on the image where analytical description of the attribute is not possible. Hough transformer is used to segment the images. This is one the popular feature extraction technique, initially it was used to identify the lines, later it is extended to detect the position of arbitrary shapes such as circles and ellipse. The most efficient and accurate deep learning models such as YOLO, RCNN, Faster RCNN are used.

RCNN is nothing but Region-Based CNN which is also called as region proposal model. Instead of working on large number of regions, RCNN proposes a bunch of boxes in the image and verifies if any of these boxes contains objects.

This model uses selective search techniques for object localization inside the image. This is a two-stage object detection model. The first stage identifies the subset of the objects in the image. This techniques combines the regions based the similar characteristics. These regions produces the region of interest. Then reshape all the region to match the input size of the CNN. The second stage classify the objects in the region of the image. SVM model will be used to classify the regions. Bounding box will be drawn to show the identified objects.

YOLO is one of the efficient neural network techniques, used for object detection in real time. The architecture of YOLO model consists of 24 convolution layer which is having two fully connected layers. The important feature of YOLO model is single-stage objection detection, this is developed for real-time object detection results in high accuracy. The entire image is processed in single shot, so it is very fast and efficient. Each and every object can be detected by drawing rectangular boxes and all objects are assigned to its class (Figs. 1, 2 and 3).

Pooling Layers

Pooling is a technique in CNN for generalizing feature extraction by convolutional filters and helping the neural network to recognize features independent of their location in the image. This layer decreases the size of the feature obtained from convolution layers, this in turn reduces the size of the parameter and computations over the network. This method gives the average of features present in the image. Table 1 and Figure shows the feature map of average pooling.

Fig. 1 Object detection and classification process

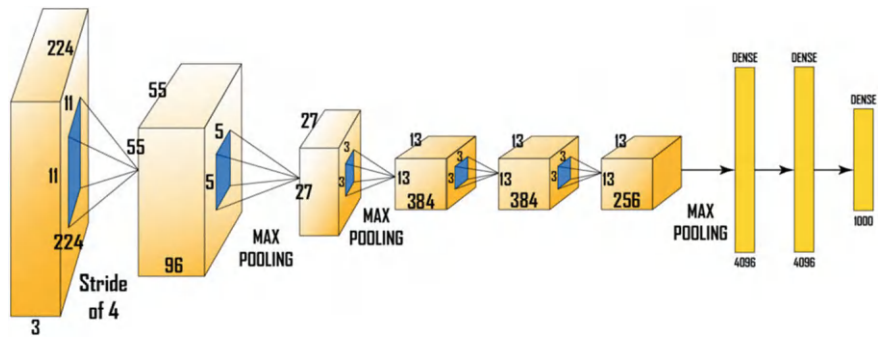
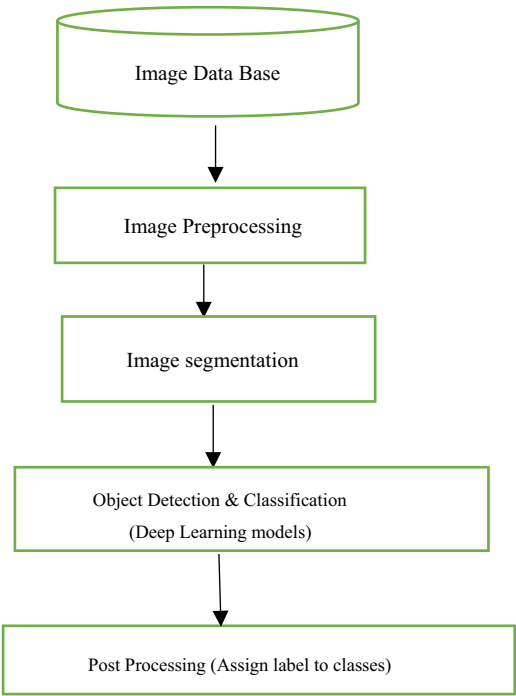


Fig. 2 CNN architecture

5 Algorithm

See Table 2.

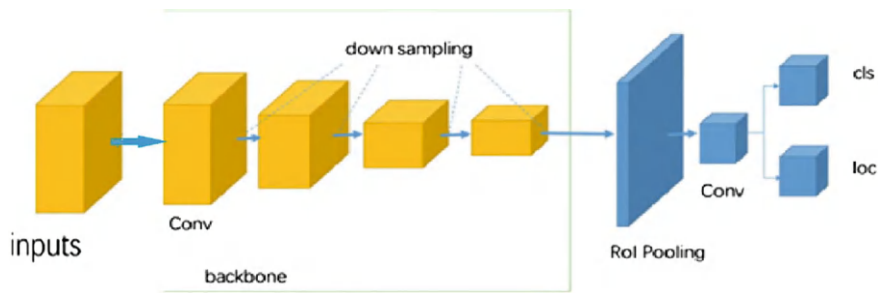
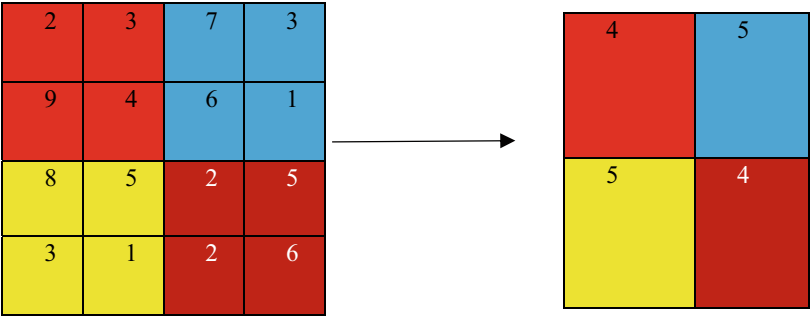


Fig. 3 YOLO architecture

Table 1 Average pooling feature map

Max pool
Filter-(2 *2)
Strides-(2,2)



6 Result and Performance Analysis

This research work is implemented using Python language in google COLAB. The image processing requires GPU server to process high dimensional images. The below specified image shows how an object recognition algorithm is implemented. All the objects in the image are discovered and detected with an unquestionable level of precision. This efficiency is achieved using the simplest algorithm called Convolutional Neural Networks(CNNs).

Each region will be considered as a different image and sends all the images to the CNN and categorized them into different classes. Once image is broken down into segments which are then assigned into its corresponding classes.

The main problem in using this technique is that the objects have inconsistent aspect ratios and spatial locations, in certain situations the object may occupy major part of the image, while in others, the object might cover a small portion of the image. The object shape could also be different. Due to which, large amount of computational

Table 2 Multidimensional object detection and classification

Input: Multidimensional Test data set Output: Model for Object detection & classification R-CNN 1. Image pre-processing 2. Hough transformers to identify the 3 D objects 3. Extract region proposals by applying selective search to extract region proposal from the input 4. Compute RCNN features 5. Classify the region proposal using SVM for all the classes validation accuracy and show the loss 6. Classify regions and assign the class for each region proposal. 7. Bounding box regressions are used, objects are captured clearly Yolo 1. Take input image 2. Divide the given image into grids 3. Apply IOU assign the thresh hold If box > thresh hold Draw the box Else discard 4. Repeat step 3 until all the boxes are considered as output or ignored 5. Apply pre trained model cv2.dnn.readNet("yolov3_training_last.weights", "yolov3_testing.cfg") on new test data 6. Predictions are plotted

time is required, region-based CNN will be used to solve this problem (Figs. 4, 5 and 6).

Region-Based Convolutional Neural Network

The RCNN model shows bundle of boxes, if any of the boxes hold any object when large number of regions are there. RCNN apply selective search to extract the features inside the boxes + (these boxes are called regions). Selective search is nothing but identifying the objects based on the features such as colors, textures, varying scales and enclosures. These regions give the region of interest (Figs. 7 and 8).

Consider a pre-trained CNN model, train the end of the layer of the network based on the total number of classes required to be identified. Identify the region of interest in all the image then change all the regions so that the images can meet CNN input size. After obtaining the regions Support Vector Machine (SVM) model is used to for object classification. Finally linear regression model used to create compact bounding boxes for the detected objects.

All the regions obtained through selective search methods are changed according to the input format required by CNN and all the regions are passed to the convNet.



Fig. 4 Object detection of image using CNN



Fig. 5 Original image

CNN extracts feature from all the regions and classifies the regions into various classes. A bound box will be drawn around the image to predict the detected region.

The main drawback is: training RCNN is very expensive. If 2000 features have to be identified based on selective search from an image using RCNN, for N images the number of CNN feature will be $N * 2000$. RCNN model extracts features, Linear



Fig. 6 Segment the image into object detection in various regions

Fig. 7 Stairs detection



Fig. 8 Stairs detection



SVM classifier for object detection and regressing model for compact bounding boxes (Fig. 9).

The combination of all these models makes RCNN very slow. It takes longer time for predicting the object in the image. When the image is overlapping, RCNN is not able to separate the overlapping feature. RCNN has better accuracy compare to other algorithms. Fast RCNN optimizes bounding box task and classification task but it

Fig. 9 Pothole detection

is slow because selective search method is used for region proposals. Faster RCNN predicts the objects efficiently in terms of aspect ratios and scales.

YOLO algorithm is used to identify and recognize different objects in the image (in real time). Object recognition in YOLO is done as a regression problem and gives the class probabilities of the detected objects. This model efficiently predicting the boundaries in the image and also predicts the false positive in the background. Comparatively this algorithm is much faster and more efficient in real-time object detection. In Fig. 7, total no. of coins in the image are 11 with yolo.

Staircase images in Figs. 7 and 8, pothole images in the Figs. 9 and 10, images in Figs. 11 and 12 have been taken from web cam to identify the different objects in real time. The main objective of this work is to identify the objects and obstacles on the road and notify about the same to the low vision and visually impaired people. In the next work all possible objects on the road like stones or any obstacles and moving objects like vehicles, animals and any human beings will be identified (Figs. 13 and 14).

Fig. 10 Pothole detection

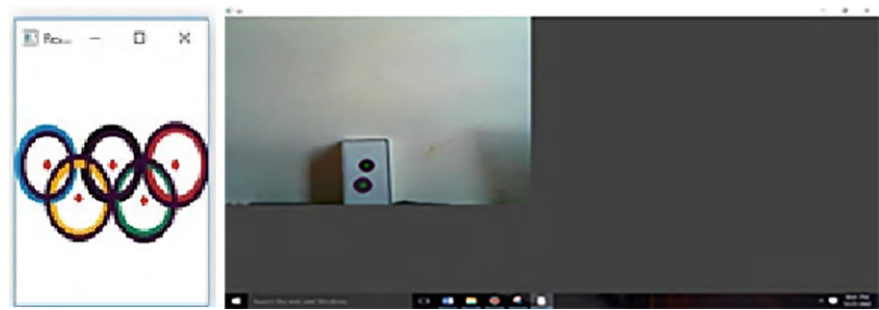


Fig. 11 Detection of overlapping Image from webcam object detection using captured geometric shapes

Fig. 12 Counting number of coins in overlapping images using yolo

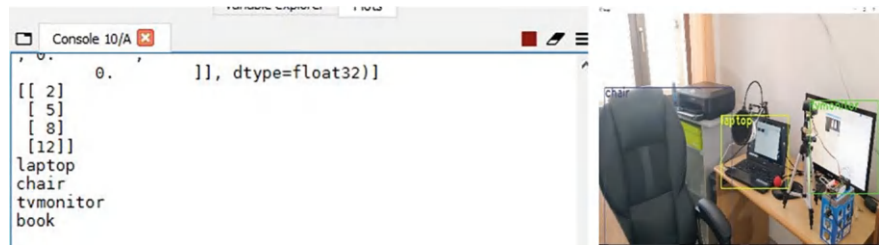
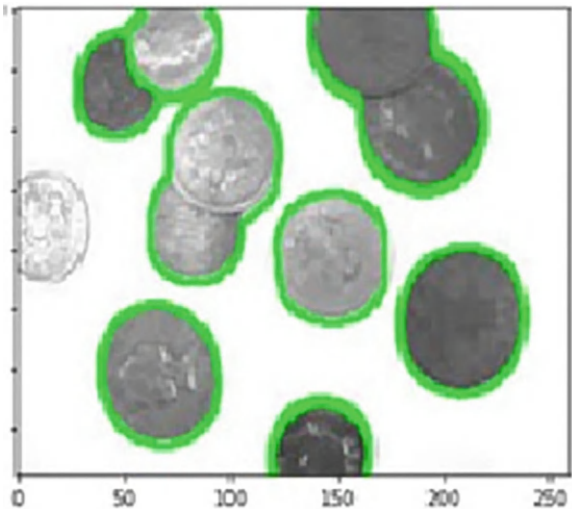


Fig. 13 Object detection and classification



Fig. 14 Object detection and classification

7 Conclusion

In this research work of multidimensional object detection and classification has been done using the RCNN, Fast RCNN and YOLO deep learning technologies. Initially object detection and classification was implemented on a couple of available datasets and by taking COCO model as training data set. Later same model is used for real-time data set. After implementing RCNN the result obtained will be very slow because it uses selective search algorithm for identifying regions. The RCNN model is not customized, it is fixed for object recognition problem. For feature extraction each region has to submit independently so in real-time object recognition especially when objects are overlapping it is not identifying the objects separately. To overcome the problem with RCNN, faster RCNN is used. This model uses a separate network to forecast the region proposal instead of selective search methods. The forecasted region proposals are later changed into RoI pooling layer, this is later used to categorize the image within the proposed region and predict the region for the bounding boxes.

YOLO is faster in magnitude compare to other object detection algorithms. This algorithm accesses the complete image and predicting the boundaries, even it predicts false positive in background. Compare to other algorithms yolo is more efficient and faster. The focus of the future work needs to be developing efficient models for multidimensional object detection in video for sign language detection. The model will be capable of applying for any dataset which is captured in real time. The proposed work is limited to small set of simple objects. These models may not

produces optimized result. There are much better models for complex images such as Masked RCNN, Faster RCNN, YOLOv7, gives optimized result in identifying the objects and object localization. The future work can be improved to identify objects, obstacles, moving objects and object localization in videos on the road.

References

1. Vaishnavi K, Pranay Reddy G, Balaram Reddy T, Srimannarayana Iyengar NC (2023) Real-time object detection using deep learning. *J Adv Math Comput Sci* 38(8):24–32. Article no. JAMCS.101284. <https://doi.org/10.9734/jamcs/2023/v38i81787>
2. Muhamed Ali A, Zhuang H, Ibrahim AK, Wang JL, Chérubin LM (2022) Deep learning prediction of two-dimensional ocean dynamics with wavelet-compressed data. *Front Artif Intell Sec Mach Learn Artif Intell* 5. <https://doi.org/10.3389/frai.2022.923932>
3. Meimetis D, Daramouskas I, Perikos I, Hatzilygeroudis I (2021) Real-time multiple object tracking using deep learning methods. *Neural Comput Appl* (Published online: 24 August 2021, Springer Nature). <https://doi.org/10.1007/s00521-021-06391-y>
4. Jadhav P, Koli V, Shinde P, Dr Pawar MM (2020) Object detection using deep learning. *Int Res J Eng Technol (IRJET)* 07(09) | Sep 2020 www.irjet.net (p-ISSN: 2395-0072, e-ISSN: 2395-0056)
5. Zhao ZQ, Zheng P, Xu ST, Wu X (2019) Object detection with deep learning: a review. *IEEE Trans Neural Netw Learn Syst* 30(11):3212–3232
6. Pathak AR, Pandey M, Rautaray S (2018) Application of deep learning for object detection. *Procedia Comput Sci* 132:1706–1717. www.sciencedirect.com
7. Galvez RL, Bandala AA, Dadios EP, Viceria RR, Maningo JM (2018) Detection using convolutional neural networks. In: *Proceedings of TENCON 2018—2018 IEEE region 10 conference* (Jeju, Korea, 28–31 October 2018)
8. Deng J, Xuan X, Wang W, Li Z, Yao H, Wang Z (2020) A review of research on object detection based on deep learning. *J Phys Conf Ser* 1684:012028 (IOP Publishing). <https://doi.org/10.1088/1742-6596/1684/1/012028>
9. Devaki P, Shivavarsha S, Bala Kowsalya G, Manjupavithraa M, Vima EA (2019) Real-time object detection using deep learning and open CV. *Int J Innov Technol Explor Eng (IJITEE)* 8(12S). ISSN: 2278-3075
10. Zaidi SS, Ansari MS, Aslam A, Kanwal N, Asghar M, Lee B (2022) A survey of modern deep learning based object detection models. *Digit Signal Process* 126:103514
11. Ahmad T, Ma Y, Yahya M, Ahmad B, Nazir S, Haq AU (2020) Object detection through YOLO neural network. Article in *Scientific Programming*. <https://doi.org/10.1155/2020/8403262>

Enhancing Campus Safety Monitoring: A Machine Learning Approach



Meenakshi, G. Bhavyashree, K. S. Prakruthi, and B. S. Keerthi

Abstract This research paper presents a data-driven approach to improving campus safety. We address the problem of enhancing security measures in educational institutions by leveraging machine learning techniques. The paper outlines the methodology, data collection, data preparation, and the application of machine learning models for predicting security incidents. Results include correlation matrices, accuracy scores, and recall values. The findings demonstrate the efficacy of our approach in creating a safer campus environment.

Keywords Campus safety · Real-time surveillance · Security systems · Predictive policing · Red time zones · Black spots

1 Introduction

Campus safety is a paramount concern in today's educational institutions. Despite existing security measures, there is a need for more proactive and data-driven approaches to reduce incidents. This research aims to leverage machine learning

Meenakshi · G. Bhavyashree (✉) · K. S. Prakruthi · B. S. Keerthi
Department of Artificial Intelligence and Data Science, Nitte Meenakshi Institute of Technology,
Yelahanka, Bangalore, Karnataka, India
e-mail: 1nt21ad014.bhavya@nmit.ac.in

An Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi,
Karnataka, India

Meenakshi
e-mail: meenakshi.rao.kateel@nmit.ac.in

K. S. Prakruthi
e-mail: 1nt21ad025.prakruthi@nmit.ac.in

B. S. Keerthi
e-mail: 1nt21ad015.keerthi@nmit.ac.in

techniques to predict and prevent security issues, fostering a safer and more inclusive campus environment.

The modern college or university campus is a dynamic and inclusive environment where students come to learn, grow, and exchange ideas. Traditional security measures, such as CCTV cameras and increased security personnel, are reactive in nature and are only triggered after an incident occurs. In recent years, there has been a growing interest in using predictive analytics to prevent crime and improve safety on college campuses. *Predictive policing* is the use of data and statistical methods to identify patterns and trends that can be used to predict future events, this a proactive approach that aims to prevent crime before it happens. In the context of campus safety, predictive analytics can be used to identify areas where crime is more likely to occur, as well as to identify individuals who may be at risk of victimization.

This article will explore the potential of using predictive policing to enhance women's safety on college and university campuses. Further we discuss the challenges and opportunities associated with implementing these models, also provided with the examples of how predictive analytics has been used to improve campus safety in the real world and will also raise awareness of the potential of predictive analytics to improve women's safety on college campuses, and to encourage institutions to explore the use of this technology.

2 Literature Survey

From a women's security standpoint, a lot of technologies are already in use. Every gadget has unique features, benefits, and disadvantages. India and other nations are becoming increasingly concerned about women's safety. The current surveillance systems are inadequate for keeping an eye on remote locations, but they are enough for keeping an eye on crowded areas. The crimes are more likely to happen in less populated areas, which makes surveillance difficult and ultimately raises the crime rate.

Smartphone app for women's safety called WoSApp, introduces this new concept of WoSApp, which is a smartphone app designed to provide women with a discreet way to call law authorities and share their precise location in an emergency [1]. WoSApp is a global software designed to protect women and provide prompt assistance in an emergency. Proposed App will include some more features like incident monitoring and will be more effective to reduce the number of case. According to a study on *UAV-based security system for prevention of harassment against woman*, drones can be used to detect motions, especially at night, and minimize public harassment [2]. The study also focuses on cost-effective solutions. This enhanced data-driven strategy and community involvement set apart from the tech-based surveillance and reaction of the UAV system. One drawback could be the potential intrusion of privacy when using drones for surveillance. Mobile passwords or credit card pins might be captured by the drones.

Several research publications highlight the usage of IoT and wearables to improve women's safety. These devices include automatic hazard detection, alert calls, and live tracking. On the other hand, disadvantages include malfunctions, misleading alerts, possible abuse of self-defence measures, and restrictions on battery life and user comfort [3–6]. While these solutions have their own special benefits and difficulties, they seek to meet women's safety concerns. With the rise in online purchases, it is imperative to *detect credit card fraud*. To attain high accuracy, a variety of algorithms have been used, including Random Forest algorithm, Adaboost algorithm, supervised methods, and Deep Learning techniques like OSCNN [7–10]. With the number of credit card fraud instances increasing, using AI and machine learning—more especially, Support Vector Machine—proves useful in automating fraud detection and increasing accuracy in identifying fraudulent transactions.

The process of learning from a training dataset and applying that knowledge to unknown data is highlighted in the article *machine learning*, which offers a succinct and educational introduction to the principles of machine learning [11]. The incorporation of a case study on code analysis lends a practical dimension to the conversation, rendering it an invaluable tool for anybody seeking to comprehend the fundamental principles of machine learning. The significant influence of Deep Learning in industries like healthcare and driverless cars is highlighted in the *Review of Deep Learning Algorithms and Architecture* [12]. It tackles the drawbacks of conventional approaches and investigates deep architectures, mathematical underpinnings, and optimization strategies, making it an invaluable tool for professionals in the industry.

Deep Learning is another possible approach to develop safety related predictive models. Article *Introduction to Artificial Neural Networks*, which take inspiration from the human brain, explains their capabilities and structure in detail. It highlights the special chances for study and practical uses of neural networks, ranging from sophisticated system modelling to language recognition [13]. It is an interesting and educational resource for anybody interested in learning more about neural networks, complete with real-world examples. A study on *Women's Safety System Using IoT* is used to solve the important issue of women's safety issues [14]. The suggested concept offers a multifaceted approach, incorporating GPS and GSM for notifying contacts and authorities during crises. It is equipped with an Arduino microcontroller and a variety of sensors. The integration of a Piezoelectric buzzer with a push-button system provides an additional degree of safety, rendering it a viable approach to augment the protection of women. One drawback is that the sensors may malfunction at crucial moments, and in some isolated locations, the internet link may be weak or non-existent.

The article *Women's Safety with a Smart Foot Device* addresses the urgent problem of insecurity and threats experienced by women and offers a smart foot device as a solution for women's protection. A discrete wearable device with capabilities like a Microcontroller board, GPS, GSM, and a buzzer is made possible by the integration of IoT technologies [15]. It increases the protection of women in public areas and offers a dependable, affordable way to stop harassment and attacks when triggered by a particular foot movement. Given that it's a foot gadget, there's a potential that the sensors could malfunction, and that data and signal will be lost at crucial moments.

The *Self Defence and Safety Monitoring System* goal is to develop a smart safety device that is both user-friendly and effective, capable of responding to dangers on its own and providing users with a proactive response [16]. The paper significantly improves personal safety and reduces the need for user participation in emergency scenarios by examining current models and suggesting novel features. However, this item has a flaw in that because it relies less on human assistance, there's a chance it could cause false alarms.

The article *Sentimental Analysis of Amazon Customers using Deep Learning Techniques* explores the realm of online shopping and customer sentiment on Amazon [17]. By utilizing Deep Learning techniques, it examines a large dataset of reviews of exquisite food and customer feedback. The article improves data labelling accuracy and offers insightful information about changing client preferences and product quality evaluation in the digital marketplace by utilizing both manual and active learning techniques. The article *Economic and Financial Data Analysis System Based on Deep Learning and Neural Network Algorithm* explores how neural networks and Deep Learning can be used to analyse data from the financial markets [18]. Utilizing state-of-the-art methods such as CNN, LSTM, and RNN, the study shows remarkable stock price prediction accuracy and presents a viable way to improve financial and economic analysis. This approach represents a major leap in this subject and has the potential to completely change our understanding and prediction of prices, trends, and dangers in the financial markets.

In examining agricultural crop yield prediction methodologies over the last two decades in India, existing models like the 'Crop Advisor' software [19] often overlook crucial parameters, hindering their precision and practical applicability. Similarly, temperature forecasting models [20] employing machine learning algorithms demonstrate varied accuracy, with certain methods exhibiting greater precision. However, these models overlook the comprehensive analysis of dynamic weather conditions, limiting their forecast accuracy. Additionally, weather prediction models [21], while diverse and encompassing, encounter challenges in accurately predicting diverse weather parameters due to their complex and evolving nature. Integrating insights from these studies could inform a predictive framework for identifying vulnerable zones for women in college campuses by considering weather-influenced risk factors. The paper *Safety leading indicators for construction sites: A machine learning approach* explores how machine learning methods can develop safety leading indicators for construction site risk assessment. Despite limitations in validated leading indicators, the study employs machine learning models trained on safety inspection and project data [22]. Drawbacks include limited feature selection techniques and a focus on construction specific variables. However, insights into predictive models could aid in predicting dangerous zones for women in college by integrating similar risk assessment methodologies.

The work *A Data-Driven Approach for Driving Safety Risk Prediction Using Driver Behaviour and Roadway Information Data* presents a novel method for real-time traffic safety risk prediction customized for drivers [23]. While leveraging naturalistic driving study data, limitations include a narrow focus on specific driver variables and reliance on logistic regression. However, insights into driver-specific

risk prediction methodologies could inform the development of systems predicting hazardous zones for women in college campuses, integrating driver behaviour with location-based risk assessments. The *work Model-Based Versus Data-Driven Approach for Road Safety Analysis: Do More Data Help?* Scrutinizes the efficacy of parametric versus nonparametric crash risk prediction models with varying data sizes [24]. Focused on negative binomial models and kernel regression, it highlights the latter's superior predictive performance with larger crash datasets. However, it overlooks nuanced factors impacting model reliability. Insights from this comparative analysis can aid in developing nuanced safety prediction models, potentially enhancing the identification of unsafe zones for women in college environments by merging data-driven methodologies with location-based risk assessments.

Article titled *Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention* explores ML and computer vision's potential in crime detection and prevention, emphasizing the transformation in law enforcement post-implementation [25]. Conversely, *Crime Prediction Model using Deep Neural Networks* focuses on neural networks for crime prediction, addressing imbalanced data issues, yet lacks comprehensive exploration of data augmentation's impact on results [26]. Insights from these papers could inform predictive models for identifying hazardous zones for women in college by integrating ML techniques with location-based risk assessments, emphasizing robust data augmentation for improved accuracy.

3 Methodology

Ensuring the safety of all members of the college community, particularly women, is paramount to fostering a welcoming and supportive learning environment. However, incidents of violence and harassment continue to occur on college campuses across the nation. To address these challenges effectively, we propose a comprehensive methodology that combines data-driven analysis with participatory action research (PAR) to develop and implement evidence-based strategies for improving women's safety.

The methodology described is designed to produce a comprehensive and rigorous understanding of the challenges of women's safety on college campuses. The findings of the study will be disseminated to campus administrators, security personnel, and students through publications, presentations, and workshops. The findings will be used to develop and implement evidence-based strategies for improving women's safety on college campuses. Our methodology involves data collection, predictive analysis, real-time surveillance, community involvement by the application of ML models.

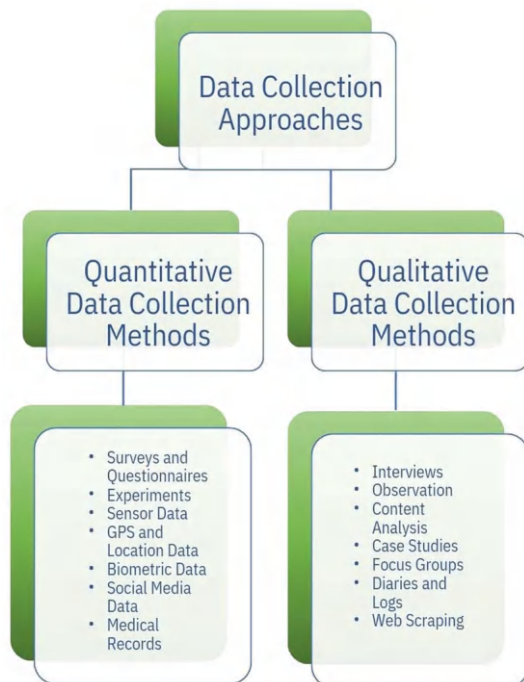
- **Data Collection:** Anonymous and non-anonymous data collection mechanisms are employed to gather incident reports and related information.

- **Predictive Analytics:** Advanced data analytics techniques are applied to predict potential mishaps based on historical data and patterns.
- **Real-time Monitoring:** Dashboards and charts are developed to monitor campus safety incidents in real time.
- **Community Involvement:** The entire student community is engaged in the prevention of mishaps through reporting and intervention.

4 Data Collection and Preparation

Data collection and preparation are essential steps in the field of machine learning. There are 2 types of data collection which can be used to train and evaluate the machine learning models as shown in Fig. 1. *Quantitative Data Collection* is a research method that involves gathering numerical information or data, typically through structured surveys, experiments, or observations, to analyse and draw statistically significant conclusions about a specific phenomenon or topic. *Qualitative Data Collection* is a research approach focused on gathering non-numerical, descriptive information, often through open-ended interviews, observations, or content analysis. It aims to understand complex behaviours, opinions, and experiences in a nuanced and contextual manner.

Fig. 1 Flowchart of various data collection methods



Some methods can be used for both quantitative and qualitative data collection, depending on how they are implemented. For example, interviews can be structured to collect quantitative data or unstructured for qualitative data. Similarly, synthetic data can be used to generate both qualitative and quantitative data, depending on the characteristics and structure of the data it's meant to mimic. It can be designed to resemble either type of data or a combination of both. Synthetic data is not a distinct data collection method but rather a technique for generating artificial data that simulates real-world data. Synthetic data is created to mimic the statistical properties, structure, and patterns of real data without containing actual information about individuals, entities, or events.

The dataset, which forms the basis for this analysis and model development, is a crucial part of this study. It consists of ten key attributes, each one intended to represent a specific aspect of security incidents in a learning environment. These descriptions are given in Table 1. *Incident Type*, which groups incidents into typical categories like theft, vandalism, assault, and harassment; *Location*, which records the location of each incident, including places like the library, dormitory, and parking lot; and *Time of Day* which designates whether incidents happened in the morning, afternoon, or evening. The dataset also includes a *Date* column containing randomly generated dates within a 365-day range. Number of witnesses, security camera, reported-by, response time, weather condition, incident severity, and incident resolution are the additional attributes.

To assure a thorough and varied representation of potential security incidents, this synthetic dataset was created. An element of unpredictability is added by purposefully introducing randomness, which mimics the complexity of data on security incidents that occur in the real world. The properties, content, and structure of the dataset serve as the foundation for the research's following data preprocessing and machine learning activities. Data preparation involves cleaning and transforming raw data into a format that is suitable for analysis and feature engineering to create a suitable dataset for machine learning. This is especially important for research on sensitive topics such as women's safety, where participants may be hesitant to share their experiences.

The data collection methods will produce a variety of data types, including quantitative data from surveys and polls, and qualitative data from interviews, focus groups, and case studies. Each of these data types will require different data preparation steps. Quantitative data from surveys and polls will need to be cleaned and coded before it can be analysed. This may involve removing incomplete or invalid responses, and creating new variables to represent key concepts. For example, the survey may ask students to rate their feeling of safety on campus on a scale of 1–5. This data could be coded into a new variable called *safety rating* with the following values: 1 = Very unsafe, 2 = Somewhat unsafe, 3 = Neutral, 4 = Somewhat safe, and 5 = Very safe.

Qualitative data from interviews, focus groups, and case studies will need to be transcribed and analysed. This may involve identifying key themes and patterns, and developing codes to represent these themes and patterns. For example, the interviews may reveal that students feel unsafe walking alone at night on campus. This theme could be coded as *unsafewalkingalone*. To ensure the data's compatibility

Table 1 Attributes of the model and their description

Attributes	Description
Incident_Resolved	The issues have been addressed and the situation has returned to normal
Incident_Type_Harassment	Unwanted behaviour that is intimidating, hostile or discriminatory
Incident_Type_Theft	The unauthorized taking of others property
Incident_Type_Vandalism	The wilful or malicious destruction or damage of property that belongs to person
Location_Classroom	Any form of incident that occurred in classroom
Location_Dormitory	Any form of incident that occurred in dormitory or hostel inside the educational sector
Location_Library	Any form of incident that occurred inside a library or the surrounding of library
Location_Parking_Lot	The wilful or malicious destruction or damage of property that belongs to person
Location_Washroom	Any form of occurrence that has occurred in and around the washroom area
Time_of_Day_Evening	Incident occurred during the evening period of day
Time_of_Day_Morning	Incident occurred during the morning period of day
Number_of_Witnesses_2	Two individuals witness the incident occur at a place
Number_of_Witnesses_3	Three individuals witness the incident occur at a place
Number_of_Witnesses_4	Four individuals witness the incident occur at a place
Security_Camera_Yes	A video surveillance device actively monitoring
Reported_by_Staff	The incident was brought to attention by a member of staff
Reported_by_Student	The incident was brought to attention by a student
Response_Time_Prompt	The nature of the incident demands an immediate and appropriate action to ensure safety and resolve the issues

with machine learning models, several critical preprocessing steps were executed. Notably, categorical attributes such as *IncidentType*, *Location*, and *Reported-by* underwent label encoding. This transformation was imperative as machine learning algorithms typically require numerical input. Label encoding assigned unique integer values to each category within these attributes, rendering them machine-readable without introducing ordinal relationships.

Furthermore, the *date* attribute was processed to extract valuable temporal information. Specifically, it was decomposed into three new features: *Year*, *Month*, and *Day*. This decomposition facilitates a more granular exploration of temporal patterns related to security incidents. These preprocessing measures enable the dataset to be used effectively for building and training machine learning models. The resultant dataset maintains its original dimensions while providing the necessary structure and format for the subsequent predictive modelling. These preprocessing steps are not only crucial for model compatibility but also essential for uncovering patterns

and relationships hidden within the data. By converting categorical attributes into numerical representations and extracting temporal components, the dataset becomes a valuable resource for training a deep neural network model designed to predict campus security incident resolution.

5 Deep Learning Prediction Model

We implemented a Deep Learning model in a neural network architecture to analyse and predict campus security incident resolutions. Details of the model, features, and training process are provided in this section.

The model consists of three layers:

- an initial input layer with 64 units
- a hidden layer with 32 units
- a final output layer with a binary unit.

The choice of the Rectified Linear Unit (ReLU) activation function in both the input and hidden layers is motivated by its ability to capture complex, non-linear relationships within the data effectively. For the output layer, a sigmoid activation function was chosen, as it is well-suited for binary classification tasks, such as determining whether a security incident was resolved. It is represented as: Rectified Linear Unit (ReLU) Activation Function $f(x) = \max(0, x)$.

Sigmoid activation function

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

The model's parameters are optimized through the *adam* optimizer and a binary cross-entropy loss function, which measures the dissimilarity between predicted and actual outcomes. The dataset is divided into training and testing sets, with 80% allocated for training and 20% for testing to ensure robust model validation. Training the model occurs over ten epochs, with a batch size of 32, allowing the neural network to adapt to the patterns present in the data. It is represented as: Binary Cross-Entropy Loss Function:

$$L(y, \hat{y}) = -y \cdot \log(\hat{y}) - (1 - y) \cdot \log(1 - \hat{y}) \quad (2)$$

This Deep Learning methodology encapsulates the steps taken to develop a predictive model capable of discerning security incident resolutions based on the provided attributes. The approach seeks to extract valuable insights from the data, culminating in the model's ability to make informed predictions regarding incident outcomes. The next stage of this research involves evaluating the model's performance and interpreting the results.

6 Results

The final output includes the results of training a neural network model on the dataset and evaluating its performance on a test dataset. Let’s break down the key components of the output:

1. **Training the Model:** The neural network model is trained for 50 epochs. During training, the model iteratively learned from the training data to make predictions. For each epoch, it shows details such as the loss, accuracy, and validation loss and accuracy. The loss represents how well the model’s predictions match the true labels, and accuracy is the proportion of correct predictions.
2. **Confusion Matrix:** The confusion matrix is a table that summarizes the model’s predictions against the actual classes in the test dataset. The model was assessed on a test dataset following training, and the confusion matrix offers an analysis of the predictions made by the model.

The count of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) is displayed in the confusion matrix, a 2×2 matrix as given in Table 2.

- The first row corresponds to the “Incident Resolved” class 0 (e.g., “False”).
- The second row corresponds to the “Incident Resolved” class 1 (e.g., “True”).
- The first column represents the model’s predictions of class 0, and the second column represents predictions of class 1.

This can be interpreted as:

- True Positives (TP): 90 instances were correctly predicted as class 1.
 - True Negatives (TN): 24 instances were correctly predicted as class 0.
 - False Positives (FP): 54 instances were incorrectly predicted as class 1.
 - False Negatives (FN): 32 instances were incorrectly predicted as class 0.
3. **Classification Report:** The classification report provides more detailed information about how well the model performs for each class, highlighting the trade-off between precision and recall for each class. It provides a summary of various classification metrics for both classes (0 and 1) or each class, it shows precision, recall, and the *F1*-score.

Precision is the true positive to total positive prediction ratio is known as precision. It is represented in following way:

Table 2 Confusion matrix

Class	Predicted class 0	Predicted class 1
Actual class 0	24	54
Actual class1	32	90

$$P = \frac{TP}{TP + FP}$$

(3)

Recall is the ratio of true positives to the overall number of positive occurrences. It is represented in following way:

$$R = \frac{TP}{TP + FN}$$

(4)

F1-score is a balanced indicator of a model’s performance that is calculated as the harmonic mean of precision and recall. The quantity of samples in each class is known as support. It is represented in following way:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

(5)

Support is the number of actual occurrences of each class in the test dataset. Here’s a breakdown of the classification report given in Table 3

- For class 0 (Incident Resolved = 0), the model has a precision of 0.43, recall of 0.31, and an *F1*-score of 0.36.
- For class 1 (Incident Resolved = 1), the model has a precision of 0.62, recall of 0.74, and an *F1*-score of 0.68.
- The “macro avg” and “weighted avg” rows provide the average values for precision, recall, and *F1*-score, considering both classes.

Table 4 gives the information about resolved incidents and types of incidents. Table 5 gives the statistics in detail. A correlation matrix is essential for quantifying and understanding linear relationships between variables in a dataset, offering insights into associations and patterns. The matrix serves as a foundational tool for advanced statistical analyses, aiding decision-making, and providing a comprehensive overview of multivariate dataset dynamics. Figure 2 shows the resultant corresponding correlation matrix.

Table 3 Classification report

	Precision	Recall	F1-score	Support
Class 0	0.43	0.31	0.36	78
Class 1	0.62	0.74	0.68	122
Macro avg	0.53	0.52	0.52	200
Weighted avg	0.55	0.57	0.55	200

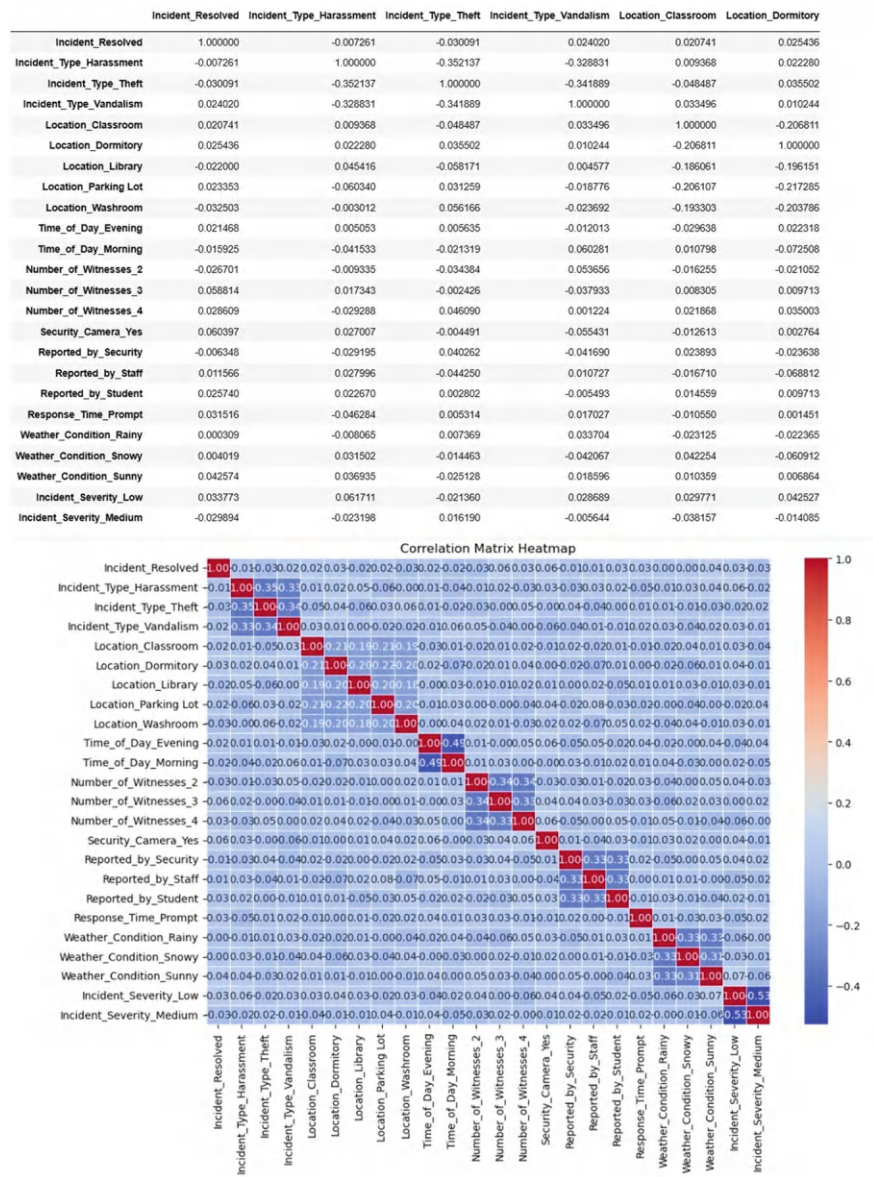


Fig. 2 Correlation matrix to analyse feature relationships, accuracy scores, recall value

Table 4 Overview of resolved incidents and their types

	Incident resolved	Incident type harassment	Incident type vandalism	Incident type theft	Location_ classroom	Location_ dormitory
0	True	0	0	1	0	0
1	True	0	1	0	1	0
2	False	1	0	0	0	1
3	True	0	0	0	0	0
4	False	0	1	0	0	0

Table 5 Descriptive statistics of incidents

	Incident type harassment	Incident type theft	Incident type vandalism	Location classroom	Location dormitory	Location library
Count	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000
Mean	0.253000	0.268000	0.242000	0.164000	0.179000	0.150000
Std	0.434948	0.443139	0.428509	0.370461	0.382544	0.357253
Min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000
Max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

7 Conclusion and Future Work

In conclusion, Women’s safety is a paramount concern, and our app is a vital platform dedicated to safeguarding college women and helping them overcome traumatic incidents and issues. This research demonstrates the potential of machine learning in enhancing campus safety. Our approach addresses the identified problem in existing security solutions, providing a proactive and data table-driven solution. The results show a significant improvement in security measures, contributing to a safer and more inclusive campus environment. The core strength of our app lies in its robust system of monitoring and analysis, where data collected from numerous women is utilized effectively. This data-driven approach is instrumental in identifying areas of concern, referred to as *black spots* and *red zones*, which may pose safety risks. By offering real-time insights and alerts, the monitoring and analysis aspect of our app plays a pivotal role in empowering women with information about the safety of their surroundings, thus enabling them to make informed decisions and stay safe. Some of the Methods and Strategies for accessing the anonymous and non-anonymous data collection in the future are listed here.

- The anonymous survey ought to be distributed to all students on campus. The survey will ask students about their experiences with violence and harassment, their perceptions of safety on campus, and their suggestions for improving campus safety.
- The online poll can be expected to posted on the university website and social media pages to ask students about their experiences with violence and harassment, their perceptions of safety on campus, and their suggestions for improving campus safety.
- The drop boxes can be placed in strategic locations around campus, such as student centres, libraries, and dining halls. Students will be able to anonymously submit their experiences with violence and harassment, as well as their suggestions for improving campus safety.
- The interviews and focus groups can be conducted with students who have experienced violence or harassment on campus. The interviews and focus groups will explore the students' experiences in more detail and gather their insights on how to improve campus safety.
- The case studies intend to examine specific incidents of violence or harassment on campus. The case studies will identify the factors that contributed to the incidents and develop recommendations for preventing similar incidents from happening in the future.

References

1. Chand D et al (2015) A mobile application for women's safety: WoSApp. In: TENCON 2015—2015 IEEE region 10 conference, pp 1–5. <https://doi.org/10.1109/TENCON.2015.7373171>
2. Mohan R et al (2017) UAV based security system for prevention of harassment against woman. In: 2017 International conference on intelligent computing, instrumentation and control technologies (ICICICT), pp 874–879. <https://doi.org/10.1109/ICICICT1.2017.8342680>
3. Monisha DG, Monisha M, Pavithra G, Subhashini R (2016) Women safety device and application-FEMME. In: Indian journal of science and technology, vol 9. <https://doi.org/10.17485/ijst/2016/v9i10/88898>
4. Hyndavi V, Nikhita NS, Rakesh S (2020) Smart wearable device for women safety using IoT. In: 2020 5th International conference on communication and electronics systems (ICCES), pp 459–463. <https://doi.org/10.1109/ICCES48766.2020.9138047>
5. Das S et al (2021) Women security system. In: Int J Eng Res Technol 10(7):247–251
6. Shah CM, Sangoi VB, Visharia RM (2014) Smart security solutions based on Internet of Things (IoT). In: Int J Curr Eng Technol 4(5)
7. Sailusha R et al (2020) Credit card fraud detection using machine learning. In: 2020 4th International conference on intelligent computing and control systems (ICICCS), pp 1264–1270. <https://doi.org/10.1109/ICICCS48265.2020.9121114>
8. Abd El Naby A, Hemdan EE, El-Sayed A (2021) Deep learning approach for credit card fraud detection. In: 2021 International conference on electronic engineering (ICEEM), pp 1–5. <https://doi.org/10.1109/ICEEM52022.2021.9480639>
9. Lopes AP et al (2021) Comparative analysis of deep learning techniques for credit card fraud detection. In: 2021 International conference on advances in computing, communication, and control (ICAC3), pp 1–5. <https://doi.org/10.1109/ICAC353642.2021.9697205>

10. Alkhatib KI et al (2021) Credit card fraud detection based on deep neural network approach. In: 2021 12th International conference on information and communication systems (ICICS), pp 153–156. <https://doi.org/10.1109/ICICS52457.2021.9464555>
11. Louridas P, Ebert C (2016) Machine learning. *IEEE Softw* 33(5):110–115. <https://doi.org/10.1109/MS.2016.114>
12. Shrestha A, Mahmood A (2019) Review of deep learning algorithms and architectures. *IEEE Access* 7:53040–53065. <https://doi.org/10.1109/ACCESS.2019.2912200>
13. Uhrig RE (1995) Introduction to artificial neural networks. In: Proceedings of IECON '95 21st annual conference on IEEE industrial electronics, vol 1, pp 33–37. <https://doi.org/10.1109/IECON.1995.483329>
14. Sangeetha K et al (2022) Women safety system using IoT. In: 2022 International conference on applied artificial intelligence and computing (ICAAIC), pp 1477–1480. <https://doi.org/10.1109/ICAAIC53929.2022.9792929>
15. Pravinth Raja S, Sheeba Rachel S, Sapna R (2021) Women's safety with a smart foot device. In: 2021 4th International conference on computing and communications technologies (IC CCT), pp 570–573. <https://doi.org/10.1109/ICCT53315.2021.9711778>
16. Suhas ML et al (2022) Self defence and safety monitoring system. In: 2022 4th International conference on advances in computing, communication control and networking (ICAC3N), pp 1472–1476. <https://doi.org/10.1109/ICAC3N56670.2022.10074098>
17. Kumar S et al (2022) Sentimental analysis of amazon customers using deep learning techniques. In: 2022 International conference on machine learning, computer systems and security (MLCSS), pp 259–266. <https://doi.org/10.1109/MLCSS57186.2022.00055>
18. Yu L (2022) Economic and financial data analysis system based on deep learning and neural network algorithm. In: 2022 International conference on knowledge engineering and communication systems (ICKES), pp 1–5. <https://doi.org/10.1109/ICKECS56523.2022.10060240>
19. Veenadhari S, Misra B, Singh CD (2014) Machine learning approach for forecasting crop yield based on climatic parameters. In: 2014 International conference on computer communication and informatics, pp 1–5. <https://doi.org/10.1109/ICCCI.2014.6921718>
20. Anjali T et al (2019) Temperature prediction using machine learning approaches. In: 2019 2nd International conference on intelligent computing, instrumentation and control technologies (ICICT), vol 1, pp 1264–1268. <https://doi.org/10.1109/ICICT46008.2019.8993316>
21. Naveen L, Mohan HS (2019) Atmospheric weather prediction using various machine learning techniques: a survey. In: 2019 3rd International conference on computing methodologies and communication (ICCMC), pp 422–428. <https://doi.org/10.1109/ICCMC.2019.8819643>
22. Poh CQ, Ubeynarayana CU, Goh YM (2018) Safety leading indicators for construction sites: a machine learning approach. *Autom Constr* 93:375–386. ISSN: 0926-5805. <https://doi.org/10.1016/j.autcon.2018.03.022>. <https://www.sciencedirect.com/science/article/pii/S0926580517309147>
23. Arbabzadeh N, Jafari M (2018) A data-driven approach for driving safety risk prediction using driver behaviour and roadway information data. *IEEE Trans Intell Transp Syst* 19(2):446–460. <https://doi.org/10.1109/TITS.2017.2700869>
24. Thakali L, Fu L, Chen T (2016) Model-based versus data-driven approach for road safety analysis: do more data help? *Transp Res Rec* 2601(1):33–41. <https://doi.org/10.3141/2601-05>
25. Shah N, Bhagat N, Shah M (2021) Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention. *Vis Comput Ind Biomed Art* 4(1):9. ISSN: 2524-4442. <https://doi.org/10.1186/s42492-021-00075-z>
26. Chun SA et al (2019) Crime prediction model using deep neural networks. In: Association for computing machinery. New York. ISBN: 9781450372046. <https://doi.org/10.1145/3325112.3328221>

AI-Driven Early Detection of Chronic Kidney Disease: Harnessing Deep Learning Techniques



Y. Umadevi, G. L. Vidyashree, K. Anitha, R. Kanagavalli,
and Kumaraswamy

Abstract The intricacy, nonlinearity, and heterogeneity of kidney pathophysiology pose a number of problems for conventional approaches of diagnosis and therapy that rely on hypothetical-deductive reasoning and linear statistics. An increasing amount of research, however, indicates that kidney disease could benefit greatly from decision support systems that are powered by artificial intelligence (AI) and are based on observed examples and patterns. These modern AI applications rival human accuracy in kidney tumor detection from imaging, identify modifiable risk factors linked to the onset and progression of chronic kidney disease (CKD), and improve decision-making and prognostication in transplantation of kidneys. Acute kidney injury (AKI) can be accurately predicted well before biochemical changes become apparent. The goal of this article is to address the growing global health issue of chronic kidney disease (CKD), which is mostly brought on by diabetes and high blood pressure. It emphasizes on the important elements of CKD, such as its causes, diagnostic indicators like Glomerular Filtration Rate (GFR), and the increased risk of early mortality in CKD patients. The work proposes a ground-breaking deep learning model customized for early CKD detection and prediction, using artificial intelligence, in recognition of the difficulties faced by medical professionals in early diagnosis. The performance of the unique deep learning model against cutting-edge machine learning approaches is rigorously compared as a key component of this research, which could revolutionize CKD diagnosis and enhance patient outcomes globally.

Y. Umadevi (✉)

Department of CSE, Government Polytechnic, Channapatna, India

e-mail: umanarendra.yeligar@gmail.com

G. L. Vidyashree

Department of CSE, Government Polytechnic, Ramanagara, India

e-mail: vidyavaishaka@gmail.com

K. Anitha · R. Kanagavalli · Kumaraswamy

Department of CSE, Global Academy of Technology, Bengaluru, India

e-mail: anitha.k@gat.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

N. R. Shetty et al. (eds.), *Advances in Artificial Intelligence and Machine Learning*,

Lecture Notes in Electrical Engineering 1335,

https://doi.org/10.1007/978-981-96-3105-6_28

Keywords Chronic kidney disease • Artificial intelligence • Acute kidney injury • Deep learning

1 Introduction

Chronic kidney disease (CKD) presents a substantial and escalating global health challenge, affecting an estimated 850 million individuals worldwide [1]. The kidneys, vital organs situated bilaterally just below the ribcage along the spine, play a pivotal role in preserving the body's internal equilibrium. Their primary function involves filtering the blood, effectively removing waste products, excess fluids, and harmful toxins, all of which are excreted through urine. Beyond waste filtration, the kidneys also serve as regulators of electrolyte levels, blood pressure, and the delicate acid–base balance within the body. Furthermore, they produce critical hormones that govern calcium metabolism and stimulate the production of red blood cells [2, 3]. In the twenty-first century, chronic kidney disease (CKD) has become a serious cause of death and suffering. In large part because of an increase in risk factors like obesity and diabetes mellitus, the prevalence of CKD has been rising. CKD reportedly afflicted 843.6 million individuals globally in the year 2019 [4]. CKD continues to be a major worldwide health concern despite advancements in the management of individuals with end-stage renal disease (ESKD) [5] that have reduced mortality. Studies on the Global Burden of Disease (GBD) have highlighted the fact that CKD has now become a primary cause of death globally. The identification, monitoring, and treatment of CKD are of the utmost significance, and it is equally important to put into place comprehensive preventative and therapeutic measures on a global scale to address this expanding health problem.

Chronic kidney disease (CKD) is defined by a steady decline in kidney function over time, making it more difficult for the kidneys to adequately filter waste from the blood and regulate the body's fluid and electrolyte levels. This dysfunction causes waste materials to build up and extra fluid to be retained, both of which can have serious health repercussions. CKD places a heavy burden on those who have it and increases the risk of a number of consequences, such as electrolyte imbalances, bone problems, hemophilia, and coronary artery disease [6, 7]. End-stage renal disease (ESRD), the most severe form of CKD that frequently requires life-sustaining therapies like dialysis or kidney transplantation, can develop if untreated or uncontrolled [8, 9]. It is crucial to identify CKD as soon as possible and to manage it correctly. The preservation of kidney function, slowed disease progression, and improved patient outcomes all depend on early detection of CKD and the use of the right treatment measures [10]. Healthcare professionals can assist patients in maintaining a higher quality of life and lessen the burden of difficulties brought on by advanced renal disease by treating CKD in its earlier stages. A common ailment called chronic kidney disease (CKD) is characterized by a decline in kidney function, which has a detrimental effect on a person's quality of life. One in ten individuals is afflicted, making it a serious global health problem. By 2040, it's expected to overtake heart

disease as the fifth greatest cause of death globally. CKD imposes enormous medical costs, especially in high-income countries where the cost of transplantation and dialysis can make up a sizable amount of the healthcare budget. Unfortunately, many people in low- and middle-income nations do not have access to kidney transplants [11] or life-saving dialysis, and it is predicted that renal failure will become more common in developing nations like China and India [12]. The body's ability to eliminate extra fluid and waste from the circulation is hampered by CKD, which increases the risk of consequences like high blood pressure, anemia, brittle bones, and nerve damage. The main indicator of renal function is an essential component in the diagnosis of CKD is the Glomerular Filtration Rate (GFR) [13]. GFR, a measurement of the kidneys' rate of blood filtering, is used by doctors to determine the severity of a condition. With or without reduced GFR (less than 60 mL/min/1.73 m²), CKD is defined as kidney damage lasting three months or more. Since GFR is the most reliable indicator of kidney function, it can assist distinguish between the several stages of CKD, each of which denotes a more severe decline in kidney function.

When it comes to recognizing the early stages of kidney damage, traditional methods for diagnosing chronic kidney disease (CKD), such as blood tests and urinalysis, may exhibit certain limitations. These techniques might also fall short in capturing variations in kidney health as it evolves over time. Routine screening is not appropriate for invasive procedures like kidney biopsy because of the hazards involved and the invasive nature of the operation. Similar to this, imaging scans, which can reveal important details regarding kidney shape and function, can be expensive and time-consuming [14, 15].

These challenges can be handled more efficiently by using the AI, Intelligent Machine Learning, and Deep Neural Network techniques. The healthcare sector has witnessed a notable upswing in the adoption of information technology. This uptick encompasses the widespread implementation of electronic health records (EHR) and the growing utilization of continuous monitoring methods. Moreover, the costs associated with the collection, storage, and retrieval of patient data have seen substantial reductions. These collective advancements have led to the accumulation of vast datasets, marking the onset of the era of big data in the realm of medicine. In essence, by utilizing the vast amount of data available, AI has the potential to change health care. It can help doctors make better decisions, make medical issues easier to spot early, and even make individualized treatment plans possible. The convergence of big data, artificial intelligence, and information technology presents a promising route for enhancing patient outcomes and quality of life.

The remaining sections of this article are structured as follows: review of the existing work of literature in Sect. 2 emphasize how unique our research is. The approaches used are described, and the proposed system framework is introduced in Sect. 3. Section 4 evaluates the results of proposed model. In Sect. 5, the comparisons between the proposed model and prior studies are worked out. In Sect. 6, the report concludes by exploring possible lines of inquiry for future work.

2 Review Work

In the diagnosis and prognosis of serious illnesses, machine learning models have demonstrated their efficacy. The early diagnosis of chronic diseases, with an emphasis on examining novel therapies for chronic kidney disease (CKD), is one of the topics attracting substantial interest from medical professionals and researchers recently. Recent research has demonstrated the potential of deep learning and machine learning models for detecting CKD with accuracy. The various existing works related to CKD using Intelligent Machine Learning and Deep Neural Network are discussed below:

To forecast chronic kidney disease (CKD), Krishnamurthy et al. [16] created different types of artificial intelligence models. Important CKD prediction factors, such as aging arthritis, hypertension, sulfate use, and Lisinopril use, were successfully detected using the LightGBM approach. Convolutional Neural Networks (CNNs) outperformed other models among the many ones investigated, reaching a remarkable AUROC measure of 0.954. The main objective of the study was to develop a machine learning model using comorbidity and medication data from Taiwan's National Health Insurance Research Database to predict the onset of CKD during the next 6 or 12 months, elucidating its prevalence in the population. The prediction model was built on the basis of demographic, medication, and comorbidity data collected over a two-year period from a cohort of 18,000 people with CKD who had been diagnosed and 72,000 matched controls. In addition to its predictive skills, this model offers policymakers hope by facilitating trend forecasting, proactive risk monitoring, early CKD detection, resource optimization, and patient-centric management options.

In order to create a neural network classifier for chronic kidney disease (CKD), Vasquez-Morales et al. [17] used a sizable dataset relevant to CKD and achieved an astounding 95% accuracy rate. An individual's risk for developing CKD can be predicted using a classifier built on a neural network, according to their study. Two independent groups—those in Colombia who were diagnosed with CKD in 2018 and a control group without CKD diagnoses—are used to train the model, which uses demographic and medical data from both. The model performed incredibly well at predicting the risk of CKD after training and evaluation. The integration of eXplainable AI (XAI) concepts was prompted by the intrinsic difficulty neural networks which have in explaining their predictions.

In their work, Xiao et al. [18] investigated how well different machine learning models could predict the course of chronic kidney disease (CKD). They divided CKD outcomes into mild, moderate, and severe groups using data from 551 individuals with proteinuria and 18 characteristics. Based on performance metrics like AUC, sensitivity, specificity, accuracy, log-loss, and precision, nine predictive models were created and compared, including logistic regression, Elastic Net, lasso regression, ridge regression, Support Vector Machine, Random Forest, XGBoost, neural network, and k-nearest neighbor. The model that performed the best was logistic regression, which had an AUC of 0.873 and AUCs for sensitivity and specificity of 0.83 and 0.82, respectively. Notably, several blood-derived tests, involving as ALB,

Scr, TG, LDL, and EGFR, were discovered to be important predictors of the severity of CKD, while others, such as CRP, HDL, and SNA, had a less significant effect. This study represents a useful tool for clinical practice by suggesting the potential use of non-urinary clinical factors, such as features of standard blood tests, for predicting nephritis development in patient follow-up.

Ma et al. [19] created an advanced Deep Neural Network model for the identification of chronic renal illness as part of their research. In comparison to Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs), the proposed model has higher accuracy. In this study, they introduce the Heterogeneous Modified Artificial Neural Network (HMANN), which was created for the specific purpose of performing early detection, segmentation, and diagnosis of chronic renal failure within the context of the Internet of Medical Things (IoMT). The backpropagation (BP) technique is included in the HMANN, which combines characteristics of the Multilayer Perceptron (MLP) and Support Vector Machine algorithms. The approach works by preprocessing ultrasound pictures, then segmenting the kidney region of interest within those images. The suggested HMANN approach considerably minimizes the time needed for contour delineation while displaying outstanding segmentation accuracy for the kidney.

Shi Yin and colleagues, as referenced in [19], introduced the Subsequent Boundary Distance and Pixel-wise Classification networks (SBD-PCN) designed for the automated segmentation of kidney structures in ultrasound images. Initially, the authors employed pre-trained Deep Neural Networks to perform image classification, discerning high-level features in ultrasound images. These extracted features were subsequently utilized as inputs for an intermediate distance regression network, facilitating the learning of kidney boundary maps. Finally, the recognized boundary distance maps were used to classify kidney pixels, contributing to accurate kidney segmentation.

Hermesen et al. developed a Convolutional Neural Network (CNN) model with the objective of segmenting different compartments within the kidney. Their dataset comprised a combination of samples from various sources, including kidney transplants with sample size $n = 101$, living donor kidneys with data $n = 10$, and nephrectomies for Renal Cell Carcinoma (RCC) with data $n = 15$, drawn from two different institutions. To prevent overfitting, they conducted internal cross-validation before external testing. The reference standard for evaluation was pathologist reading assisted by ASAP software for annotation. Performance assessment was based on the Dice coefficient (DC), a metric measuring image similarity. The highest DC scores were observed for glomeruli, followed by the interstitium, capsule, and proximal tubules. However, challenges were encountered in correctly classifying sclerotic glomeruli and empty Bowman's space, potentially attributable to the limited sample size.

Polat et al. [20] in their study examined the value of efficient feature selection methods in improving the precision of chronic kidney disease (CKD) prediction. In order to reduce the dimensionality of the CKD dataset, their research used both wrapper and filter feature selection techniques. A Support Vector Machine (SVM) was then fed the chosen features in order to classify and diagnose CKD. The testing

results clearly showed that the SVM outperformed conventional wrapper and filter selection techniques, especially when using the Best First search strategy with a filtered subset evaluator. Particularly, SVM accomplished a remarkable accuracy rate of 98.5% in CKD diagnosis. SVM was used in their study to diagnose CKD with an emphasis on lowering dataset dimensionality using two basic types of feature selection methods: wrapper and filter approaches. These findings highlight the effectiveness of SVM in the accurate identification of chronic kidney disease, especially when paired with filtered subset evaluation and the Best First search engine.

Hassan, Md Mehedi et al. in their study presented a novel method for improving the accuracy of the diagnosis of chronic kidney disease (CKD), which is based on the extraction of the best features from clinical datasets. The author used a number of well-known machine learning methods to predict CKD using datasets from CKD patients. The proposed approach included dealing with missing data and running K-means clustering. The model was developed with application of the XGBoost feature selection algorithm to the feature selection process. In order to find the most accurate classification models, we subjected our dataset to a variety of machine learning models, such as Neural Network (NN), Random Forest (RF), Support Vector Machine (SVM), Random Tree (RT), and Bagging Tree Model (BTM). On the basis of accuracy, sensitivity, specificity, and Kappa values, the performance of the model was assessed. This thorough method enables more accurate CKD diagnosis.

3 Proposed Ensembled Deep Neural Network Model to Predict the CKD

Methods for predicting CKD that are based on clinical criteria, biomarkers, and medical history have some significant limitations. They frequently aren't equipped to identify subtle and complex trends in CKD data, which could delay or erroneously diagnose the condition. The interactions and nonlinear correlations between various risk factors may be difficult for these strategies to take into consideration. Additionally, they could be less able to adapt to changing CKD risk factors and may not be able to manage high-dimensional data, such as genetic or imaging data.

However, the introduction of ensemble Deep Neural Network models offers a potential solution. These models may extract complex features from various data sources, including unstructured data types like medical photos and clinical notes, by leveraging the capabilities of deep learning and merging several neural networks. Ensemble models enable more precise and timely CKD predictions by adjusting to shifting data distributions. They represent an important development in healthcare decision-making due to their capacity to recognize intricate patterns and manage large amounts of data.

3.1 Problem Statement

Timely and accurate diagnosis of chronic kidney disease (CKD) is of paramount importance and can significantly impact a patient's well-being. To achieve precise insights into identifying renal diseases, medical professionals have traditionally relied on fundamental methods, including medical examinations and laboratory tests such as blood tests to measure the Glomerular Filtration Rate (GFR), a crucial indicator of kidney function, and urine tests to assess albumin levels, providing insights into kidney health. Given the potential wealth of valuable data sources available for medical diagnosis, it becomes imperative to develop robust and widely applicable diagnostic models. The proposed ensemble model aims to assist healthcare experts in making informed and timely decisions, ultimately enhancing the accuracy and efficiency of CKD diagnosis.

Ensemble Deep Neural Network CKD model

Various computer algorithms are used to predict chronic kidney disease (CKD). The Random Forest algorithm, which has historically been used in medical research and healthcare settings, has shown to be an effective technique for CKD prediction. As an ensemble learning method, Random Forest combines different decision trees to produce predictions. Nevertheless, employing ensemble deep learning techniques is a different strategy to take into account. These models have the advantage of capturing complex patterns and correlations within CKD data since they are made up of interconnected neural networks, comparable to decision trees in Random Forest. Adopting ensemble deep learning can produce predictions for the diagnosis of chronic renal disease that are more reliable and precise.

Chronic kidney disease (CKD), which may coexist with other diseases like hypertension, liver diseases, anemia, and heart diseases, is greatly impacted by the aforementioned characteristics, such as decreased albumin levels, increased blood pressure, and a significant decrease in white blood cells. The phases of our suggested framework are depicted in Fig. 1 and consist of the chosen dataset, initial steps, methods for selecting pertinent attributes, model design, and optimization procedures. Each of these procedures will be explained in detail in the sections as follows:

Data Descriptions

The benchmark chronic kidney disease dataset is being used for model to train, which was obtained from the UCI machine learning repository [21]. Numerous academics have used this dataset extensively for their empirical investigations [22]. A total of 400 cases are covered, including 150 cases of negative CKD and 250 cases of positive CKD. The 24 attributes in the dataset used in our investigation can be divided into 13 category features and 11 numeric features. The class label is binary, with 1 denoting a positive state of CKD and 0 denoting a state of CKD. It's critical to include a variety of pertinent variables that may be able to detect the presence of chronic kidney disease (CKD) in order to build an efficient dataset for CKD analysis. Age, blood

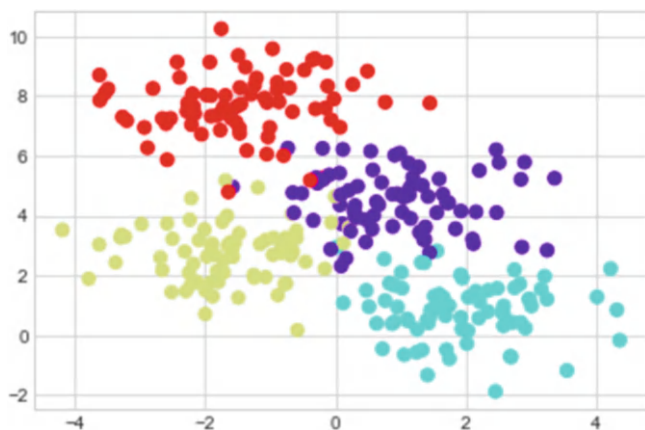


Fig. 1 K-means clustering for grouping dataset

pressure, serum creatinine, blood sugar, and urine test results are only a few examples of these characteristics. Age is regarded as a crucial factor because the risk of CKD frequently rises with advancing age. The relationship between kidney function and blood pressure readings reveals information about cardiovascular health. Serum creatinine concentrations are important markers of renal function, with increased concentrations indicating probable impairment. Particularly in cases of CKD caused by diabetes, blood glucose levels are crucial. Important information can be obtained from urine testing, such as the presence of albumin or proteinuria, both of which are signs of kidney impairment.

Data Preprocessing

Medical datasets frequently experience a variety of problems, including the occurrence of null values or severe outliers, which are usually linked to malfunctioning sensors, network problems, or data entry errors. Machine learning (ML) and deep learning (DL) models' performance can be greatly hampered by these data-related issues. As a result, the main goal of this phase is to improve data quality by resolving outliers and missing information. Given that the dataset contains both category and numerical variables, data encoding is crucial. Notably, ML and DL tend to perform better with numerical qualities than with categorical ones when choosing features. To overcome this, all category features are encoded using the label encoder module from the Scikit-learn toolkit.

Depending on the volume of missing data and the importance of the missing feature, managing missing values calls for a careful strategy. Traditional statistical techniques in the proposed model, where the number of missing values is quite small, such as mean imputation is used. Furthermore, the significance of locating and managing outliers is taken care, which are values that greatly deviate from the normal distribution of feature values and might impair the robustness of a model. Our strategy used statistical analysis to find outliers and a medical viewpoint to validate

their validity. The mean value for each feature was then used to replace any dataset outliers that were found.

K-Means Clustering

In the data analysis process, one crucial technique employed is data clustering, which aims to categorize or group data points within a dataset based on their inherent similarities or patterns. In our case, we employed the K-means clustering algorithm to achieve this objective with our dataset. The selection of the optimal number of clusters, a critical decision in the K-means process, was determined using a technique known as the “elbow method.” This method involves running the K-means algorithm for a range of cluster numbers and then plotting the results. The point at which the plot demonstrates an “elbow” or a significant change in the within-cluster sum of squares (WCSS) is typically chosen as the optimal number of clusters. In our analysis, this optimal number was found to be three, suggesting that the data could be effectively categorized into three distinct clusters. These clusters, each represented by a centroid point, were then used in the K-means clustering process.

Figure 1 visually illustrates the steps involved, including (a) the elbow method, (b) the K-means clustering outcome, and (c) the Dendrogram representation of the resulting clusters.

The K-means clustering technique was employed to help reveal inherent patterns and groupings within our dataset, aiding in the subsequent analysis and interpretation of chronic kidney disease-related data.

4 Methodology

An important first step in creating a reliable predictive model is data separation. The dataset is split into two separate subsets: a training set and a testing set. This division’s main goal is to make it easier to create and assess the Random Forest model. The training set is essential to the development of the model. It serves as the basis for training the Random Forest algorithm to find patterns and connections in the data. The model gains the ability to predict outcomes based on the input features during training, making it more adept at detecting cases of chronic kidney disease (CKD).

The AdaBoost Classifier, a deep learning algorithm that makes use of ensemble learning, is trained during the model training phase. Using the training dataset, a succession of weak learners—often represented as decision trees—is sequentially trained. AdaBoost stands out because of its iterative methodology, which emphasizes cases that were previously misclassified while correcting the mistakes committed by earlier weak learners. Until a predefined number of weak learners have been included into the ensemble or a suitable level of performance is reached, this recurrent refining continues. The AdaBoost Classifier then combines these distinct decision trees into a strong ensemble model, utilizing their combined advantages to improve overall forecast accuracy. For tasks like diagnosing chronic kidney disease, where complex

and subtle data patterns may be present and call for a collaborative and adaptive learning technique, an ensemble approach is very beneficial.

The process of hyperparameter tuning entails methodically examining various combinations of these hyperparameters to find the best values that enhance the model’s capacity for generalization and prediction. When searching across the hyperparameter space, methods like grid search and random search are frequently used to assess the model’s performance on a validation set for each combination of hyperparameter. In order to ensure that AdaBoost performs to its fullest capacity when used for tasks like the detection of chronic renal disease, the objective is to identify the hyperparameter values that produce the optimum model performance.

4.1 Random Forest

The Random Forest Classifier is a powerful Intelligent Machine Learning algorithm built on the idea of ensemble learning. By combining various independent classifiers, its main goal is to improve model performance and address complicated issues. Named after the algorithm, this classifier comprises a large number of decision trees; each built on a portion of the information, with their predictions then averaged to enhance overall predictions. The Random Forest algorithm generates forecasts from each decision tree instead of depending on just one and bases its final prediction on the majority of decision votes. This ensemble strategy dramatically lowers the risk of overfitting, a key worry in machine learning. Figure 2 represents the general model of Random Forest.

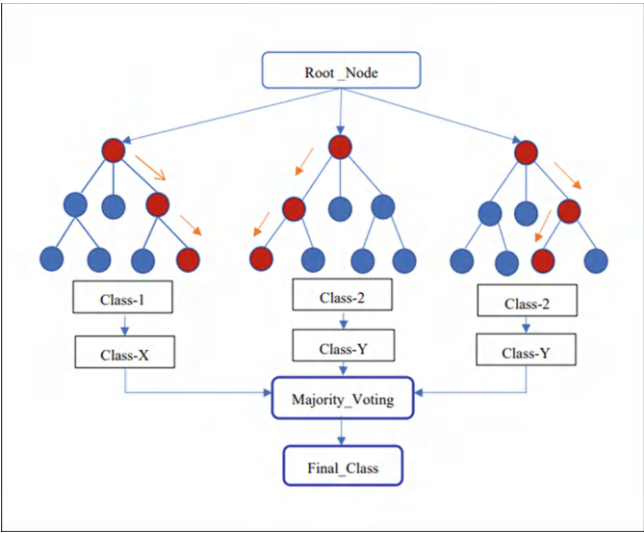


Fig. 2 General model of Random Forest

The technique uses a variety of decision trees to forecast classes inside the database, some of which may correctly predict outcomes and others of which may not. As a result, two fundamental presumptions govern prediction accuracy. In order for the algorithm to predict an outcome with greater accuracy than mere estimation, the feature variable values must first be included. Second, there shouldn't be much association between the forecasts each tree makes. The Random Forest Classifier is an effective tool for tasks requiring data-driven decisions since these circumstances are necessary for achieving high forecast accuracy.

A decision tree ensemble method called Random Forest builds several trees and predicts by combining their results. It also provides insights into robustness and feature relevance. Another ensemble strategy is called AdaBoost, which sequentially instructs unreliable learners while adjusting their weighting depending on prior failures, frequently leading to increased accuracy. AdaBoost's strength rests in enhancing the performance of poor learners, however with somewhat less interpretability than Random Forest, which provides feature importance ratings, making it robust and interpretable.

5 Model Development

Figure 3 provides a comprehensive overview of the model's architectural framework, which encompasses three pivotal phases: Preprocessing, model hyper tuning, and classification. Among these, the preprocessing phase assumes paramount importance, as it addresses potential data imperfections, including noise and redundancy, to ensure the data's quality and reliability for subsequent analysis.

Within the preprocessing phase, an array of techniques is systematically applied to refine the chronic kidney disease (CKD) dataset. These techniques include the handling of missing values, encoding of categorical data, data transformation procedures, the identification and removal of outliers and extreme values, as well as feature selection. Given the potential presence of noise and irregularities in real-world healthcare data, meticulous data preparation is crucial to minimize their impact on subsequent modeling.

Furthermore, the CKD dataset is logically partitioned into distinct training and testing subsets following preprocessing. To enhance model efficiency and mitigate the risk of overfitting, a judicious feature selection approach is employed, specifically Recursive Feature Elimination (RFE). This technique evaluates the relevance and significance of each feature within the dataset, facilitating a reduction in processing complexity. Through RFE, the most salient and informative features are identified and retained, while redundant and extraneous attributes are filtered out.

Ultimately, the learning model is primed with this distilled set of essential characteristics, setting the stage for the subsequent Classification phase. This meticulous approach to model development ensures that the model operates on a refined and relevant feature set, optimizing its capacity to discern and predict chronic kidney disease, a task of critical importance in healthcare decision-making.

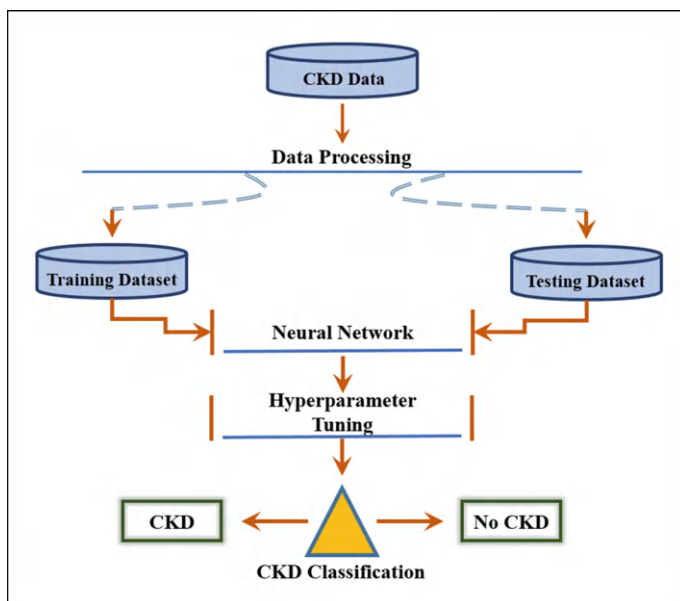


Fig. 3 Proposed model architecture

6 Result Analysis and Discussions

Performance evaluation of the model developed is discussed below.

Evaluating the model: The effectiveness of deep learning models for predicting chronic kidney disease (CKD) is frequently evaluated using a number of important criteria. Accuracy, precision, recall, and the F1 score are some of these metrics [23]. Same is been discussed below for the prediction of CKD.

Accuracy: Accuracy measures the proportion of correct predictions made by the CKD prediction model. In the context of CKD, this means the percentage of patients whose disease status was accurately classified by the model. An accurate model would correctly identify both CKD-positive and CKD-negative cases.

Precision: Precision assesses the quality of positive predictions made by the CKD model. In CKD prediction, precision signifies the model's ability to correctly identify patients as CKD-positive among those it has classified as positive. In other words, it quantifies how many of the predicted CKD-positive cases are genuinely positive.

Recall: Recall, also known as sensitivity, determines the model's capability to detect relevant CKD data. It represents the percentage of actual CKD-positive patients that the model correctly identifies. In the context of CKD, high recall indicates that the model is proficient at capturing true CKD cases, minimizing the chances of false negatives.

F1 Score: The F1 score is a metric that balances recall and precision. It calculates the harmonic mean between these two values. In CKD prediction, the F1 score accounts for both the model’s ability to accurately identify CKD-positive cases (precision) and its capacity to capture most of the actual CKD-positive patients (recall). It provides a single metric that gauges overall model performance while considering the trade-off between precision and recall.

The datasets after feature selection are given as input to train the model. The sample data is shown in Fig. 4

Figure 5 is a heat map designed to offer a comprehensive overview of kidney health status. Specifically, this heat map provides valuable information by utilizing a color-coded system to indicate the risk levels associated with the progression of chronic kidney disease (CKD) and the likelihood of developing heart disease. It serves as a valuable tool for guiding the course of treatment and healthcare decisions. The CKD heat map employs a combination of numerical values and distinct colors to convey critical information regarding kidney health. By analyzing this heat map, healthcare professionals and patients can gain insights into the potential severity of CKD and its impact on overall health.

The color-coded scheme used in the heat map allows for quick and intuitive interpretation. Different colors are used to represent various risk levels.

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	36.0	1.2	NaN	NaN	15.4	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	18.0	0.8	NaN	NaN	11.3	38	6000	NaN	no	no	no	good	no	no	ckd
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	53.0	1.8	NaN	NaN	9.6	31	7500	NaN	no	yes	no	poor	no	yes	ckd
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	56.0	3.8	111.0	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	26.0	1.4	NaN	NaN	11.6	35	7300	4.6	no	no	no	good	no	no	ckd

Fig. 4 Input data with the features filtered and selected

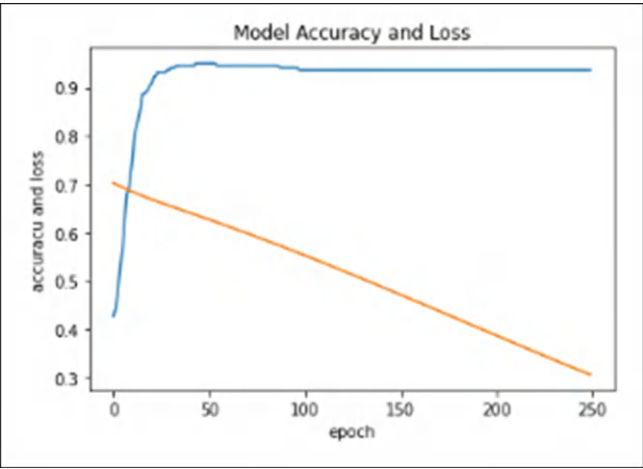


Fig. 5 Model accuracy representation

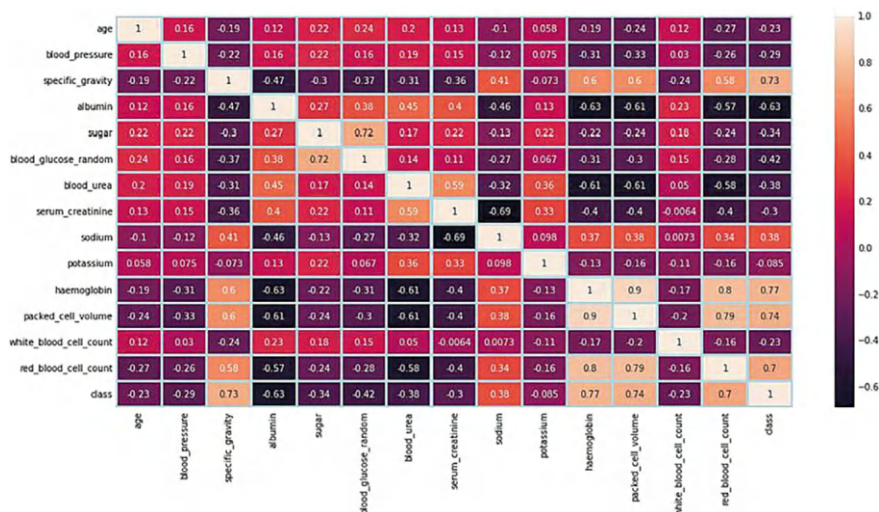


Fig. 6 Heat map representation of CKD prediction of the proposed model

These colors serve as visual cues to help individuals and medical practitioners easily identify the degree of risk associated with kidney disease progression and its connection to the risk of developing heart disease. This aids in making informed decisions regarding treatment strategies, lifestyle adjustments, and necessary interventions.

In essence, the chronic kidney disease heat map given in Fig. 6 is a valuable resource in the management of kidney health, offering a clear and accessible way to assess the risks associated with CKD progression and its relationship to heart disease. It empowers healthcare providers and patients alike with the information needed to tailor treatment plans and mitigate potential health risks effectively.

7 Conclusion

A novel hybrid model has been put out that considerably boosts accuracy rates by utilizing the advantages of both the AdaBoost and Random Forest algorithms. This novel method shows promise for accurate clinical use and is useful for the diagnosis of chronic kidney disease (CKD). The hybrid methodology may also be used in the medical field to diagnose a number of other disorders. Future improvements could be made by getting a larger dataset that includes a variety of patient records from multiple healthcare facilities. With expectations of a system that is more effective and accurate as dataset quantity and quality continue to increase, this extension intends to improve prediction accuracy.

References

1. Arif MS, Mukheimer A, Asif D (2023) Enhancing the early detection of chronic kidney disease: a robust machine learning model. *Big Data Cogn Comput* 7(3):144
2. Wadei HM, Textor SC (2012) The role of the kidney in regulating arterial blood pressure. *Nat Rev Nephrol* 8:602–609
3. Mukoyama M, Nakao K (2005) Hormones of the kidney. *Basic Clin Princ* 353–365
4. Jager KJ, Kovesdy C, Langham R, Rosenberg M, Jha V, Zoccali C (2019) A single number for advocacy and communication—worldwide more than 850 million individuals have kidney diseases. *Nephrol Dial Transplant* 34(11):1803–1805
5. Saran R, Robinson B, Abbott KC, Bragg-Gresham J, Chen X, Gipson D, Gu H et al (2020) US renal data system 2019 annual data report: epidemiology of kidney disease in the United States. *Am J Kidney Diseases* 75(1):A6–A7
6. Webster AC, Nagler EV, Morton RL, Masson P (2017) Chronic kidney disease. *Lancet* 389:1238–1252
7. Kalantar-Zadeh K, Jafar TH, Nitsch D, Neuen BL, Perkovic V (2021) Chronic kidney disease. *Lancet* 398:786–802
8. Hall ME, do Carmo JM, da Silva AA, Juncos LA, Wang Z, Hall JE Obesity, hypertension, and chronic kidney disease. *Int J Nephrol Renov Dis* 75–88
9. Ghaderian SB, Hayati F, Shayanpour S, Mousavi SSB (2015) Diabetes and end-stage renal disease; a review article on new concepts. *J Ren Inj Prev* 4:28
10. Wagner LA, Tata AL, Fink JC (2015) Patient safety issues in CKD: core curriculum 2015. *Am J Kidney Dis* 66:159–169
11. 2020 Wkd Theme. Available online: <https://www.worldkidneyday.org/2020-campaign/2020-wkd-theme>
12. Jha V, Garcia-Garcia G, Iseki K, Li Z, Naicker S, Plattner B, Saran R, Wang AY-M, Yang C-W (2013) Chronic kidney disease: global dimension and perspectives. *Lancet* 382:260–272
13. Natakaional Kidney Foundation (2020) Available online: <https://www.kidney.org/kidneydisease/global-facts-about-kidneydisease>
14. Chen TK, Knicely DH, Grams ME (2019) Chronic kidney disease diagnosis and management: a review. *JAMA* 322:1294–1304
15. Ferguson MA, Waikar SS (2012) Established and emerging markers of kidney function. *Clin Chem* 58:680–689
16. Krishnamurthy S, Kapeleshh Ks, Dovgan E, Luštrek M, Piletič BG, Srinivasan K, Li Y-C, Gradišek A, Syed-Abdul S (2021) Machine learning prediction models for chronic kidney disease using national health insurance claim data in Taiwan. *Healthcare* 9(5):546
17. Vásquez-Morales GR, Martínez-Monterrubio SM, Moreno-Ger P, Recio-García JA (2019) Explainable prediction of chronic renal disease in the Colombian population using neural networks and case-based reasoning. *IEEE Access* 7:152900–152910
18. Xiao J, Ding R, Xu X, Guan H, Feng X, Sun T, Zhu S, Ye Z (2019) Comparison and development of machine learning tools in the prediction of chronic kidney disease progression. *J Transl Med* 17(1):1–13
19. Ma F, Sun T, Liu L, Jing H (2020) Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network. *Futur Gener Comput Syst* 111:17–26
20. Polat H, Mehr HD, Cetin A (2017) Diagnosis of chronic kidney disease based on support vector machine by feature selection methods. *J Med Syst* 41:1–11
21. Chronic Kidney Disease Dataset. Available: online: https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease. Accessed on 8 Feb 2023
22. Poonia RC, Gupta MK, Abunadi I, Albraikan AA, Al-Wesabi FN, Hamza MA (2022) Intelligent diagnostic prediction and classification models for detection of kidney disease. *Healthcare* 10:371

23. Ghosh P, Azam S, Hasib KMd, Karim A, Jonkman M, Anwar A (2021) A performance based study on deep learning algorithms in the effective prediction of breast cancer. In: 2021 International joint conference on neural networks (IJCNN). IEEE, pp 1–8

Interaction Modeling in Logistic Regression: An Analysis of Dropout Regularization and Adaptive Optimization



Niveditha Kundpuram , Subha Vidisha Atilli , and K. S. Srinivas 

Abstract The introduction of regularization techniques and neural networks in the field of Machine Learning has garnered substantial research. This provides sufficient knowledge on the effects of dropouts and optimizers in the case of classical neural networks. However, the consequences of the same using logistic regression, across various interaction degrees, haven't been thoroughly studied. In this paper, we aim to discuss the effects of dropouts with different optimizers (of different values for the hyperparameters—momentum and dropout probability) on interaction degrees varying from 1 to 3 and demonstrate, both mathematically and empirically, that the best performance for a logistic regression model is a combination of Adam optimizer and Gaussian dropouts with Interaction degree = 2.

Keywords Dropout · Optimizer · Logistic regression · Interaction degree · Momentum

1 Introduction

Most machine intelligence algorithms use regularization approaches to lessen the need to memorize noisy input and enhance the functionality of the neural network model. One such strategy is dropout. It is a regularization strategy that avoids overfitting by choosing which nodes to ignore during a model's training phase.

N. Kundpuram (✉) · S. V. Atilli (✉)
CSE Department, PES University, Bengaluru, India
e-mail: nivedithakund@gmail.com

S. V. Atilli
e-mail: subhavidisha@gmail.com

K. S. Srinivas
CSE (AI/ML) Department, PES University, Bengaluru, India
e-mail: srinivas.katharguppe@gmail.com

Furthermore, this improves robustness. The model's internal parameters, or the model weights and biases, are updated using optimizers. Optimizers are used to update the internal parameters of the model, i.e., model weights and biases. Different optimizers use different hyperparameters such as momentum and learning rate, thereby adjusting the rate at which the model parameters are up-to-date.

Dropouts are employed to lessen overfitting and avoid memorization, and optimizers are included to update the model's parameters to minimize the loss function and lower the model's total error.

By capturing the correlation between variables, we hope to enhance the model's overall performance by incorporating interaction variables. Therefore, we propose to enhance the performance of logistic regression on imbalanced, noisy, or sparse data by reducing overfitting, enhancing resilience, and accelerating convergence by introducing optimizers, dropouts, and different degrees of interaction variables.

2 Data and Methods

The dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases, collected from females at least 21 years old of Pima Indian heritage. Based on diagnostic measurements, the outcome is binary (1 or 0, whether a patient has diabetes or not respectively). The features covered by the dataset are—(i) glucose concentration, (ii) no. of pregnancies, (iii) Body Mass Index (BMI), (iv) age (in years), (v) diabetes pedigree function, (vi) triceps skin fold thickness (mm), and (vii) insulin (μ U/ml).

2.1 Model Construction and Implementation

A customized logistic regression model with fit, forward, predict, and dropout scheduler was created. This model was trained on the diabetes dataset with different optimizers and dropouts. The following are the dropouts and optimizers employed.

Dropouts

1. Standard Dropout

This dropout refers to ignoring units (i.e., neurons) during the training phase of a certain set of neurons that are selected at random. Individual nodes are either removed from the network with probability $1 - p$ or maintained with probability p at each training stage, leaving a smaller network behind. A dropped-out node also loses its incoming and outgoing edges.

2. Alpha Dropout

To maintain the self-normalizing feature even after this dropout, the alpha dropout maintains the mean and variance of inputs at their original levels. The output of

alpha dropout preserves the original mean and standard deviation of the input for an input with zero mean and unit standard deviation [1]. Using samples from a Bernoulli distribution, it randomly masks portions of the input tensor’s components with probability p during training. Every forward call involves randomization of the items to be hidden, along with scaling and shifting to maintain a zero mean and a single standard deviation. The module merely computes an identity function during evaluation.

3. *Gaussian Dropout*

It is used to apply multiplicative 1-centered Gaussian noise. Instead of setting some dimensions to 0, it multiplies a normal distributed noise with all the concealed embedding dimensions. As a result, even though the standard deviation and variance alter, the outputs’ mean will remain the same. The gate following each node in a standard dropout is a Bernoulli gate with a probability of $(1 - p)$ when the gate is closed (resulting in output = 0 and effectively no neuron weight associated). The alternative result is that, with probability p , the gate will be open, allowing the signal to pass through as-is with a gate output of 1. In the case of Gaussian dropout, the Bernoulli gate is simply replaced by a Gaussian gate with zero mean and standard deviation equal to the activation of the unit.

4. *Monte Carlo Dropout*

To make predictions that are more accurate and to calculate the uncertainty of forecasts, the Monte-Carlo dropout uses dropout regularization. Monte Carlo Dropout, as explained in Fig. 1, differs from other dropouts as it applies dropout even during inference. This gives the model’s predictions’ level of uncertainty a more precise measurement [2]. It is applied at both training and test time. By doing so, the model is introducing an element of non-determinism in predictions. Depending on the nodes that are deactivated or ignored, there is randomness introduced. Therefore, given the same data point, your model could predict different values each time. Bayesian interpretation, or making random predictions and interpreting them as samples from a probabilistic distribution, is the main objective of Monte Carlo dropout.

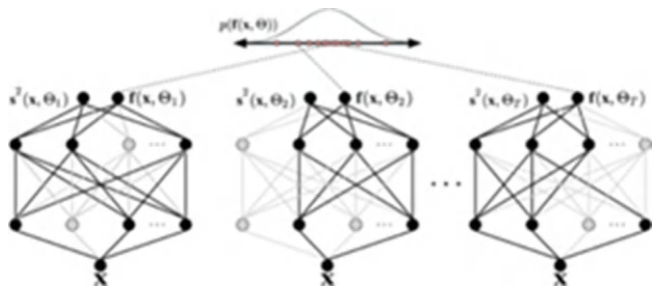


Fig. 1 Each dropout configuration yields a different output by randomly switching neurons off (gray circles) and on (black circles) with each forward propagation. Multiple forward passes with different dropout configurations yield a predictive distribution over the mean $p(f(x, \theta))$

5. Scheduled Dropouts

Exact dropout probability estimation is challenging and model-dependent. Furthermore, a model with a high dropout probability underfits the data samples, whereas a model with a low dropout probability overfits the data samples. These observations inform the use of a dropout scheduling training strategy [3]. We add a dropout layer before the output layer of the model to be trained in the suggested training approach. A high dropout probability is used to initialize the dropout layers. Additionally, the dropout probability decays linearly or polynomially with each training phase. The hyperparameter r controls how quickly this decay occurs, which is stated in terms of the maximum number of training iterations.

Optimizers

1. Stochastic Gradient Descent (SGD)

It is an optimizer that uses an iterative process to change the weights. Since it calculates the gradient stochastically across different subsets of the data as opposed to doing it across the entire dataset at once, it approximates the Gradient Descent approach. Let J be the loss function that we wish to minimize defined as such

$$Q(w) = \frac{1}{\alpha} \sum_{i=1}^n Q_i(w). \quad (1)$$

In Eq. 1, Q here is defined as a function of the parameter w , i.e., the weights of the model. SGD aims to minimize the above by changing weights at each step using α as the learning rate and n as the number of steps.

2. Adam (Adaptive Moment Estimation)

Adam optimizer involves a combination of two Gradient Descent methodologies: Gradient Descent with momentum and RMSprop. Thus, this optimizer gains the advantages of these two methods and expands on them to provide a Gradient Descent that is much more optimized [4]. It uses the same ideas behind exponential moving averages and exponential weighted averages that are utilized in RMSprop and the Gradient Descent with Momentum. This makes the optimizer's learning process stable and allows it to quickly converge to the global minima while avoiding learning oscillations.

Interaction Variables

The term “interaction” in statistics refers to a situation where the outcome of one causative variable depends on the state of a second causal variable (i.e., when the effects of the two causes are not additive) and may occur when analyzing the link among three or more variables, as explained in Fig. 2.

Although interactions are frequently associated with causal connections, they can also refer to non-causal linkages (also known as moderation or effect modification).

In the case of 1st degree interaction variables,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

In the case of 2nd degree interaction variables,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$$

In the case of 3rd degree interaction variables,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1 x_2 + \beta_5 x_2 x_3 + \beta_6 x_1 x_3 + \beta_7 x_1 x_2 x_3 + \epsilon$$

Fig. 2 Equations of interaction variables

In other words, interaction terms allow you to determine whether the value of one independent variable affects the relationship between the target and the independent variable.

Momentum

Momentum is a Gradient Descent optimization strategy that accelerates learning by adding a portion of the previous update vector to the present update vector. It is a technique for minimizing oscillations and speeding up convergence by smoothing out model parameter updates and enabling the optimizer to move forward in the same direction as before. The optimizer uses the exponentially weighted moving average of the prior gradients rather than updating the parameters with the current gradient. It is a strong optimization approach that can speed up convergence, minimize oscillations, avoid local minima, and make the optimization process more resistant to noisy gradients. It is used with the SGD optimizer.

We constructed two data models with interaction degrees 2 and 3. We then trained the logistic regression model on the different data models with different combinations of dropouts and optimizers. Hyperparameter tuning was employed to implement different momentum values for the SGD optimizer and different dropout probabilities for Alpha, MC, Gaussian, and Standard dropouts. Our base case was training the model with different optimizers and no dropouts.

3 Results

We employed the data model of both interaction degrees 2 and 3 with both optimizers, but no dropouts. These form our baseline models.

Using standard dropouts with interaction degree 2, under SGD optimizer, we see that the performance is the same across all momentum values for dropout probability (dropout probability) 0.2 and 0.3.

For momentum values 0.8 and 0.93, we see an oscillation in terms of the $F1$ -score, but there is a gradual increase as the dropout probability increases to 0.9, at which a momentum value of 0.93 gives the best performance. The highest $F1$ -score of 0.54 is achieved at four different dropout probabilities—at 0.2 and 0.3 by all the momentum

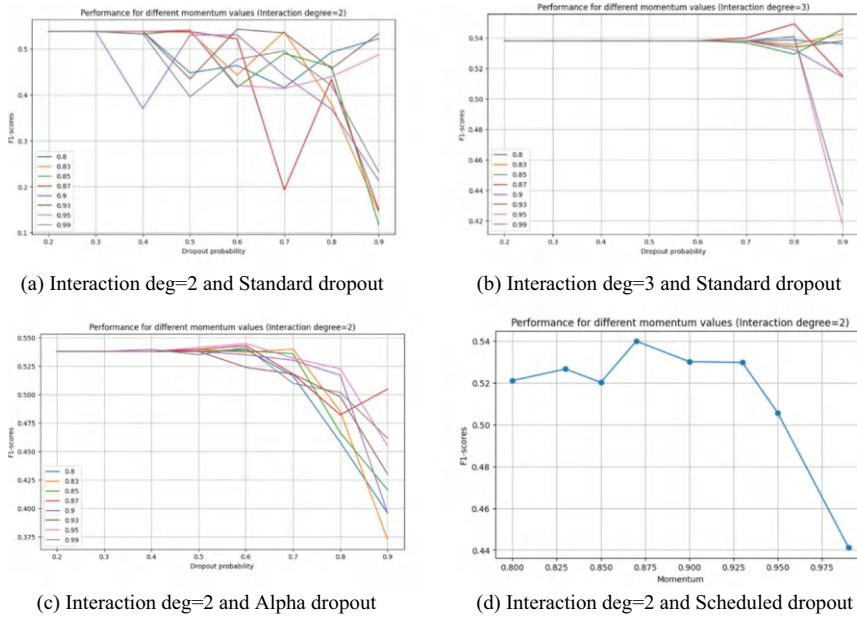


Fig. 3 Graphs showing the performance of the LR model on a data model of different interaction degrees, trained with SGD optimizer with different dropouts

values, at 0.4 by all momentum values except 0.85, 0.9 and 0.99, at 0.4 by 0.85 and 0.95, and finally at 0.6 by momentum value 0.93. This can be observed in Fig. 3a. The same approach was implemented with interaction degree 3, which proved to be overfitting with the data model, as observed in Fig. 3b. This effect was seen with other combinations too and hence was discarded in the subsequent approaches.

We then implemented Alpha dropouts with interaction degree 2, under SGD optimizer, and we see that the highest $F1$ -score of 0.54 is achieved by momentum value 0.95 at dropout probability = 0.6. We see that the general trend across all momentum values is a gradual increase till probabilities 0.6–0.7 and then a sharp decrease till 0.9. Only one momentum value goes against this trend—0.87, which shows a sharp increase from 0.8 and obtains the highest $F1$ -score of 0.508 at dropout probability = 0.9 (compared to other momentum values). This can be observed in Fig. 3c.

For scheduled or curriculum dropouts, the polynomial decay scheduler outperformed the linear one across all optimizers under different hyperparameter tuning and was chosen for subsequent tests. We set the start and end values of probability at 0.9 and 0.1, respectively, and implemented the polynomial curriculum dropout with SGD optimizer for different momentum values ranging from 0.8 to 0.99. The results are shown in Fig. 3d.

There is a gradual increase in the $F1$ -score till momentum value of 0.875 after which there is a sharp decrease till 0.99. The highest $F1$ -score recorded in this approach is 0.54 at a momentum value of 0.875. We implemented standard dropouts

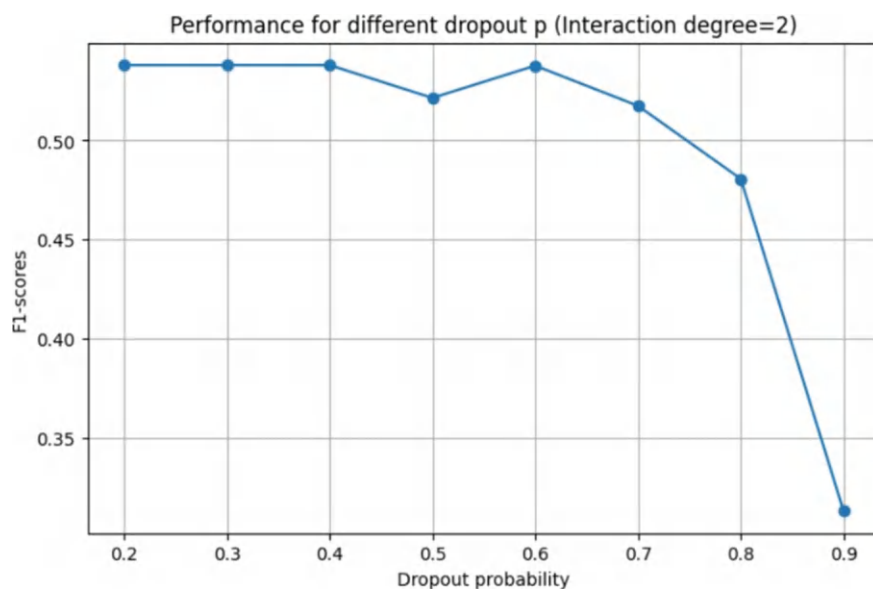


Fig. 4 Graph showing the performance of the LR model on a data model of interaction degree 2, trained with Adam optimizer with standard dropouts

with interaction degree 2, under Adam optimizer and found similar results as that with that of SGD and Scheduled dropout approach. However, here there is no increase in performance. There is a gradual decrease till the dropout probability is 0.5, a slight increase from 0.5 to 0.6 followed by a sharp decrease till 0.9. This effect can be observed in Fig. 4.

The highest recorded $F1$ -score is 0.535 at 0.2, 0.3, 0.4, and 0.6 probabilities. The lowest recorded $F1$ -score is 0.32 at dropout probability = 0.9.

We implemented Alpha dropouts with interaction degree 2, under Adam optimizer and found similar results as those of Adam and standard dropout approach.

There is an initial decrease from a dropout probability of 0.2–0.3 and then an increase from 0.3 to 0.4. This is followed by a gradual decrease in performance till dropout probability = 0.7 and a sharp decrease from 0.7 to 0.8. This is followed by a unique trend—a small increase from 0.8 to 0.9, that was not observed in the previous approach. This effect can be observed in Fig. 5.

The highest recorded $F1$ -score is 0.54 at 0.2 and 0.4 probabilities. The lowest recorded $F1$ -score is 0.41 at dropout probability = 0.8.

Polynomial scheduled/curriculum dropouts with interaction degree 2 under Adam optimizers gave a slightly poorer $F1$ -score of 0.52 as compared to the previous approach. Further analysis was carried out by using Gaussian dropout, along with SGD optimizers for various momenta across 0.87–0.99. The dropout rate varied from 0.6 to 0.9, and the logistic regression was carried out on interaction degrees 1, 2, and 3.

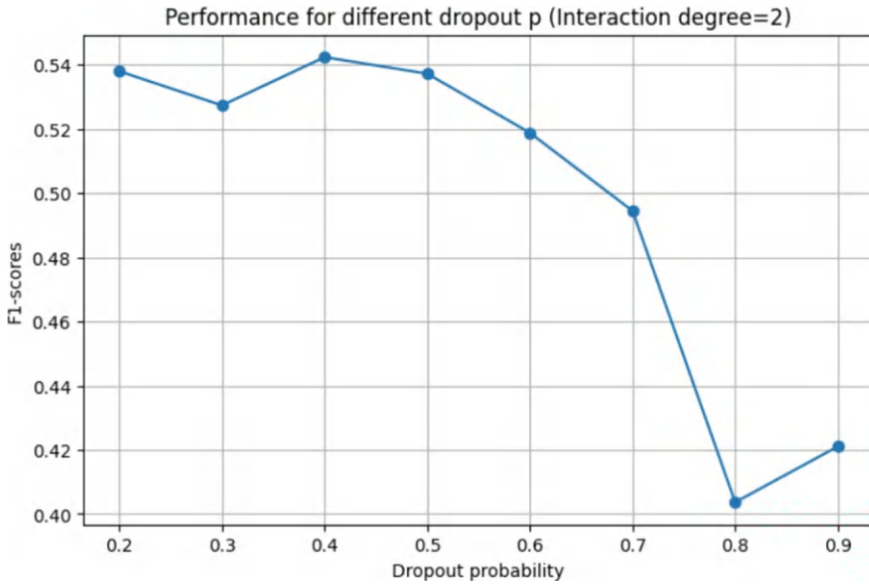


Fig. 5 Graph showing the performance of the LR model on a data model of interaction degree 2, trained with Adam optimizer with alpha dropouts

While using Gaussian dropout with SGD optimizer using momentum = 0.87, we can notice that in general, the interaction degree 2 achieves greater *F1*-score as the dropout rate increases, particularly at $p = 0.7$ and $p = 0.8$. This can be observed from Fig. 6a. However, the same cannot be said for interaction degrees 1 and 3, as both remain almost constant across all dropouts. Interaction degree 1 has a constant *F1*-score of 0.63 across all p values, while interaction degree 2 reaches a maximum of 0.60 at later p values (around $p = 0.9$). Interaction degree 3 achieved an *F1*-score of 0.39 at $p = 0.90$.

We have also experimented the same configuration as done in the previous case, using momentum = 0.92, and as observed from Fig. 6b, the interaction degree 2 follows an upward curve for p greater than 0.7. However, as p increases beyond 0.8, interaction degree 1 notes a decrease in *F1*-score, which was not observed previously. Interaction degree 3 remains the same as before and shows a slight increase toward greater p values, possibly due to memorization of the dataset. Interaction degree 1 sees an initially high *F1*-score of 0.625 as p ranges from 0.60 to 0.80, after which it linearly decreases, and its *F1*-score drops to 0.38. Interaction degree 2 reaches a maximum of 0.649 at greater p values (around $p = 0.9$). Degree 3 achieves an *F1*-score of 0.39 at $p = 0.90$. When the same was done with momentum values across 0.95–0.99, a general trend was observed which suggested that at $p = 0.70$ and $p = 0.80$, interaction degree 2 increased to a local maximum and decreased to a local minimum, respectively, following which it increased steadily until $p = 0.90$.

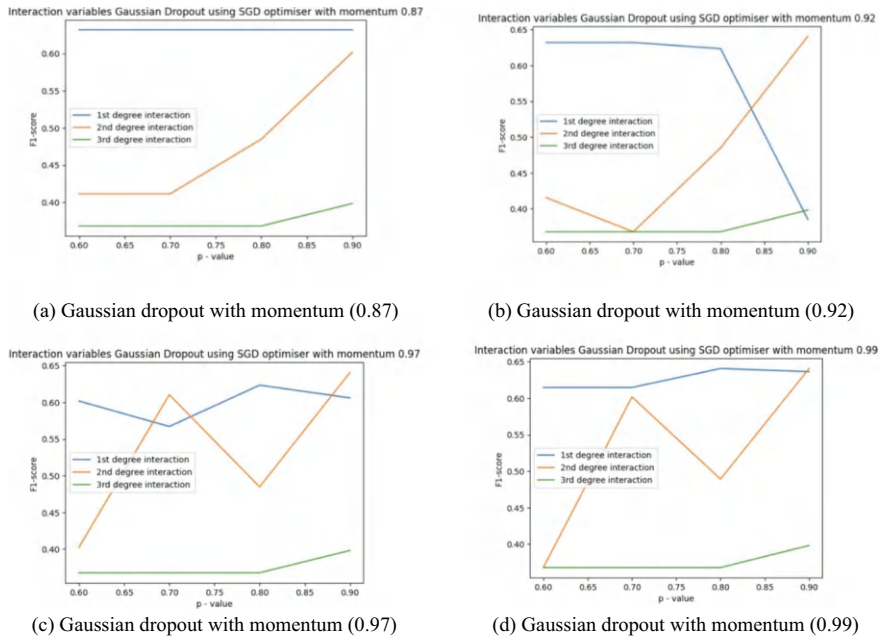


Fig. 6 Graphs showing results of various SGD optimizers with Gaussian dropout on the LR model

Interaction degree 1 followed a similar trajectory, however being less steep in its changes. At $p = 0.90$, it can be observed that the $F1$ -score achieved by 2nd-degree interaction was greater than that achieved by 1st-degree interaction. As the momentum value was increased, there was an increase in the $F1$ -score achieved by interaction degree 1. It rose from $F1 = 0.60$ when its momentum was 0.97 to 0.63 when its momentum was 0.99. These trends are depicted in Figs. 6c and 3d.

We then implemented the same classification task using Gaussian dropout, while varying the dropout rate from 0.60 to 0.90. However, in this case, ADAM optimizer was used. The change in optimizer resulted in a shift in the $F1$ -score trends. These changes can be observed in Fig. 7.

Interaction degree 1 showed a steady rise in its $F1$ -score until $p = 0.90$. Interaction degree 2 observed a global maxima at $p = 0.70$ and experienced a drop until $p = 0.80$, after which it increased by a small rate. An interesting feature about this combination is the improvement in $F1$ -score demonstrated by the 3rd-degree interaction variables, which gradually increased as p varied from $p = 0.80$ to $p = 0.90$. It converged with the same $F1$ -score as the 2nd-degree interaction variable did at $p = 0.90$. Interaction degree 1 achieved its highest $F1$ -score = 0.632 in this case.

Monte Carlo dropout was employed with our model as well, in combination with SGD + momentum optimizer. To begin with, the momentum was set to 0.87, and p was varied from 0.60 to 0.90.

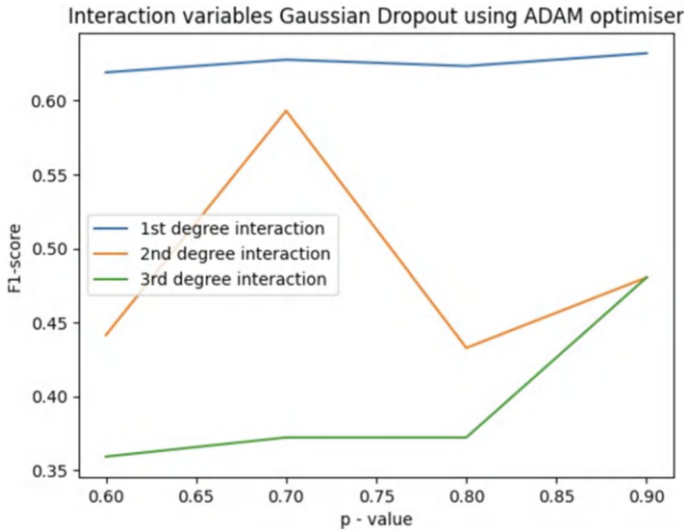


Fig. 7 Graph showing the performance of LR model across degrees 1, 2, and 3, trained with ADAM optimizer and Gaussian dropouts

From Fig. 8, we can note that interaction degree 2 and degree 3 showed similar trends, wherein the $F1$ -scores they produced rose steadily until $p = 0.70$, decreased linearly until $p = 0.80$ and finally increased until $p = 0.90$. Interaction degree 2 seemed to show the highest $F1$ -score at $p = 0.70$. Interaction degree 3 showed an unusual trend in this model—it started from a relatively high $F1$ -score (0.59) and progressively decreased until $p = 0.70$, after which it increased by little and plateaued until $p = 0.90$.

Like in the previous dropout’s case, when Monte Carlo dropout was used with momenta 0.92 and 0.95, similar trends were observed across all the interaction degrees. Figures 9 and 10 depict this performance. From these figures, we can observe that degrees 1 and 2 decreased in their $F1$ -score until a certain p -value and further increased steeply after $p = 0.80$.

The $F1$ -score achieved was greater for degree 1 in both these cases, and it rose to a maximum of 0.601 at $p = 0.90$. The $F1$ -score observed by degree 2 decreased significantly as the momenta were increased. It stooped from an $F1$ -score of 0.59 to 0.485, as the momenta rose from 0.92 to 0.95. Interaction degree 3 remained almost constant as seen in Gaussian dropout.

When momentum was increased to 0.99, a slightly different trend was observed, as seen in Fig. 11. Degree 2 achieved its maximum $F1$ -score of 0.54 at $p = 0.80$ and decreased gradually after that. Degree 1 and degree 3 interaction seemed to show an increase in their $F1$ -score at the very same p -value. Degree 1 achieved an $F1$ -score of 0.595 and degree 3 achieved an $F1$ -score of 0.39, at $p = 0.90$.

We also tried this Monte Carlo dropout along with the ADAM optimizer and varied the dropout rate from 0.60 to 0.90. From Fig. 12, we infer that interaction

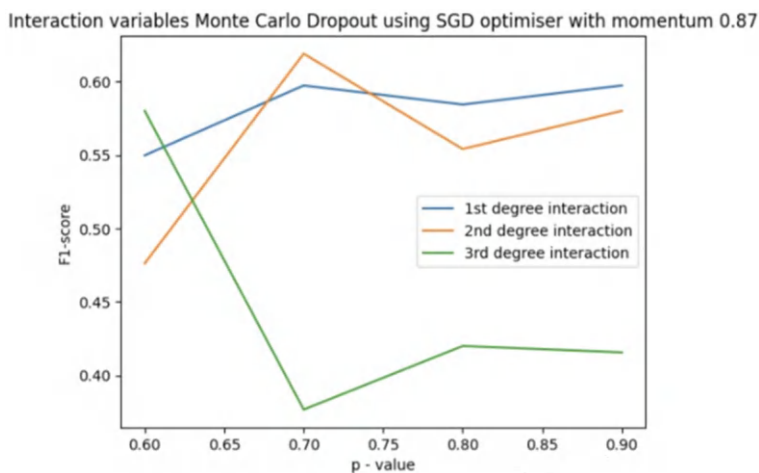


Fig. 8 Graph showing the performance of LR model across degrees 1, 2, and 3, trained with SGD + momentum(0.87) optimizer and MC dropouts

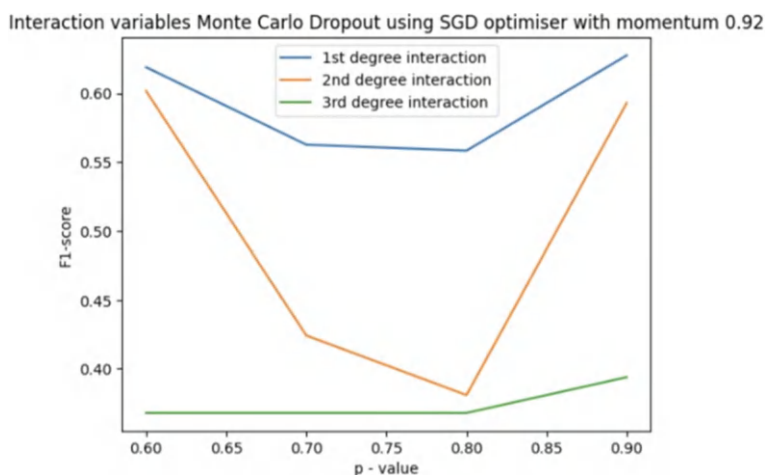


Fig. 9 Graph showing the performance of LR model across degrees 1, 2, and 3, trained with SGD + momentum(0.92) optimizer and MC dropouts

degree 1 showed an initial decrease in its $F1$ -score until $p = 0.70$, after which it demonstrated an increase across all p values till $p = 0.90$, at which its $F1$ -score was 0.540.

Degree 2 mimicked a similar trend; however, its improvement was less steep, and the greatest $F1$ -score it achieved was 0.415. Degree 3's $F1$ -score seemed to increase until $p = 0.70$, decrease at $p = 0.80$, and finally increased to the same $F1$ -score as degree 2 at $p = 0.90$.

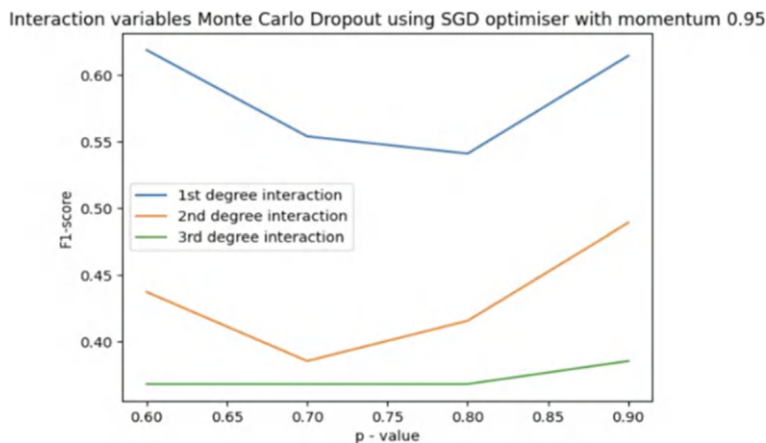


Fig. 10 Graph showing the performance of LR model across degrees 1, 2, and 3, trained with SGD + momentum(0.95) optimizer and MC dropouts

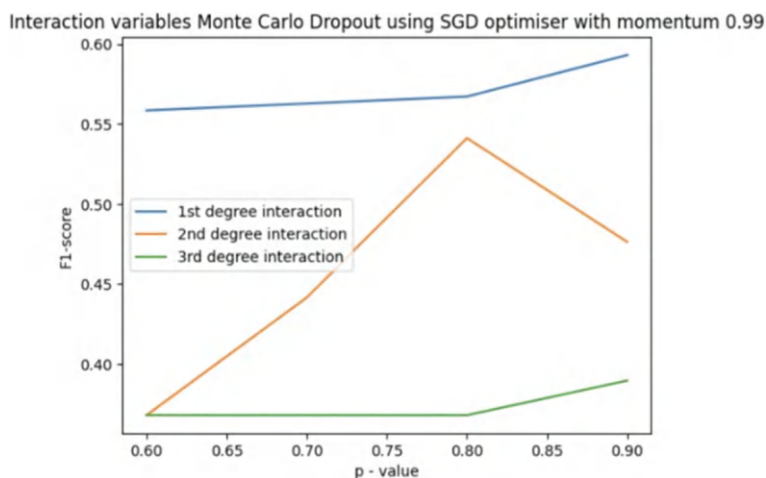


Fig. 11 Graph showing the performance of the LR model across degrees 1, 2, and 3, trained with SGD + momentum(0.99) optimizer and MC dropouts

These combinations demonstrated different trends for each of the interaction variables. However, the models that were trained using Gaussian dropout resulted in greater and more consistent *F1*-scores than their Monte Carlo counterparts.

Further, the combination of Gaussian dropout with ADAM optimizer demonstrated that for degree 1 and degree 3, a greater *F1*-score was achieved, without experiencing any oscillations in their performance. Interaction degree 1 achieved an *F1*-score of 0.632 at $p = 0.90$. Degree 2 achieved a comparable *F1*-score of 0.59 at $p = 0.70$. The *F1*-score of interaction degree 3 rises to a maximum value of 0.47 at $p =$

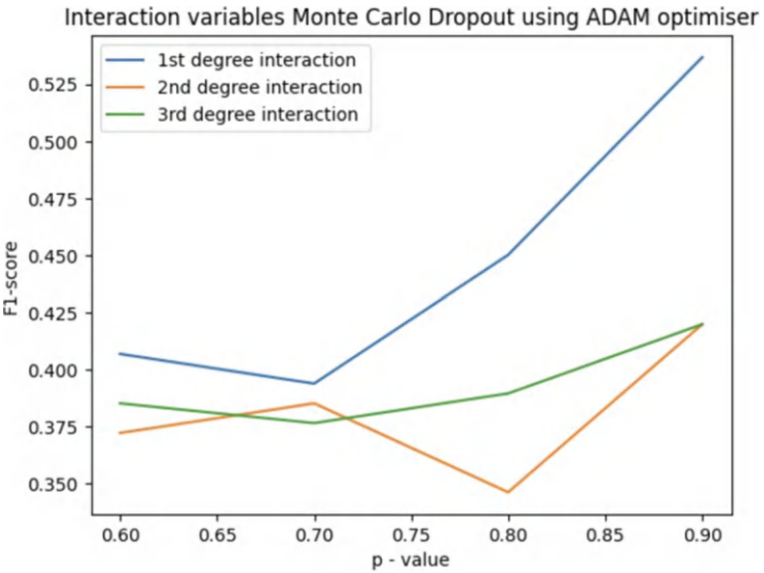


Fig. 12 Graph showing the performance of LR model across degrees 1, 2, and 3, trained with ADAM optimizer and MC dropouts

Table 1 Table summarizing the results of Gaussian dropouts with various optimizers at $p = 0.9$

Optimizers	Interaction degree 1	Interaction degree 2	Interaction degree 3
ADAM	0.632	0.59	0.47
SGD + momentum(0.85)*	0.61	0.645	0.39
SGD + momentum(0.87)	0.63	0.60	0.39
SGD + momentum(0.92)	0.38	0.649	0.39
SGD + momentum(0.95)*	0.525	0.649	0.39
SGD + momentum(0.97)*	0.60	0.64	0.385
SGD + momentum(0.99)*	0.63	0.64	0.385

The optimizers marked * indicate oscillations in $F1$ -score across p values 0.60–0.90, making it less consistent for future use

0.90. It is also observed that the ADAM optimizer seemed to show an upward trend for interaction degrees 1 and 3 across all the dropouts that were implemented along with it, i.e., standard dropout, Gaussian dropout and Monte Carlo dropoutTable 1.

4 Proof and Discussion

The Logistic Regression model can be written as

$$p(y = 1|x) = \sigma(w^T x + b). \quad (2)$$

This equation models the probability of the target variable y being 1 given the input x . The Gaussian dropout procedure can be modeled as

$$w'_i = g_i * w_i. \quad (3)$$

For each weight w_i , g_i is a random variable sampled from a Gaussian distribution, typically $N(1, \alpha)$, where α controls the dropout rate. Using Bayesian Interpretation [5], we see that Gaussian dropout can be seen as placing a Gaussian prior over the weights, suggesting the weights come from a Gaussian distribution. Then we implement Posterior Approximation with Gaussian dropout as,

$$q(w) \propto p(D|w) * p(w). \quad (4)$$

Given data D , Gaussian dropout approximates the true posterior distribution of the weights w .

Using predictive distribution, we have

$$p(y * |x^*, D) = \int p(y * |x^*, w) q(w) dw. \quad (5)$$

For new data x^* , the prediction averages over all possible weight settings, weighted by their posterior probabilities.

We then define the variational objective as,

$$L(q) = E_{q(w)}[\log p(D|w)] KL(q(w) || p(w)). \quad (6)$$

This objective measures the fit between the Gaussian dropout approximation $q(w)$ and the true posterior, balancing data fit and complexity.

When Gaussian dropout is modeled as variational inference, we have to minimize the negative variational objective that is, $\mathcal{L}(q)$.

Gaussian dropout seeks weights that both fit the data well and have high posterior probability. Gaussian dropout acts as a form of probabilistic regularization, penalizing weights that deviate significantly from the prior. Adam, an adaptive learning rate optimization algorithm, computes adaptive learning rates for each parameter.

In the context of logistic regression with Gaussian dropout, we have

$$w^* = \text{Adam}(N_w L(q), w, \text{hyperparameters}). \quad (7)$$

Adam's adaptability makes it particularly effective for optimizing the variational objective in the presence of Gaussian dropout.

Finally, we perform the prediction with Gaussian dropout as,

$$p(y = 1|x) = \sigma(w^{*T} x + b). \quad (8)$$

Using the optimized weights w^* with Gaussian dropout, predictions are made for new data.

5 Conclusion

Gaussian dropout, interpreted as a Gaussian process, introduces a probabilistic form of regularization to the logistic regression model. When combined with the Adam optimizer, which adaptively adjusts learning rates, the model can efficiently navigate the loss landscape. This combination captures complex patterns in the data while preventing overfitting, often leading to improved $F1$ -scores, showcasing the model's enhanced precision and recall capabilities. Therefore, the logistic regression model works best with a combination of Adam optimizer and Gaussian dropouts with interaction degree = 2.

6 Future Work

In our future work, we aim to enhance the interpretability of our logistic regression model with dropout regularization and adaptive optimization. We plan to explore ensemble techniques like bagging and boosting for improved robustness and generalization. Leveraging transfer learning and AutoML approaches will be crucial, especially in scenarios with limited labeled data. Addressing imbalanced datasets will involve exploring oversampling, undersampling, and alternative loss functions. Benchmarking against deep learning models will provide insights into model complexity and performance trade-offs. Real-world application studies will validate the practical impact of our methods, while investigating online learning scenarios will ensure adaptability to evolving patterns. Additionally, integrating explainable AI techniques will contribute to transparency and user trust.

References

1. Klambauer G, Unterthiner T, Mayr A, Hochreiter S (2017) Self-normalizing neural networks. In: Advances in Neural Information Processing Systems, vol. 30
2. Ronald S (2020) Qualitative Analysis of Monte Carlo Dropout
3. Morerio P, Cavazza J, Volpi R, Vidal R, Murino V (2017) Curriculum dropout. In: Proceedings of the IEEE international conference on computer vision, pp 3544–3552
4. Diederik K, Ba J (2014) Adam: a method for stochastic optimization
5. Calvetti D, Somersalo E (2018) Inverse problems: from regularization to Bayesian inference. In: Wiley Interdisciplinary Reviews: Computational Statistics, vol. 10, p. e1427

Crop Yield Prediction in Karnataka Using Machine Learning Techniques



Aman Chhabria and Jayati Bhadra

Abstract The activity of preparing the soil for planting various crops and raising livestock is known as farming and the economy of a complete nation heavily depends on the agriculture. All these while farmers have used traditional farming methods. These methods used to take a lot of time and weren't so accurate which resulted in the decrease of population. With almost 60% of the Indian people employed in this sector, agriculture is one of the largest contributors to India's economy (18% of GDP). The decisions made at each level must be brief due to the numerous aspects that affect this industry. From environmental factors like variability in soil, rainfall, temperatures to man-made inputs like fertilizers, various levels of input, etc., but this paper is looking into temperature and rainfall as a deciding factor. When the environmental factors are not conducive, crop selection using machine learning can be used to minimize the losses. In this paper, we have analyzed the production of crop yields in the state of Karnataka, based on temperature and rainfall as environmental factors. Here, the prediction of when the crop production gives maximum yield according to the season and climate factors.

Keywords Machine learning · Crop yielding · Random forest · Decision tree

1 Introduction

India has continuously been an agricultural country. Its economic system relies upon on agriculture yield boom and agroindustry products. In India, agriculture is basically laid low with rain that may be an exceptionally unpredictable difficulty. Agriculture boom conjointly relies upon on numerous soil parameters, specifically elements like Phosphorus, Potassium, crop rotation, soil, and floor temperature and also on weather

A. Chhabria (✉) · J. Bhadra

Department of Advanced Computing, St Joseph's University, Bengaluru, Bengaluru, India
e-mail: amanchab@gmail.com

J. Bhadra

e-mail: jayatibhadra@sju.edu.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025
N. R. Shetty et al. (eds.), *Advances in Artificial Intelligence and Machine Learning*,
Lecture Notes in Electrical Engineering 1335,
https://doi.org/10.1007/978-981-96-3105-6_30

429

components like temperature, rainfall, and so on. India is gradually progressing closer to technical improvement. Hence, generation can be useful to agriculture that is capable of successively increase crop productiveness leading to better yields for the farmer [1].

The primary goal of agricultural research is to maximize crop yield costs while making use of a certain quantity of to be had land sources. Crop output rates can be raised with the help of a diffusion of system gaining knowledge of procedures. When there is a loss due to negative climate situations, crop choice can be used to minimize the losses and under suitable circumstances, it can be utilized to increase crop yield rates. By optimizing yield rates accordingly based on a particular season and the climate, one could predict when to grow which crop and with this nations' economies are boosted [2].

The best practice of agriculture is developing crops and maintaining farm animals, or farming as it's far extra popularly acknowledged word. It considerably boosts a nation's GDP. Many food items and raw substances are created via agriculture. Industries rent uncooked substances like cotton and jute to create a spread of the various items which can be used on a daily basis. The main contributor toward the production of food is agriculture, in addition to the materials had to make goods on the market. Plants have been grown the usage of conventional agricultural methods. Majority of people of farming international is traditional or traditional. It entails techniques endorsed by means of pro farmers. Those techniques require dedication and take a whole lot of time to pursue due to the fact they are not exact [3].

It is important to make the distinction between historical climatic variability and future climate change. We can think about the climate alternate that passed off within the past as having been absorbed into edition and popular practical and theoretical know-how in agriculture technology. Climate trade in an effort to occur within the destiny, but, can simplest be expected via climate technological know-how models, including research that studies the impact of weather variability of the past and present [4].

India's GDP boom has been one the very best in the beyond decade with an annual increase among 6–7% and agriculture contributing to 18% of its far a huge deal and will continue to be so. India also ranks second worldwide in farm output. It plays a totally great position in the socio-financial improvement in India that is dependent on geographic and economic factors. Some of which are soil, climate, irrigation, temperature, rainfall, etc. With the entire supply chain thriving on this crop prediction and crop analysis could be the main assets for optimizing their production scheduling and avert the risks associated with these. But with decrease in agricultural work force as compared to past decades which leads to higher costs in manual operation machinery can cover this large area and lesser amount of time. With these variables we can look into two aspects of the analysis,

- i. Predicting the crop yield based on collected attributes.
- ii. Grouping district-based prediction on their yield and other factors on similarity basis.

The enhancement of crop control and the making plans of fruit advertising and marketing are both believed to be noticeably depending on yield prediction in precision farming. The farm inputs, such as fertilizers, might be carried out in a selection of approaches depending on the anticipated crop and soil desires after the yield was website-particularly anticipated. To allow yield prediction in agriculture, a number techniques, models, and algorithms have been supplied and put to apply. Primarily based on a small quantity of soil samples, proposed were simple linear correlations between yield and soil characteristics, yet the predictive outcomes exhibited spatial and temporal variability. To assess the connection between various soil properties and grain yield, component analysis was employed. Several different research the usage of complicated linear techniques along with a couple of linear regression analyses showed similar consequences [5].

Within the examination of data, data mining technology is crucial. Data mining is a computational system that uses strategies from the intersection of synthetic intelligence, machine studying, records, and database structures to locate styles in huge information sets [6].

The evaluation and prediction of this statistics plays a critical function and this could be achieved by means of the use of device getting to know strategies. It is able to be performed the use of artificial intelligence which is a broader component of machine learning however, this paper looks into simplest the data analysis and then deriving results using machine learning algorithms. “Unsupervised (clustering) and supervised” are two exclusive gaining knowledge of most commonly used methods. Clustering is the approach in which comparable statistics points are grouped collectively the usage of some distance metrics. The goal is that the distance is minimized within the clusters and is maximized between 2 or more different clusters. This cluster analysis is helpful since it divides the data points into well-defined groups. But due to the lack of data this paper only concentrates on the crop yield prediction. Some of the assumptions that we make in this paper is that the temperature during a season doesn’t change to a great extent across districts and the temperature doesn’t make such an impact like it generally does. This is assumed so as to make the models easier to learn (Fig. 1).

2 Literature Survey

Improving the quality and quantity of crops has become a goal, for the food industry in order to ensure sustainable food production, for a growing population. As a result farmers have started utilizing machine learning (ML) models to manage their farms by making decisions based on data driven insights. In the past, researchers primarily relied on linear iterative multivariate optimization methods and empirical piecewise linear equations to predict crop yield using vegetation indexes and meteorological parameters. However, this paper takes an approach by utilizing regression problems to forecast crop yield. [7].

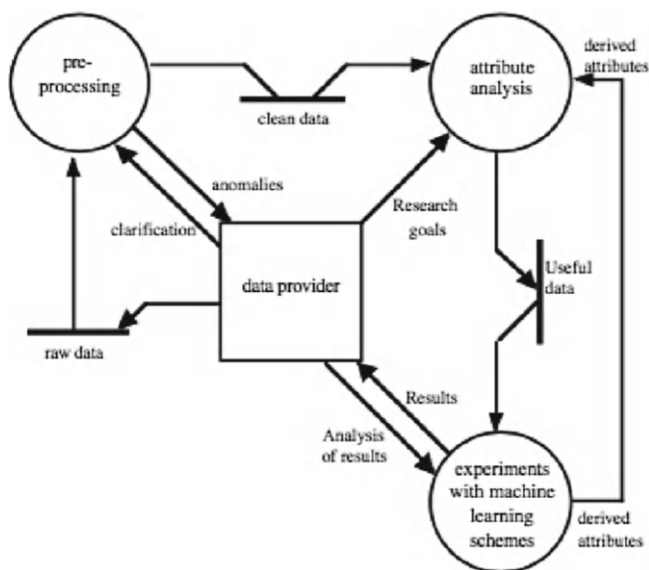


Fig. 1 Process model for a machine learning application

The nutritional composition and presence of minerals, in the soil greatly impact crop growth. This is because the term “soil” encompasses factors such as rainfall, humidity, sunlight, temperature, and pH levels. Durai et al. have suggested a technique, for predicting crops by utilizing a vector system and decision tree algorithm to differentiate crop types based on micronutrients and climatic characteristics [3].

Majumdar et al. [6] proposed a method of unsupervised classification, clustering is used for various uses, a great number of “clustering algorithms have been created. Partitioning clustering, Hierarchical clustering, Density-based methods, Grid-based methods, and Model-based approaches” are the several types of clustering techniques.

Multiple linear regression) is a supervised learning but instead of the labels belonging to a discrete class they belong to the set of real numbers. Supervised algorithms like choice decision tree and random forests were used but no longer for a classification assignment but for a regression venture (random forest regressor) [8], but in this paper we’ve used the supervised algorithms as a classification-based problem. The algorithm was looked at and evaluated the effectiveness of the “random forest (RF) machine learning model”. Its ability to predict crop yield has been compared with Linear Regressions (MLR) which serves as a benchmark.

Precision farming is quite an endeavor to achieve. Numerous research studies have been conducted to predict yields employing “machine learning algorithms (MLA)”. However only a limited number of studies have explored the use of “ensemble machine learning algorithms (EMLA)”.

Prasad et al. [8] has the primary intention to boom the accuracy for the forecasting of the topnotch developments for overcoming starvation demanding situations. They accompanied “stacking regression (SR) and cascading regression (CR)” with a unique aggregate of MLA primarily based on the dataset of wild blueberry.

There are one-of-a-kind forecasting methodologies/techniques developed through researchers everywhere in the world inside the subject of agriculture. the agricultural facts from the years 1965–2009 from the district “East Godavari of Andhra Pradesh”, India become analyzed by means of the researchers Ramesh et al. and was proposed in [9].

MLR is the method used to version the linear relationship between an established target variable and one or many impartial variables. Reason of this conclusions is to find suitable statistics fashions that obtain accurate MSE ratings and an excessive generality in terms of prediction talents. Since clustering of various nations primarily based on their similarity measures became additionally observed which can be used similarly for the districts of every nation, if we accumulated the data, then we’d have got better insights. If rainfall changed into protected it would’ve made the model greater robust alongside temperature, which puts an adverse consequence at the crop manufacturing. After the pre-processing and exploratory facts evaluation, the required and regular attributes with the models were selected. A software device called “Crop consultant” become advanced by way of Veenadhari et al. [10] and proposed that it is a user-excellent internet web page for predicting the impact of those climate parameters on the crop yield.

Numerous algorithms like weighted k-nearest neighbors and Gaussian kernel-based totally help vector machines had been proposed by means of Medar et al. [2]. Experimental outcomes show that the proposed guide vector system-based method performs higher than many different present techniques. From experimental effects, we see that although k-nearest neighbor and Bagged tree display comparative accuracy, assist vector machine shows better accuracy than other strategies used here.

MotiurRahman et al. [11] studied Bangladesh and proposed, there are several different rice kinds available in Bangladesh, each with its own planting season. In preparation for this, previous research of Bangladesh’s climate (temperature, rainfall, and other factors effect) and its impact on rice output in agriculture was conducted. Then, this study was used in a regression analysis that included rainfall and temperature. Crop output suffers as a result of temperature.

Together with all advances within the machines and technologies utilized in farming, useful and accurate facts approximately extraordinary topics also play a substantial function in it. The concept of this paper is to put in force the crop choice technique so that this technique enables in solving many agriculture and farmers troubles. This improves our Indian economy with the aid of maximizing the yield rate of crop manufacturing [2].

Another study focused on improving crop production, in agriculture by utilizing techniques such as “PAM, CLARA DBSCAN” and multiple instances of “Linear Regression Mining.” The researchers analyzed an amount of existing data related to crops, soil and climate well as new non-experimental data. This comprehensive

analysis aims to optimize production and enhance the resilience of agriculture in response, to climate change as discussed in the research conducted by Majumdar et al. [6].

Deep Recurrent Q-Network model that uses a combination of deep learning and reinforcement learning to predict crop yield based on environment, soil, and crop parameters were also proposed by Elavarasan et al. [12].

Qiao et al. [13] predicted crop yield based on a novel 3-D convolutional neural multi-kernal network based on remote sensing data. The proposed model captures hierarchical features of constructing a full 3-D convolutional neural network to explore more regarding the data.

Ensemble methods were also proposed by Keerthana et al. [14] they used 2 machine learning algorithms i.e., Decision Tree Regressor and AdaBoost Regressor to predict the crop yields based on various climate factors.

3 Methodology

3.1 Data Collection

The main goal of this research proposal is to examine agricultural data for the state of Karnataka (District Wise) using the appropriate techniques for prediction. The data regarding this study is collected from the following sources:

Crop Yield Dataset for each state (district wise): "<https://desagri.gov.in/>".

Data related to temperature, humidity, weather: "<https://www.visualcrossing.com/weather/weather-data-services#>" [15]

Crop Yield dataset consists of data from the years ranging from (2010–2020) with following parameters namely: year, states, district, crop, season (kharif, rabi, summer), area, production, yield, tempmin (Minimum Temperature), tempmax (Maximum Temperature), temp, dew, humidity, and precip (Precipitation). Further analysis could be done if we had extra variables and make the model more robust and accurate in the prediction (Fig. 2).

3.2 Model Proposed

“Decision Tree.”

A decision tree represents a tree-like structure that is similar to a flowchart, the internal node represents a test on any attribute selected by some criteria and every internal node represents an outcome [16–20]. Finally, each leaf node (terminal node) is the class label. Decision trees make predictions on the training data by passing data downwards from root nodes to leaves. This split occurs repeatedly according to the target/predictor variables so as to make the child nodes pure (homogenous). The

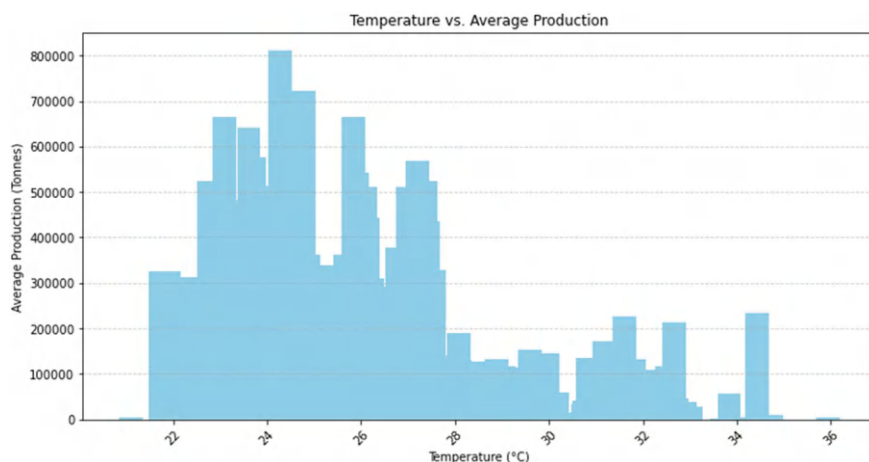


Fig. 2 Plot of temperature versus production

root node begins with the entire set of the training data. The root split is done by one of the predictor variables. A decision tree is a type of machine learning model that is trained to associate data with outcomes. During the training phase, the model is provided with data and the corresponding desired outcomes for learning purposes. It acquires knowledge about any connections or patterns, between the input data and the target variable. The tree's accuracy is greatly affected by the strategic splits it takes. Decision criteria are different for a regression and a classification task. "Mean Square Error (MSE)" is used to decide the split of a node into two or more. In this paper, we have used decision tree as a classification problem in order to perform crop classification. The key concept used was splitting the data into training and testing based on various concepts like Gini impurity and Information Gain and decision tree is used to find the best splits for the crop classifications in each and every district. Prevention of overfitting is mainly required for decision tree algorithms. The accuracy and classification reports are used to evaluate the model (Fig. 3).

One of the ensemble methods in machine learning is random forests, capable of performing both "regression and classification tasks [15, 22–25,]". It utilizes a combination of decision trees. Employs a statistical approach known as bagging. Instead of just averaging out the predictions of the decision trees a random forest uses samples that are randomly sampled while building trees and uses a random subset of features for splitting nodes. To improve the precision and consistency of its predictions the system constructs trees. Combines their predictions. The random forest learns, from selected training observations, with replacement, a process known as bootstrapping. Consequently, each individual tree may exhibit variability based on its set of training data but overall, this variance is reduced across the entire forest.

But why is a random forest better than a single decision tree? The basic idea behind using random forests on decision trees is to combine the predictions made by multiple decision trees and encapsulate them into a single model. This is done for

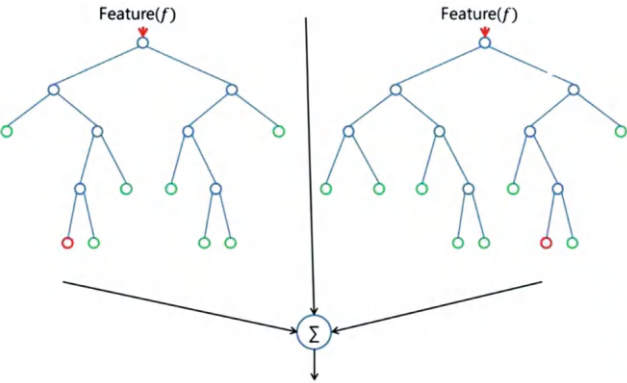
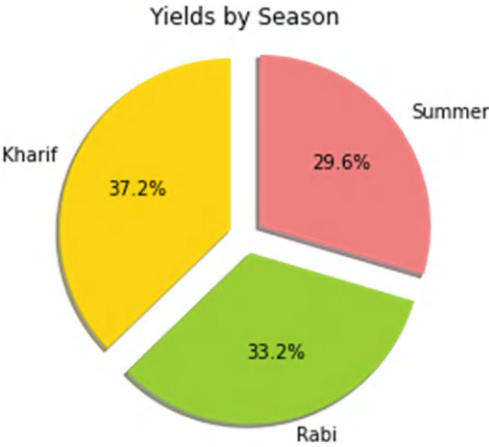


Fig. 3 Decision tree [21]

a specific reason: the individual predictions made by each decision tree may not be accurate, but when combined, the average predictions will be close to the true value. Since a set of random features and a set of random training data points (“Random Forest”) are considered, each tree captures different sources of information instead of a limited range of information. The limit brought by a tree is shown. If each tree uses the same data set (same set of data points), then each tree will be largely affected by the same outliers. The diversity provided by using multiple trees makes the prediction of the model more robust [26–30].

Overfitting is a problem with decision trees because they might keep expanding until there is only one leaf node for each and every observation. The decision tree’s predictions are susceptible to even minor changes in the original parameters. And to get around this, we use bagging, which increases bias while decreasing variance (Figs. 4, 5, 6, and 7).

Fig. 4 Plot of overall crop yield according to seasons (3 main seasons included)



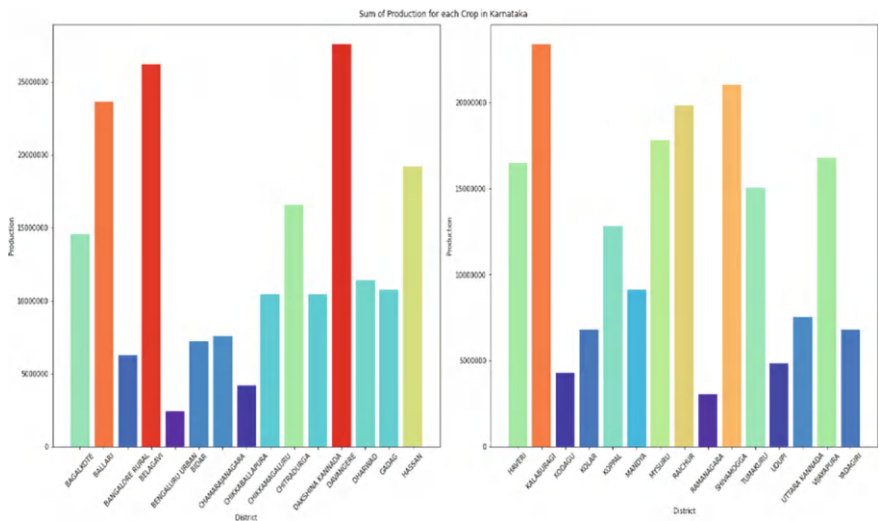


Fig. 5 Plot for Karnataka districts versus total production over the years

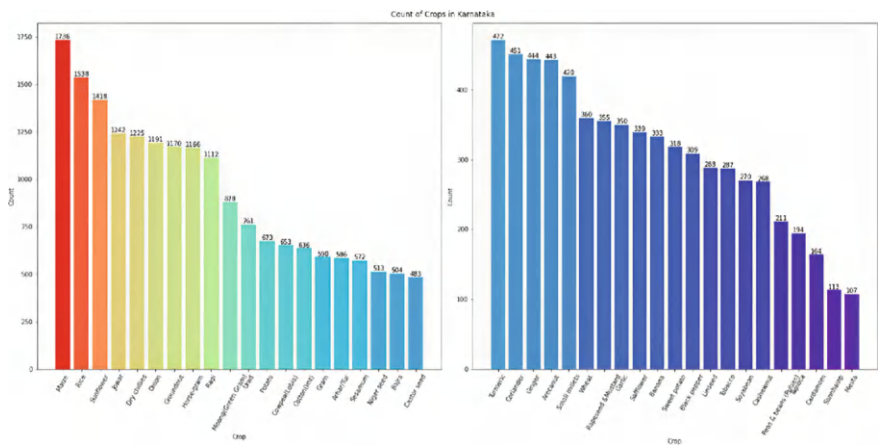


Fig. 6 Plot for crop versus count over the years

3.3 Experimented Results

Modified Approach of Decision Tree

As we know, decision tree is mainly used for classification problems. But in our case, we are predicting the crop yield and we are predicting it district wise for each and every crop, thus we used a classification problem to classify each and every crop according to the district. Since we are predicting for every crop according to the

district, we went ahead with a classification problem to predict the crop yield based on features like area, temperature, due, humidity, precipitation, etc. Here, Classification and Regression Tree (CART) algorithm was used for predicting.

CART uses some of metrics for choosing best nodes which are:

1. “Gini Index”: it’s the measure of impurity in building the structure of the trees.
2. “Information Gain”: it’s a decrease in entropy value after data is split on the basis of attribute.
3. “Entropy”: it’s a metric used to find impurities.

In machine learning, we hold out 80% of the dataset for training and 20% for testing. This 80% was trained using a decision tree classifier and the hyper parameters are chosen according to the validation test.

3.4 Accuracy

Algorithms	Accuracy (%)
1. Random forest classifier	96
2. Decision tree Classifier	93

The test data for decision tree scored an accuracy of 93% for the Karnataka Data and to improve the accuracy and making the model more robust we went ahead with random forest which gave an accuracy of 96% for the Karnataka data (All districts Included). Accuracy is measured through the accuracy_score function from the scikit-learn library, it compares the y_test with the y_pred to give the accuracy of the model and the report is generated with the help of classification_report function.

4 Conclusion

Out of the many machine learning techniques that were implemented on the input data to predict the best methods in terms of performance, we used “SVM, XGBoost, decision trees and random forests”. Using “decision trees and random forests”, we could find out the most important features to achieve better predictions of yield, while SVM and XGBoost didn’t perform that well since there are many features that are random in nature and we weren’t getting our desired accuracy. We used classification algorithms so all the models were compared with their accuracy score and made note of which model was performing well and which were not performing so well with our dataset. Analysis on the data showed that random forests worked best for this input data while decision trees weren’t that back in terms of the accuracy, random forest scored 96% and for decision tree we got 93% for the same. This study can be

extended and would probably give better results if many other factors would've been included and modeled for. The only problem with this study is that many factors that are included in this study are random; and historical data which might not be so accurate as well therefore modeling for these is a herculean task.

References

1. Crop yield prediction and alternate crop prediction using ML
2. Medar R, Rajpurohit V, Shweta S (2019) Crop yield prediction using machine learning techniques, pp 1–5. <https://doi.org/10.1109/I2CT45611.2019.9033611>
3. Durai SKS, Shamili MD (2022) Smart farming using machine learning and deep learning techniques. *Decis Anal J* 3:100041. ISSN 2772-6622. <https://doi.org/10.1016/j.dajour.2022.100041>
4. Murari K, Mahato S, Jayaraman T, Swaminathan M (2019) Extreme temperatures and crop yields in Karnataka, India. *Rev Agrar Stud* 8(2):92–114
5. Pantazi XE, Moshou D, Alexandridis T, Mouazen AM (2016) Wheat yield prediction using machine learning and advanced sensing techniques. *Comput Electron Agric* 121:57–65
6. Majumdar, J, Naraseeyappa S, Ankalaki S (2017) Analysis of agriculture data using data mining techniques: application of big data. *J Big Data* 4. <https://doi.org/10.1186/s40537-017-0077-4>
7. Choudhary N, Chukkapalli S, Mittal S, Gupta M, Abdelsalam M, Joshi A (2020) Yieldpredict: a crop yield prediction framework for smart farms, pp 2340–2349. <https://doi.org/10.1109/BigData50022.2020.9377832>
8. Prasad A, Singh R, Tare V, Kafatos M (2007) Use of vegetation index and meteorological parameters for the prediction of crop yield in India. *Int J Remote Sens* 28:5207–5235. <https://doi.org/10.1080/01431160601105843>
9. Ramesh D, Vishnu Vardhan B (2013) Data mining techniques and applications to agricultural yield data. *Int J Adv Res Comput Commun Eng* 2(9)
10. Veenadhari S, Misra B, Singh CD (2011) Data mining techniques for predicting crop productivity—a review article. *Int J Comput Sci Trends Technol* 2(1)
11. MotiurRahman M, Haq N, Rahman RM (2014) Application of data mining tools for rice yield prediction on clustered regions of Bangladesh. *IEEE* 2014:8–13
12. Elavarasan D, Vincent PMD (2020) Crop yield prediction using deep reinforcement learning model for sustainable agrarian applications. *IEEE Access* 8:86886–86901. <https://doi.org/10.1109/ACCESS.2020.2992480>
13. Qiao M et al (2021) Exploiting hierarchical features for crop yield prediction based on 3-D convolutional neural networks and multikernel Gaussian process. *IEEE J Sel Top Appl Earth Observations Remote Sens* 14:4476–4489. <https://doi.org/10.1109/JSTARS.2021.3073149>
14. Keerthana M, Meghana K, Pravalika S, Modepalli K (2021) An ensemble algorithm for crop yield prediction, pp 963–970. <https://doi.org/10.1109/ICICV50876.2021.9388479>
15. Seireg HR, Omar YMK, El-Samie FEA, El-Fishawy AS, Elmahalawy A (2022) Ensemble machine learning techniques using computer simulation data for wild blueberry yield prediction. *IEEE Access* 10:64671–64687. <https://doi.org/10.1109/ACCESS.2022.3181970>
16. Gleaso CP (1982) Large area yield estimation/forecasting using plant process models. paper presentation at the winter meeting American society of agricultural engineers palmer house. Chicago, Illinois, pp 14–17
17. Majumdar J, Ankalaki S (2016) Comparison of clustering algorithms using quality metrics with invariant features extracted from plant leaves. In: Paper presented at international conference on computational science and engineering
18. Jain A, Murty MN, Flynn PJ (1999) Data clustering: a review. *ACM Comput Surv* 31(3):264–323

19. Berkhin P (2006) A survey of clustering data mining technique. In: Kogan J, Nicholas C, Teboule M (eds) Grouping multidimensional data. Springer, Berlin, pp 25–72
20. Han J, Kamber M (2001) Data mining: concepts and techniques. Morgan Kaufmann Publishers, Massachusetts
21. Jhajharia K, Mathur P, Jain S, Nijhawan S (2023) Crop yield prediction using machine learning and deep learning techniques. *Procedia Comput Sci* 218:406–417. ISSN 1877-0509. <https://doi.org/10.1016/j.procs.2023.01.023>
22. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Paper presented at international conference on knowledge discovery and data mining
23. Alberto GS, Juan FS, Ojeda-Bustamante W (2014) Predictive ability of machine learning methods for massive crop yield prediction. *Span J Agric Res* 12(2):313–28
24. Veenadhari S, Misra B, Singh D (2014) Machine learning approach for forecasting crop yield based on climatic parameters. In: Paper presented at international conference on computer communication and informatics (ICCCI-2014), Coimbatore
25. Multiple linear regression (2017) <http://www.originlab.com/doc/Origin-Help/Multi-Regression-Algorithm>. Accessed 3 July 2017
26. Sahu S, Daniya T, Cristin R (2023) A study on weather based crop prediction system using big data analytics and machine learning. In: 2023 5th international conference on smart systems and inventive technology (ICSSIT), Tirunelveli, India, pp 794–800. <https://doi.org/10.1109/ICSSIT55814.2023.10060865>
27. Devan KPK, Swetha B, Uma Sruthi P, Varshini S (2023) Crop yield prediction and fertilizer recommendation system using hybrid machine learning algorithms. In: 2023 IEEE 12th international conference on communication systems and network technologies (CSNT), Bhopal, India, pp 171–175. <https://doi.org/10.1109/CSNT57126.2023.10134662>
28. Maheswari MU, Ramani R (2023) A comparative study of agricultural crop yield prediction using machine learning techniques. In: 2023 9th international conference on advanced computing and communication systems (ICACCS), Coimbatore, India, pp 1428–1433. <https://doi.org/10.1109/ICACCS57279.2023.10112854>
29. Lakshmanarao A, Kumar MN, Ratnakar KSV, Satwika Y (2023) Crop yield prediction using regression models in machine learning. In: 2023 2nd international conference on applied artificial intelligence and computing (ICAAIC), Salem, India, pp 423–426. <https://doi.org/10.1109/ICAAIC56838.2023.10141462>
30. https://web.archive.org/web/20181229092825if_/https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd
31. Al-Mulla Y et al (2022) Assessment of urban expansion's impact on changes in vegetation patterns in Dhofar, Oman, using remote sensing and GIS techniques. *IEEE Access* 10:86782–86792. <https://doi.org/10.1109/ACCESS.2022.3198942>

Autism Spectrum Disorder (ASD) Prediction Using EfficientNet-B7 and Haar Cascade



Anson J. Maliackal, Ancyline J. Maliakkal, and Jayati Bhadra 

Abstract Autism Spectrum Disorder (ASD) is a neurodevelopmental condition affecting early childhood and impacting diverse aspects of a child's development. In the absence of medical tests, the observation of behavioral patterns remains the primary diagnostic approach for ASD. Recently, neural network-based models have emerged for early ASD identification. In this paper, we present an ASD prediction model utilizing Convolutional Neural Network architectures, specifically EfficientNet-B7, in combination with Haar Cascade. This approach offers a significant improvement in the accuracy and prediction of ASD in children. This model is integrated into a web application for image-based ASD prediction using Dash module from Python. This application helps healthcare professionals with a practical tool for early ASD identification.

Keywords Machine learning • Autism spectrum disorder • Efficientnet • HaarCascade • Deep learning • CNN

1 Introduction

Autism Spectrum Disorder (ASD) is a multifaceted neurodevelopmental condition that impacts both communication and behavior. It exhibits a broad spectrum of symptoms, and individuals with ASD may encounter challenges related to social interaction, communication, and repetitive behaviors. As per the recent review by

A. J. Maliackal (✉)

Big Data Analytics), Department of Advanced Computing, St. Joseph's University, Bengaluru, Bengaluru, India

e-mail: ansonjmaliackal@gmail.com

A. J. Maliakkal

Data Engineer, Biofin Capital LLC, Princeton, UK

J. Bhadra

Department of Advanced Computing, St. Joseph's University, Bengaluru, Bengaluru, India

e-mail: jayatibhadra@sju.edu.in

J. Zeidan et al. [1], 1 in 100 children worldwide is affected by autism. Early diagnosis of ASD is essential for early intervention. Intervening at an early stage can enable individuals with ASD to realize their maximum potential and enhance their quality of life. According to the American Psychiatric Association, the symptoms of ASD typically become apparent during early childhood development and encompass restricted interests, social deficits, communication delays, challenges in expressing and recognizing emotions, avoidance of eye contact, and the presence of the idiosyncratic hand, head, and body movement [2]. These hallmark features often coexist with repetitive behaviors and obsessive interests, adding layers of complexity to the condition.

Early diagnosis plays a pivotal role in improving the quality of life for individuals with Autism Spectrum Disorder (ASD). Unfortunately, conventional diagnostic methods, including the Autism Diagnostic Interview Revised (ADI-R), Autism Diagnostic Observation Schedule (ADOS) [3], Childhood Autism Rating Scale (CARS) [4], Gilliam Autism Rating Scale (GARS) [5], Screening Tool for Autism in Toddlers and Young Children (STAT) [6], and Autism Spectrum Quotient (AQ) [7], have long been the mainstays of ASD assessment. These methods often rely on time-consuming, resource-intensive, and subjective clinical evaluations, leading to lengthy waiting periods and hindering early intervention initiatives. With the growing demand for accurate and timely diagnosis, it has become imperative to explore alternative approaches that leverage contemporary technologies and scientific advancements.

Recent advances in machine learning (ML) have made it possible to develop computer-aided diagnostic (CAD) systems for ASD. These systems can be trained on large datasets of images and clinical data to identify patterns associated with ASD. This has the potential to expedite the ASD classification process and improve access to early diagnosis and intervention. ML-based CAD systems for ASD have been shown to achieve promising results in recent studies. Deep learning models have been shown to be effective in a variety of image classification tasks, including facial recognition and medical image analysis. This says that deep learning could be used to develop accurate and efficient CAD systems for ASD diagnosis. There are several potential benefits in using deep learning for ASD diagnosis. First, deep learning models can be trained on large datasets of images and clinical data, which can help to improve the accuracy and robustness of the models. Second, deep learning models can be used to extract subtle features from images that may be difficult for humans to detect and extract. Third, deep learning models can be used to develop CAD systems that are easy to use and accessible to healthcare workers.

In this research paper, the system utilizes an EfficientNet-B7 model of EfficientNet family, which is a family of CNNs that are known for their computational efficiency and low error rates. In addition, we propose a novel approach using Haar Cascade algorithm for extracting specific facial features such as the eyes, nose, and mouth for the model training.

2 Literature Review

Autism Spectrum Disorder (ASD) is a developmental disorder that may affect communication and behavior. Autism diagnosed individuals will have severe behavioral changes and social interaction difficulties. Early diagnosis of ASD can help individuals with ASD to improve their quality of life. However, conventional diagnostic methods, such as the Autism Diagnostic Interview Revised (ADI-R) and the Autism Diagnostic Observation Schedule (ADOS) are not up to the date. Several studies in recent years have emphasized the importance of utilizing advanced technologies, such as machine learning and deep learning in the prediction and diagnosis of ASD.

In recent years, the integration of machine learning (ML) techniques into ASD diagnosis, particularly the utilization of image datasets, has emerged as a promising avenue for progress [8]. Notably, numerous researchers in the ASD community have investigated the potential of ML to expedite the ASD classification process and identify key factors influencing diagnostic outcomes [9–16].

Convolutional Neural Network (CNN) models have achieved remarkable success in various classification tasks, particularly in image classification. Their ability to provide high accuracy, extract meaningful features, and allow for hyperparameter tuning has made CNNs a prominent choice for image-based models. In the context of early ASD detection, studies using CNN architectures such as VGG [17] and Inception [18] have achieved accuracy rates of 85% and 78%, respectively.

EfficientNet, a family of CNNs introduced by M. Tan et al. [19], has also gained popularity in ASD classification models for both adults and children. This network framework is favored for its computational efficiency and low error rates [20]. It's worth noting that most of these models traditionally treat the entire image as a whole. However, K. Aldridge et al. has shown that autistic children often exhibit distinct facial features, including wide eyes, a short nose, and a wide mouth [22].

In our approach, we utilize Haar Cascade classifiers along with EfficientNet-B7 to extract specific facial features such as eyes, nose, and mouth, placing particular emphasis on these regions during model training.

3 Proposed Methodologies

3.1 Dataset

We acquired a dataset consisting of 5118 images of children categorized into autistic and non-autistic groups from a publicly available Kaggle source [21]. To facilitate our research, the dataset was partitioned into three distinct categories: the training set (4421 images), the validation set (259 images), and the test set (438 images). In preparation for model training, we employed the Image Data Generator from

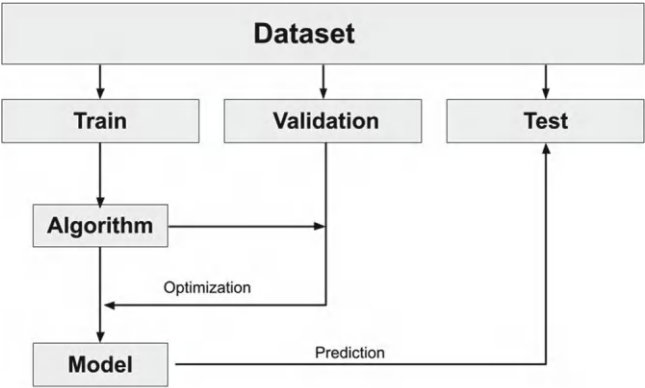


Fig. 1 Represents how the dataset is trained using the model

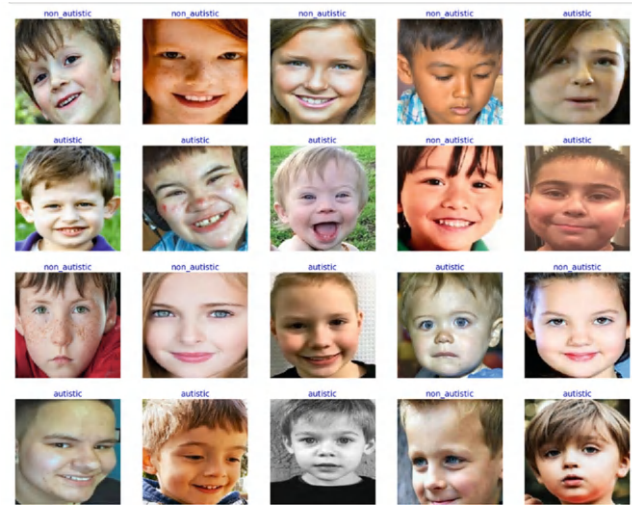


Fig. 2 Dataset sample

Keras for image preprocessing. This involved shuffling the images, resizing them to a uniform 200×200 pixels, and retaining the “RGB” color mode (Figs. 1 and 2).

3.2 Facial Feature Emphasis

A distinctive feature of our approach is the focus on specific facial features extracted from the images. To accomplish this, we harnessed the Haar Cascade algorithm,

acknowledged for its robust and efficient facial feature detection capabilities. Pre-trained XML files were employed to identify and extract crucial facial components, such as eyes, nose, and mouth. Notably, these features exhibit distinct characteristics in autistic children as highlighted by K Aldridge et al. in 2011 [22]. These extracted features were designated as regions of interest during the model training process.

3.3 Model Architecture

Our approach centers on the development of an autism detection model utilizing the state-of-the-art Convolutional Neural Network (CNN) architecture known as EfficientNet-B7. CNN models are highly adept at image classification across diverse datasets. Within the architecture, there are three fundamental layers. The convolutional layer extracts features from the input image, which are subsequently dimensionally reduced by the pooling layer to lower computational costs. The fully connected layer flattens the feature map and produces a probability vector.

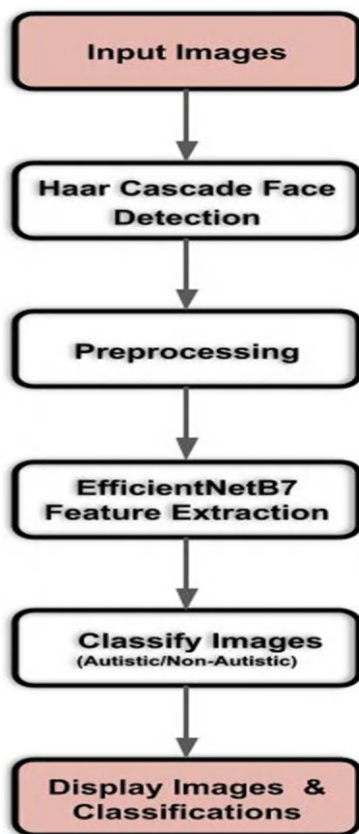
EfficientNet models employ a combination of depthwise convolutions and point-wise convolutions, also known as separable convolutions. These convolution types reduce computational load while sustaining high performance. The systematic approach of scaling up dimensions, depth, width, or resolution, referred to as compound scaling, demonstrated by M. Tan et al. led to the development of EfficientNet [19].

EfficientNet-B7, the model we utilized, was pre-trained on the ImageNet dataset, a comprehensive resource for image classification tasks. We incorporated a global max-pooling layer for feature aggregation, reducing the feature map to a single vector. Batch normalization was applied to normalize network activations, enhancing training stability. Regularization techniques, including L1 and L2 regularization, were employed in the dense layers to prevent overfitting. The model concludes with an output layer featuring a sigmoid activation function and a dense layer with a single unit, typically used for binary classification tasks. The model's initialization involved pre-trained weights from the ImageNet dataset, a pivotal step in transfer learning, which equips the model with valuable features learned from extensive data (Fig. 3).

4 Results

4.1 Model Training and Validation

The ASD classification model underwent training using the designated training dataset and was subsequently validated with the validation dataset. Training extended over 20 epochs with a batch size of 20. The results exhibited a significant accuracy

Fig. 3 Model architecture

of 85% for the validation data, while the training data achieved an accuracy of 99%, as illustrated in Fig. 4.

4.2 Sensitivity, Specificity, and Confusion Matrix

Sensitivity of the model shows how accurately the models predicts true positive and is calculated using (1),

$$\text{Sensitivity} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \times 100\% \quad (1)$$

Our model predicts “Autistic” with an accuracy of 88.6%.

Specificity of the model shows how accurately the models predicts true negative and is calculated using (2),

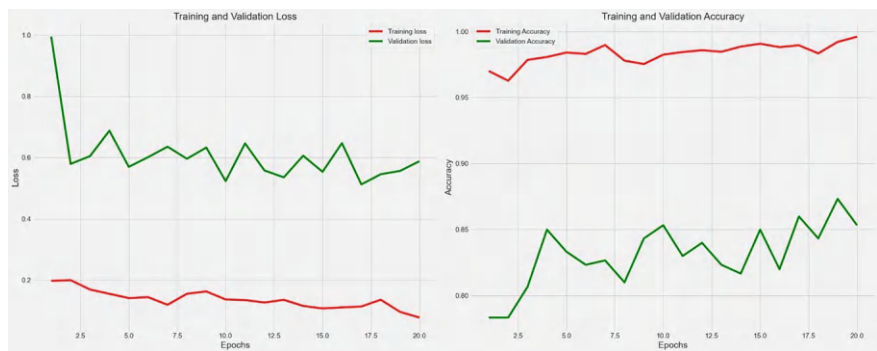


Fig. 4 Graph showing accuracy and loss for train and validation datasets

$$\text{Specificity} = \frac{\text{True negatives}}{\text{True negatives} + \text{False negatives}} \times 100\% \tag{2}$$

Our model predicts “Non-Autistic” with an accuracy of 87.4%. Confusion matrix depicts the actual number of true positives, false positives, true negatives, and false negatives (Fig. 5).

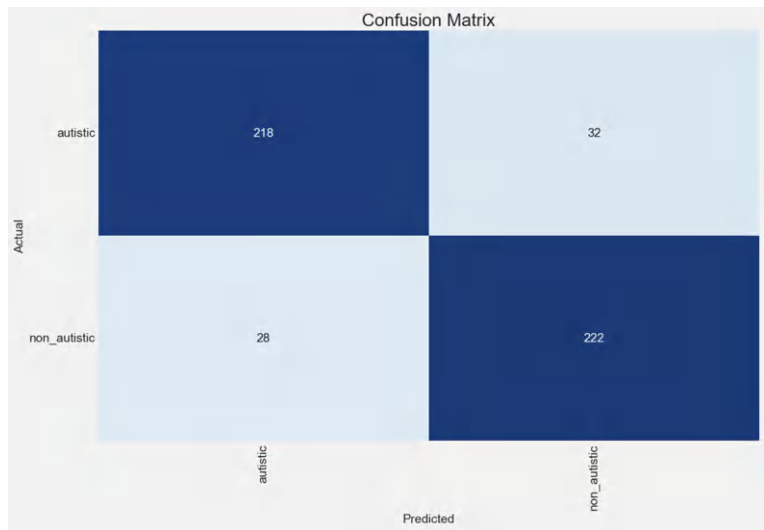


Fig. 5 Confusion matrix

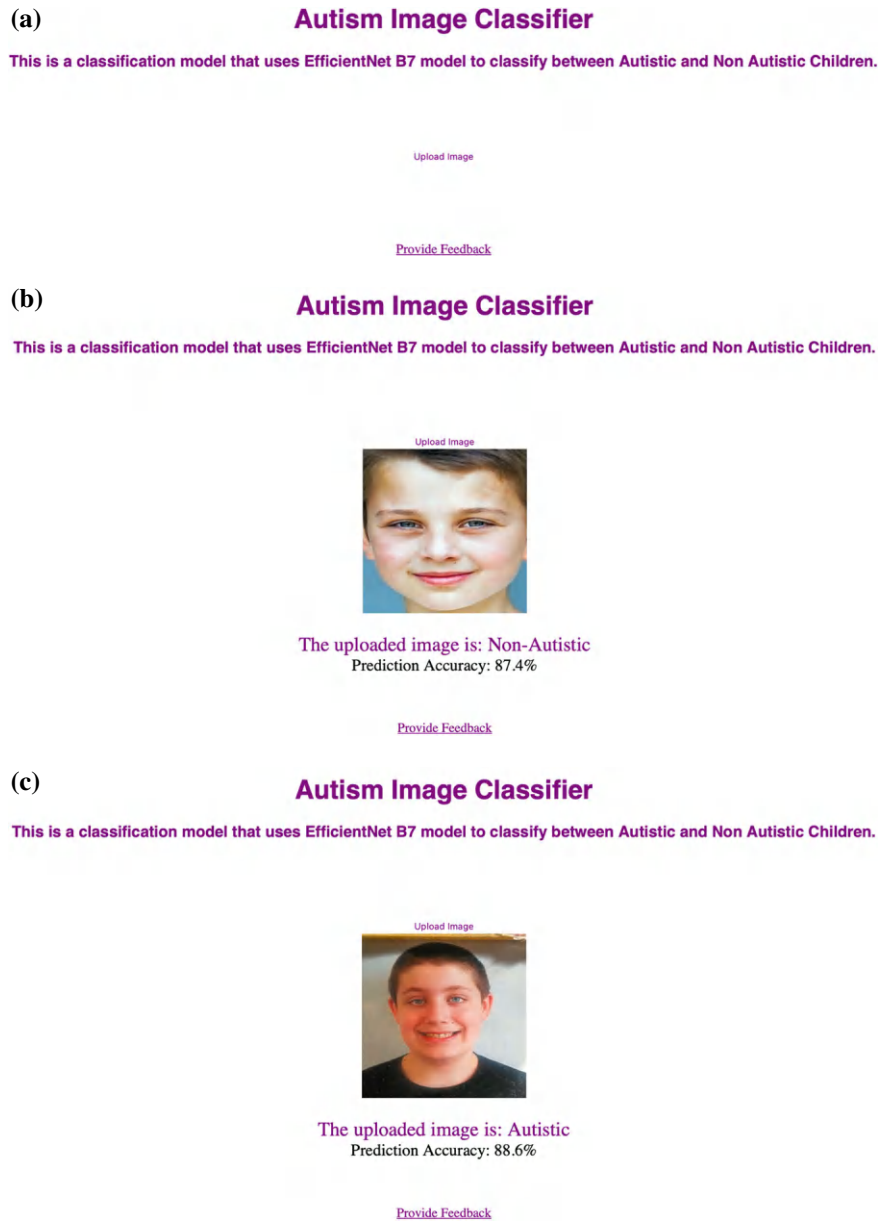
4.3 Web Application Development

In addition to model development, a web application is developed. This web application was designed and developed using Dash module in python. It incorporates the trained model in the backend to classify any uploaded image of a child as either autistic or non-autistic with a high level of accuracy. It's user interface is easy and accessible to healthcare professionals and individuals interested in early ASD detection. Users can upload an image of a child, and the application will process the image through our trained model and display the result.

Snapshots of the application before and after uploading the image taken from autistic and non-autistic children are shown in Fig. 6a, b and c, respectively, and Fig. 6d shows the snapshot with autism from outside the dataset. These snapshots illustrate the process of testing the application's ability to provide accurate result.

5 Conclusion

This study utilized Convolutional Neural Network architectures, specifically EfficientNet-B7, and augmenting their capabilities with the Haar Cascade algorithm for precise facial feature extraction. Our proposed model demonstrates a significant achievement in accuracy for predicting ASD in children. The Haar Cascade algorithm enhances the model's sensitivity and specificity which are crucial factors in reliable diagnostic systems. The achieved accuracy of 85% for the validation dataset underscores the effectiveness of our approach. Moreover, the development of a user-friendly web application, incorporating the trained model, extends the practical utility of our system as an accessible tool for early ASD detection. Our research contributes to the ongoing efforts to expedite ASD diagnosis, paving the way for timely interventions and improved outcomes for individuals affected by this neurodevelopmental condition.



(d)

Autism Image Classifier

This is a classification model that uses EfficientNet B7 model to classify between Autistic and Non Autistic Children.

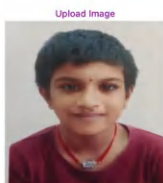


Fig. 6 (continued)

References

1. Zeidan J et al (2022) Global prevalence of autism: a systematic review update. *Autism Res* 15(5):778–790
2. American Psychiatric Association (2013) Diagnostic and statistical manual of mental disorders (DSM-5-TR), vol 15(5), pp 778–790
3. Lord C et al (2000) The autism diagnostic observation schedule-generic: a standard measure of social and communication deficits associated with the spectrum of autism. *J Autism Dev Disord* 30(3):205–223
4. Van Bourgondien ME et al (1992) Comparison of DSM-III-R and childhood autism rating scale diagnoses of autism. *J Autism Dev Disord* 22(4):493–506
5. South M et al (2002) Utility of the Gilliam autism rating scale in research and clinical populations. *J Autism Dev Disord* 32(6):593–599
6. Stone WL et al (2000) Brief report: screening tool for autism in two-year-olds (STAT): development and preliminary data. *J Autism Dev Disord* 30(6):607–612
7. Baron-Cohen S et al (2001) The autism-spectrum quotient (AQ): evidence from Asperger syndrome/high-functioning autism, males and females, scientists and mathematicians. *J Autism Dev Disord* 31(1):5–17
8. Omar KS et al (2019) A machine learning approach to predict autism spectrum disorder. *IEEE Xplore*
9. Al-diabat M et al (2018) Fuzzy data mining for autism classification of children. *Int J Adv Comput Sci Appl* 9(7)
10. Bone D et al (2014) Applying machine learning to facilitate autism diagnostics: pitfalls and promises. *J Autism Dev Disord* 45(5):1121–1136
11. Duda M et al (2016) Use of machine learning for behavioral distinction of autism and ADHD. *Transl Psychiatry* 6(2):e732–e732
12. Kosmicki JA et al (2015) Searching for a minimal set of behaviors for autism detection through feature selection-based machine learning. *Transl Psychiatry* 5(2):e514–e514
13. Levy S et al (2017) Sparsifying machine learning models identify stable subsets of predictive features for behavioral detection of autism. *Mol Autism* 8(1)
14. Pancarz K et al (2015) Consistency-based preprocessing for classification of data coming from evaluation sheets of subjects with ASDs. In: Position Papers of the 2015 federated conference on computer science and information systems

15. Pratap A et al (2014) Predictive assessment of autism using unsupervised machine learning models. *Int J Adv Intell Parad* 6(2):113
16. Wall DP et al (2012) Use of artificial intelligence to shorten the behavioral diagnosis of autism. *PLoS ONE* 7(8):e43855
17. Rabbi MF et al (2023) Autism spectrum disorder detection using transfer learning with VGG 19, Inception V3 and DenseNet 201. *Commun Comput Inf Sci* 190–204
18. Kalaiselvi A et al (2021) Detection of autism spectrum disorder using transfer learning. *Semantic Scholar*
19. M. Tan and Q. Le (2019) EfficientNet: rethinking model scaling for convolutional neural networks. *proceedings.mlr.press*
20. Mujeeb Rahman KK et al (2022) Identification of autism in children using static facial features and deep neutral networks. *Brain Sci* 12(1):94
21. Dataset from kaggle. <https://www.kaggle.com/datasets/cihan063/autism-image-data>
22. Aldridge K et al (2011) Facial phenotypes in subgroups of prepubertal boys with autism spectrum disorders are correlated with clinical phenotypes. *Mol Autism* 2(1):15

Lung Cancer Prediction Using Integrated Deep Learning Model



M. V. Sudhamani  and Anil Kumar 

Abstract This paper presents a thorough scheme that combines a ConvNet and Long Short-Term Memory (LSTM) model to predict lung cancer using CT scans. This integrated model accepts preprocessed CT scan images and classifies them into three groups: benign, malignant, and normal cases. This work aims to contribute improved lung cancer diagnosis and patient care by enabling early identification of malignant cases and improving the model's capacity to evaluate both temporal and spatial data for precise predictions. The CNN extracts spatial features, capturing local patterns and visual cues related to lung cancer, whereas the LSTM captures temporal dependencies, enabling the method to study from sequential CT scan data. The proposed work aims to develop a robust prediction model, optimize its performance through suitable preprocessing techniques and architecture fine-tuning, and compare the effectiveness with existing approaches.

Keywords Cancer · CNN (ConvNet) · Prediction analysis · CT scans · LSTM

1 Introduction

Lung cancer detection is a critical task in health care and advancements in medical imaging and ML has opened the door for further development. Here, a modular design that addresses the challenges of lung cancer discovery by integrating Long Short-Term Memory (LSTM) into a comprehensive system is discussed. The design consists of interconnected modules, including data acquisition, image preprocessing,

M. V. Sudhamani (✉) · A. Kumar
Department of ISE, BMS Institute of Technology and Management, VTU, Bengaluru, India
e-mail: dr.mvs@bmsit.in

A. Kumar
e-mail: anil.kumar@bmsit.in

and model building stages. Acquiring data is the process of obtaining pertinent information from the source, and the image preprocessing modules enhances the superiority of lung cancer images. The model building stages encompass model construction using Synthetic Minority Over Sampling Technique (SMOTE), Class-Weighted Approach (CWA), and Data Augmentation. These stages address the class imbalance and focus on forecasting algorithms to recognize lung cancer. The output of individual model is compared on the basis of various parameters. This approach aims to improve accuracy, mitigate class imbalance, and provide robust predictions for effective lung cancer diagnosis and treatment planning. By integrating LSTM alongside other modules, end-to-end performance and efficacy of our comprehensive system are tested. This paper presents the implementation details, experimental setup, and evaluation results of our proposed modular design.

2 Related Works

A ML approach [1] was employed to predict lung cancer using AdaBoost, an ensemble ML approach that predicted the various forms of lung cancer. Performance metrics were used for evaluation, and ensemble learners were trained on the retrieved features from the lung CT images. AdaBoost was given a 90.74% accuracy, 81.80% sensitivity, 0.81 F1-score, and 0.75 kappa. Comparing this paper's backpropagation accuracy to other neural networks, however, shows that it is insufficient at only 85%.

According to an investigation of several machine learning (ML) algorithms in [2] utilizing numerous metrics, including sensitivity, specificity, F1-score, and support, random forest had the greatest accuracy at 99.09%, followed by logistic regression at 97.09%.

For diagnosing lung cancer, in [3] proposed a classification model using multiple ML and DL techniques, with nodule volume as the key parameter. The authors collected a dataset of lung nodules and their volumes, applied feature extraction techniques, and trained ML models as random forest, support vector machine (SVM), and k-nearest neighbors and DL model as CNN to classify the nodules on the basis of cancerous or non-cancerous. The results demonstrated high accuracy in classification, with the DL model exhibiting slightly better performance. As per the study, deep learning may help with nodule volume-based lung cancer diagnosis.

The work discusses developing a preemptive diagnosis model [4] for Branchio-genic Carcinoma using ML techniques with medical data. The study utilizes a dataset containing clinical information of lung carcinoma patients, and several ML techniques, such as SVM, random forests, and decision trees, to train models that estimate the risk of lung cancer. The results demonstrate good accuracy in preemptive diagnosis, enabling the identification of individuals at risk of developing lung cancer and facilitating initial detection and intervention for improved treatment outcomes. The actual findings showed that when full features were used, SVM and LR classifiers produced the greatest accuracy of 97.27%. But as SVM offered the best recall rate of 98.33%, it fared better than LR.

A method for identifying early-stage lung cancer using an Artificial Neural Network (ANN) classifier was published in [5]. Twenty photographs of cancer and twenty images of normalcy were used in the trial. In the cancer scans, the researchers found 29 desired nodules, and in the normal images, they found 16 desired nodules. The end-to-end performance of the ANN classifier was evaluated based on its capability to predict whether a lung nodule is benign or malignant, given a threshold value of 50%. With respect to the results, when the ANN prediction exceeded the 50% threshold value, the nodules were classified as malignant, while predictions below 50% were classified as benign. The ANN consistently provided predictions above 85% for the cancer images.

Focused on using weakly supervised DL techniques [6] for analyzing whole slide lung cancer images. The authors projected a method that leverages weakly labeled data, where only slide-level labels are available instead of pixel-level annotations, for training DL models. The thing is to directly describe and classify lung cancer regions within whole slide images. The authors reported a delicacy of 89.6% for whole slide lung cancer bracket, and also, the system achieved mean crossroad over Union (IoU) of 76.2% for lung cancer region segmentation, which indicates the delicacy of the model in relating the boundaries of cancerous regions within t slides.

In [7] presented a multi-stage approach for lung cancer discovery and vaticination, employing a multi-class Support Vector Machine (SVM) classifier. The authors' proposed meth achieves significant results with certain performance parameters. The multi-class SVM classifier successfully predicts the presence and type of lung cancer using the extracted features obtained through the multi-stage framework which achieved an accuracy of 91.8% for lung cancer discovery and an accuracy of 89.2% for cancer-type prediction using the multi-class SVM classifier. With an accuracy of 97%, the suggested algorithm identified 126 photos as malignant and 4 as non-cancerous out of 130 images with cancer. These statistics shown how good the proposed approach works to identify and categorize cases of lung cancer.

This paper discusses a real-time application machine learning-based system [8] for predicting lung cancer. An automated process for approximating lung cancer stage is proposed in this paper. The suggested approach predicts the degree of lung cancer as high, medium, or low using the KNN algorithm. The findings obtained indicate a 95.8% accuracy rate. It is implemented as a web-based system to give the outcomes of the classification model which ultimately provides a better user-interface.

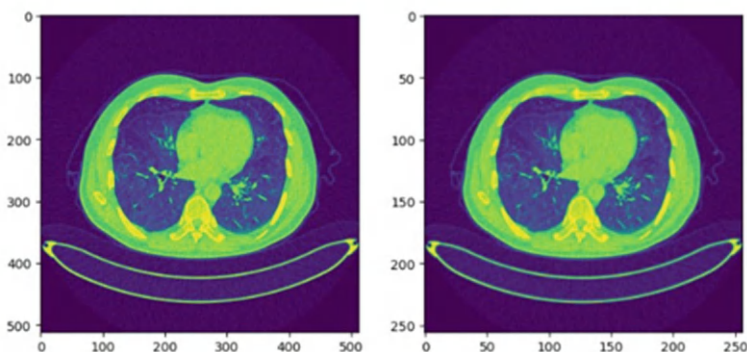


Fig. 1 Preprocessed sample images

3 Working Model

3.1 Datasets

The National Centre for Cancer Conditions at the Iraq Oncology Teaching Hospital provided the dataset. Since the dataset is freely accessible on Kaggle, it contains CT reviews of both healthy individuals and cases with several phases of lung cancer diagnosis. Oncologists and radiologists marked the IQ- OTH/NCCD slides. Three kinds of these cases are distinguished: normal, benign, and malignant. Of them, forty have been diagnosed as malignant, fifteen as benign, and fifty-five as normal instances. Initially, the CT reviews were gathered in DICOM format. Siemens' SOMATOM scanner is currently in use. The CT protocol consists of the following parameters: 120 kV, 1 mm slice thickness, window width and center ranges from 350 to 1200 HU and 50 to 600, respectively, were employed for readings while the breath was held at full inspiration. Analyses were conducted when each image was de-identified. There are multiple slices in every checkup. There are between seventy and two hundred of these slices, each one depicts a different side and angle of the fatal box. The instances differ in terms of living situation, area of hearthstone, age, gender, and level of education. A portion of them work for the Ministry of Transportation and Oil Painting in Iraq, while others are cultivators and winners. The majority of them are from Iraq's middle region, specifically from the cities of Wasit, Diyala, Babylon, Baghdad, and Salahuddin. Preprocessed sample lung CT scans are displayed in Fig. 1.

3.2 Methodology

1. Collect a diverse and representative dataset of lung CT scans, including images of both benign and malignant cases, along with normal scans.

2. Apply image preprocessing techniques like resizing, normalization, and noise reduction to standardize the images. Use SMOTE to address class imbalance by generating synthetic samples for the minority class (lung cancer cases). Implement Data Augmentation to increase dataset diversity and improve model generalization. Implement Class-Weighted Approach (CWA) to assign higher weights to the minority class during training, thus emphasizing its importance in the model.
3. Selecting ConvNets and LSTM networks as the most effective DL methods for image categorization is crucial.
4. Design the CNN or LSTM architecture with suitable layers, activation functions, and pooling operations. Train the model using the preprocessed data, with the goal of minimizing the loss function (e.g., sparse categorical cross-entropy).
5. Only by utilizing assessment criteria on a validation dataset, such as precision, recall, accuracy, and F1-score, can the efficacy of the model be determined. To maximize the model's effectiveness, the hyperparameters for learning rate and batch size must then be fine-tuned.
6. Evaluate the final model on a separate test dataset to measure its accuracy in real-world scenarios. Interpret the results through a confusion matrix to understand true-positives, false-positives, true-negatives, and false-negatives for each class.
7. Create a user-friendly interface to accept CT scan inputs in batches. Use the trained model to predict lung cancer presence in the input scans. Display the predicted results with corresponding probabilities and class markers for easy interpretation.
8. Analyze the Neural Network's limitations and identify potential areas for improvement. Explore additional techniques like ensemble methods or multi-modal data integration to build on accuracy and robustness. Collaborate with medical professionals to confirm the model's impact on real-world clinical datasets.
9. Summarize the model's prediction results and highlight the potential effect of the work on initial analysis and treatment preparation for better patient outcomes.

3.3 *Proposed Architecture*

Figure 2 displays the suggested system's modular design, which encompasses a systematic approach to resolve the challenges in lung cancer detection. The design consists of several interconnected modules, each with a distinct function and adding to the system's overall precision and efficiency. The initial stage of the modular design is data acquisition and ingestion. This entails compiling pertinent information from a sequence of sources, including imaging databases, clinical research, and medical records. The collected data is then processed and prepared for further analysis.

Following data acquisition, the next module in the design is image preprocessing. This stage emphasizes on improving the usability and applicability of the lung cancer images. Preprocessing techniques are used to the acquired images, include denoising,

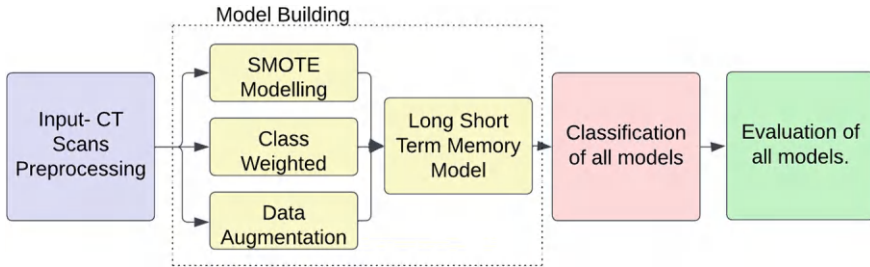


Fig. 2 Proposed architecture

normalization, contrast adjustment, and image registration. These techniques aim to reduce noise, standardize the images, and improve the visibility of important features for subsequent analysis. The modular design then splits into three different stages: Model Building using SMOTE, CWA, and data augmentation approach. These stages represent different methodologies or algorithms employed in the development of the cancer detection models. In the Model Building using SMOTE stage, the SMOTE technique is utilized. It is a popular algorithm used to address class disbalance in ML tasks. It generates synthetic examples of the smaller class (e.g., lung cancer cases) to balance the data archive and prevent bias toward the outvoted class (e.g., non-cancer cases). This stage involves training a model using the balanced dataset to build a predictive model specifically for lung cancer detection. In the Model Building using the CW approach stage of the system, the data intake and collection phase are the initial phase in the modular design process. To do this, pertinent data must be gathered from a variety of sources, including clinical studies, imaging databases, and medical records. The collected data is then processed and prepared for further analysis.

Approach assigns different costs or weights to different classes to handle class imbalance effectively. By assigning higher costs or weights to the minority class, the model can provide more importance to the discovery of cancer cases. This stage involves training another model using the cost-weighted dataset to build separate predictive framework for lung cancer detection.

Finally, both modules are integrated in the subsequent stage, where the outcomes of the individual models are combined. The integration process may involve ensemble techniques, such as stacking or voting, to take advantage of both models and raise the overall predicted accuracy. The integrated model provides the final results, which can include predictions of lung cancer presence, stage, or other relevant information. Overall, the modular design ensures a systematic and comprehensive approach to lung cancer detection. By incorporating data acquisition, image preprocessing, and two distinct model building stages, the proposed system aims to enhance accuracy, mitigate class imbalance issues, and provide robust predictions for effective diagnosis and treatment planning.

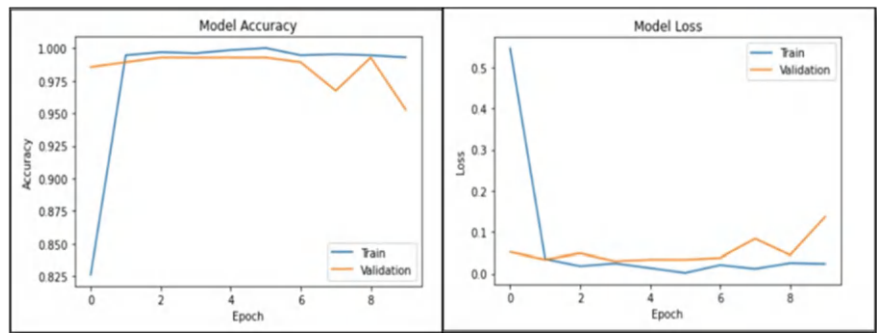


Fig. 3 SMOTE model accuracy and loss

4 Results and Discussion

4.1 SMOTE Modeling Approach

The outcomes of the SMOTE technique for lung cancer prediction indicate high precision, recall, and F1-score values across all three classes: benign (0), malignant (1), and normal (2). For each class, the precision values are 0.96, 0.99, and 0.94, respectively, that can be inferred from Fig. 3, as it indicates a high proportion of properly predicted examples for each class. The recall values, measuring the proportion of actual positive instances correctly identified, are 0.80, 1.00, and 0.97 for benign, malignant, and normal classes, respectively. These values demonstrate the model’s ability to effectively detect instances of lung cancer across different classes. The neural network accurately predicts the class labels for 95% of the existences in the dataset, conferring to the overall metrics of model, which are reported as 0.95. This high accuracy demonstrates the effectiveness of the SMOTE technique in addressing the class imbalance in the neural network.

4.2 Class-Weighted Approach

The outcome of CWA for lung cancer prediction indicates exceptional performance across all three classes: benign (0), malignant (1), and normal (2). The precision values for all classes are 1.00, 1.00, and 0.98, respectively, demonstrating that the model has correctly identified a high proportion of instances. The recall values for all three classes are also excellent, with values of 0.93, 1.00, and 1.00. These results show that the model effectively captures the positive cases for individual class as shown by Fig. 4. According to reported accuracy of the model, 0.99 indicates that it predicts class labels for 99% of the instances in dataset with accuracy. As realized

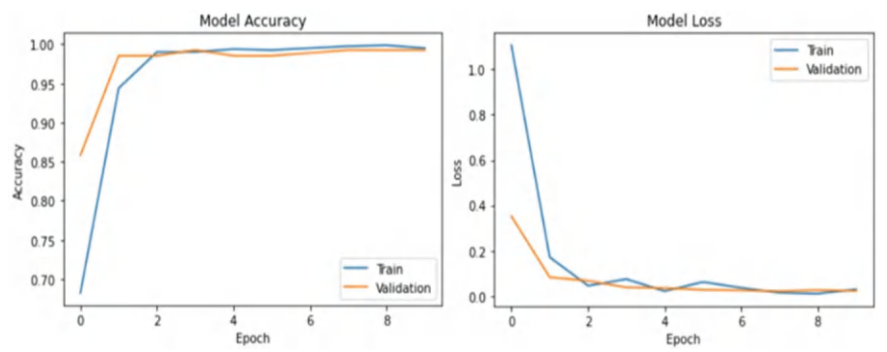


Fig. 4 Class-weighted model accuracy and loss

in Fig. 4, this high degree of accuracy illustrates how well CWA works to enhance the performance of the lung cancer prediction model.

4.3 Data Augmentation Approach

Data Augmentation has yielded promising outcomes with lung cancer prediction. For the three classes benign, malignant, and normal, the precision, recall, and F1-score values are constantly high and promising results are attained with this model. The precision values range from 0.88 to 0.98, suggesting a high proportion of successfully predicted data. For the three classes benign, malignant, and normal, the precision, recall, and F1-score values are constantly high. The precision numbers show a high percentage of correctly predicted instances for every class, varying from 0.88 to 0.98. The recall values, which fall between 0.93 and 0.99, show how well the model can recognize positive examples. Strong overall performance is shown by the F1-scores, which range from 0.90 to 0.98 and measure the balance between precision and recall. The model’s reported accuracy of 0.96 indicates that it can accurately predict the class labels in 96% of cases. Average values of all models for all classes are given in Table 1.

Table 1 Average performance for classes (Support = 30)

Technique	Precision	Accuracy	Recall	f1-score
SMOTE modelling	0.96	0.97	0.80	0.87
Class weighted	1.0	0.99	0.93	0.97
Data Augmentation	0.88	0.94	0.93	0.90
Long Short-Term Memory	0.87	0.80	0.83	0.79

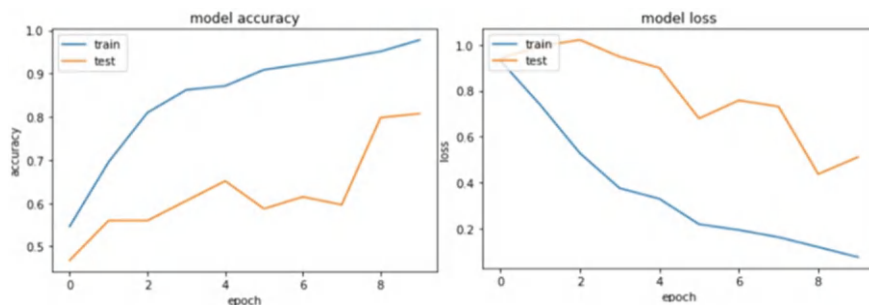


Fig. 5 Long Short-Term Memory model accuracy and loss

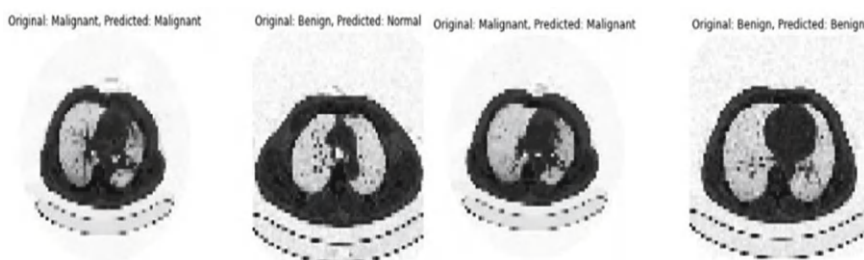


Fig. 6 Sample model outputs

4.4 Implementation of LSTM Approach

The outcomes of the LSTM for cancer prediction show modest performance across the three classes: benign (0), malignant (1), and normal (2). The precision values for all classes are relatively low, with 0.80, 0.83, and 0.87 for benign, malignant, and normal classes, respectively. This specifies that the model struggles to accurately predict instances for all classes. However, low accuracy reflects the limitations of the LSTM model in accurately classifying lung cancer cases from CT scan data because of small size of dataset. This inconsistency in the dataset can be noticed in Fig. 5. Sample predicted outputs are shown in Fig. 6.

5 Conclusion

The paper has discussed a comprehensive investigation into diverse methodologies to predict lung cancer. The study encompasses the efficacy of the SMOTE technique in handling class imbalance, yielding remarkable precision, recall, and F1-score values across all classes. The Class Weighting Approach demonstrates exceptional precision and recall, attaining an impressive overall accuracy of 99%. Data Augmentation

showcases consistent high performance in precision, recall, and F1-scores, with a 94% accuracy. However, the LSTM model's precision is relatively lower due to dataset limitations, reflecting challenges in accurately classifying lung cancer cases from CT scan data. This work underscores the significance of tackling class imbalance and highlights the potential of different techniques in advancing prediction of lung cancer, while acknowledging the need for further refinement.

References

1. Zhou J, Khushi M, Moni MA, Uddin S, Poon SK (2021) Lung cancer prediction using curriculum learning based deep neural networks. In: 2021 IEEE International Conference on Digital Health (ICDH), pp 11–18. Chicago, IL, USA
2. Sruthi G, Ram CL, Sai MK, Singh BP, Majhotra N, Sharma N (2022) Cancer prediction using machine learning. Second International conference on innovative practices in technology and management (ICIPTM), India, pp 217–221
3. Mahadevaswamy UB, Siddhanti SR, Bhumika SM, Mounashree B (2021) A real time application for lung cancer prediction using machine learning. Second International conference on communication, computing and industry 4.0 (C2I4), pp 1–6. Bangalore, India
4. Kumaran SY, Jeya JJ, Khan SB, Alzahrani S, Alojail M (2022) Lung cancer detection using machine learning. Int Conf App Artif Intell Comput (ICAAIC), Salem, India, pp 539–543
5. Thallam C, Peruboyina A, Raju SST, Sampath N (2020) Early-stage lung cancer prediction using various machine learning techniques, In: 4th International conference on electronics, communication and aerospace technology (ICECA), Coimbatore, India, pp 1285–1292
6. Bhise SS, Khot SR (2021) Early-stage lung cancer diagnosis using ANN classifier. Int Conf Artif Intell Smart Syst (ICAIS), pp 22–27, Coimbatore, India
7. Olatunji SO et al (2022) Machine learning based preemptive diagnosis of lung cancer using clinical data, In: Seventh international conference on data science and machine learning applications (CDMA), pp 115–120, Riyadh, Saudi Arabia
8. Saini S, Maithani A, Dhiman D, Bisht A (2021) Analysis of different machine learning algorithms used for identification of lung cancer disease. In: Ninth international conference on reliability, infocom technologies and optimization (Trends and Future Directions) (ICRITO), pp 1–5, Noida, India

Leaf Disease Detection and Classification in Grapevine and Potato Leaves Using Deep Learning



A. Hari Krishna, K. Ranjith, and Jeshma Nishitha Dsouza

Abstract Global agriculture is seriously threatened by plant diseases, which lower agricultural production and cause financial losses. Deep learning models have made significant progress in the past few years in automating the identification of leaf diseases, allowing for early intervention and customized therapies. This study focuses on the use of three cutting-edge deep learning models for the detection and classification of different types of illnesses on grape and potato leaves: YOLOv5, YOLO-NAS, and YOLOv8. Our research focused on the identification of specific leaf diseases, including leaf blight, ESCA, and black rot in both grapevine and potato leaves. The primary objective was to enhance the speed and accuracy of these algorithms for real-time detection and classification. The research results revealed a remarkable mean Average Precision (mAP) value of 98.6% for the YOLOv8 model. This finding suggests that, among the three models we trained, YOLOv8 demonstrated the highest level of accuracy and efficiency in detecting and classifying leaf diseases.

Keywords Black rot · Leaf blight · ESCA · Deep learning · YOLOv5 · YOLO-NAS · YOLOv8

1 Introduction

Plant diseases impact crop yields, food security, and economic stability. Hence timely detection and intervention are crucial for effective disease management, as early identification of infections can prevent their rapid spread and minimize yield losses. Our study mainly focuses on the application of these advanced deep learning models for disease detection on the leaves of two agriculturally important crops: grapes and potatoes. The significance of these crops cannot be understated, as the grapes are the foundation of the multi-billion-dollar wine industry and potatoes are among the most widely consumed staple foods worldwide. Now as technology is converting a lot of

A. H. Krishna · K. Ranjith · J. N. Dsouza (✉)

Department of Advanced Computing, St Joseph's University, Bengaluru, Karnataka, India
e-mail: jeshma.dsouza@sju.edu.in; jeshma.research@gmail.com

imaginary ideologies into reality, our trained models have taken some spotlight in reducing the traditional way of manual inspection in detecting leaves and increasing the detection and classification with high accuracy and speed.

In our approach, we delve into three major diseases affecting grapevines and potato leaves—ESCA, leaf blight, and black rot. The chosen diseases are prevalent and widespread, making them major concerns for grape and potato growers. By focusing on these diseases, the research aims to address issues that are commonly encountered by farmers, thereby increasing the practical relevance of the study. Let's have quick introduction toward these three major categories:

1.1 ESCA

ESCA, also known as grapevine trunk disease, is a fungal infection that primarily targets the woody tissues of grapevines. This insidious disease can lead to a slow decline in vine health, marked by characteristic foliar symptoms such as yellowing and browning of leaves. The disease is of significant concern to viticulturists, as it affects both the quality and quantity of grape production. Subsequent paragraphs, however, are indented.

1.2 Leaf Blight

Another prevalent disease affecting grapevines is called leaf blight, which is brought on by a range of fungi. As seen in the diagram, it appears as the browning and wilting of potato and grape leaves. This has an impact on the vines' ability to photosynthesize their leaves. The illness may cause early leaf drop and a reduction in the quality of potatoes and grapes. This will be divided into the following two subclasses:

Potato Leaf Early Blight

Unlike grapevine leaves, potato leaves are affected by leaf early blight. Potato early blight usually appears as dark, irregularly shaped to circular lesions on the lower leaves. These lesions frequently resemble targets with concentric rings around them.

Potato and Grape Leaf Late Blight

Phytophthora infestans, a water mold pathogen, is the cause of late blight on both grape and potato leaves. The symptoms of late blight are typically dark, asymmetrical lesions with a pale green halo surrounding them.

1.3 Black Rot

This disease mainly affects grapevines, particularly those planted to produce wine. It's not frequently connected to potato leaves. On grape leaves, this illness appears as tiny, reddish-brown dots. These enlargements become more pronounced and larger as the infection worsens (Fig. 1).

2 Related Work

Visual inspection of leaves for indications including lesions, discolorations, and distortions is frequently used to identify leaf diseases early. Finding a dataset with correct photos of leaf disease is difficult, and it can be difficult to train deep learning models and develop an accurate model for our dataset. However, looking at studies done on problems with similar assertions gave us a clear concept of the models we could employ.

An improved vgg16 model, Syed Amir Hussein (2022) was able to successfully classify all leaf diseases in grapevines with an accuracy of 99%. The key feature of this classification is the automation process, which detects the disease and sends signals to quadcopters to spray pesticides [1]. Using multispectral images obtained during in vivo (under lab conditions) experiments, Alesandro Benfenati (2023) has employed unsupervised deep learning techniques, such as convolutional auto encoders, for the automatic detection of plant diseases [2]. With a custom dataset,

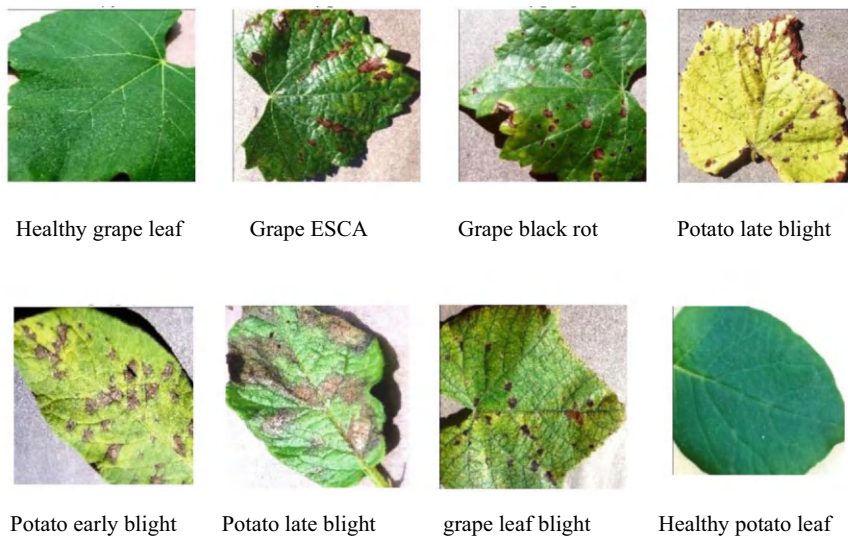


Fig. 1 Types of grape and potato leaf diseases along with the picture of healthy leaves

Hilman F. Paridede (2020) experimented with a variety of deep neural networks, including Alexnet, vggnet, Googlenet, Xception, Mobilenet, zfnets, and Resnet. He was able to classify leaf diseases with (90–98%) accuracy for all the models [3]. Better results were obtained by Md. Ershadul Haque and colleagues who implemented a YOLOv5 model for rice leaf disease classification and detection with a mAP value of 81 [4]. The paper proposed by Zhaoyi Chen integrates the Involution Bottleneck and SE modules, improves the loss function, and proposes an improved YOLOv5 model for accurate plant disease recognition. A mean average precision of 70% is attained through testing on a rubber tree disease database, showing a 5.4% improvement in performance over the original YOLOv5. With 86.5% and 86.8% accuracy rates, respectively, in diagnosing powdery mildew and anthracnose, this enhanced model shows promise for practical disease diagnosis in agriculture [5]. EDMUNDO CASAS evaluated the YOLO models using a dataset of smoke and wildfires for detection. Based on metrics, the models were compared, with YOLONAS outperforming other models in terms of recall. The analysis concluded that there isn't a single model that fits all aspects of the detection problem and that the best model to use depends on the requirements [7]. M. Alessandrini (2021) achieved a 99% accuracy rate in detecting the ESCA disease in grape vine leaves using a basic convolutional neural network (CNN) [9].

3 Research Domain and Methodology

3.1 Dataset Overview

The leaf disease detection image dataset, sourced from Roboflow, comprises a set of photos of plant leaves, including grapevine and potato leaves, afflicted with major diseases like leaf blight, ESCA, and black rot. The images are further classified into early and late subclasses, which indicate the stage at which the disease has spread throughout the leaf. The dataset consists of 6969 photos that have been split up into 7 classes, each of which represents a different type of leaf disease or healthy leaves. The photos were taken in well-lit environments and are appropriate for image classification applications. After closer examination, we discovered a class imbalance that may have biased the models and decreased their performance. As a result, we carefully trimmed the dataset to eliminate classes with extremely few photos. Ultimately condensing the pictures into a manageable and training-appropriate format.

The Leaf Disease Detection dataset's salient features consist of:

Widespread Coverage: The collection includes seven different leaf diseases, which are the most prevalent plant diseases.

High-Quality Photos: The high-resolution images are taken in a variety of lighting circumstances, guaranteeing the dataset's resilience and suitability for use in practical contexts.

Distribution that Is Balanced: Following preprocessing, the dataset shows that the photos are distributed fairly among the various classifications with its ability to facilitate early plant disease prevention, and the leaf disease detection dataset has the potential to have a substantial impact on the field of agricultural technology. This can therefore result in prompt interventions, increased crop yields, and a decrease in farmer's financial losses.

4 Models

In the field of agricultural technology, the application of advanced machine learning techniques has revolutionized the way we address plant health and crop management challenges. Among these technologies, convolutional neural networks (CNNs) [6] have emerged as a pivotal tool for processing images, particularly in the context of leaf disease classification. CNNs are equipped with specialized convolutional layers, have proven their efficacy in handling the image data, and are now at the forefront of research in the critical task of classifying leaf diseases in grape and potato leaves.

YOLOv5, YOLO-NAS and YOLOv8 are the three prominent deep learning architectures in the YOLO family [8] which plays a vital role in this research. They are harnessed to tackle the pressing issue of identifying and categorizing diseases in these essential agricultural plants, with the potential to revolutionize crop management and ensure food security for our growing population.

4.1 Dataset Overview

The optimization of data preprocessing and augmentations of leaf disease classification data stands as a pivotal strategy to enhance the model performance. The data preprocessing pipeline encompasses several critical steps such as Auto orienting images to ensure consistency in orientation, applying static cropping to focus on central regions of the leaves, resizing all images to a uniform 416×416 resolution for standardization, employing contrast stretching to enhance image contrast, and simplifying the class labels through remapping and dropping. These preprocessing steps not only streamline the data but also provide the model with more informative and standardized inputs. Moreover, data augmentation techniques are strategically applied to diversify the dataset. By generating three outputs for each training example, the dataset is enriched, and the model learns from various angles and viewpoints. Horizontal and vertical flips introduce variability, while 90-degree rotations in both clockwise, counterclockwise, and upside-down orientations increase adjustments which empower the model to better handle diverse leaf conditions and orientations. We have mainly used Pytorch and super Gradients library to prepare the dataset. For YOLOv5 [9], the data preparation is done by exporting the dataset in the YOLOv5 Pytorch format from Roboflow. For YOLO-NAS [9, 10], the dataset is

exported as YOLOv5 Pytorch format similar to the YOLOv5 model. The data is loaded for training, validation, and test set using super Gradients library. Batch size is kept the same as the previous model, and few transformations are applied to the training dataset using transformers. Few data augmentations are applied such as zooming, shifting, normalizing pixels value which enhances the dataset's diversity. Finally, for YOLOv8 the data has been fed in the YOLOv8 format which was extracted from Roboflow's export option.

4.2 *Yolov5*

YOLOv5 is a state-of-the-art real-time object detection model that is renowned for its speed and accuracy. It is an improvement over its predecessors in terms of speed and accuracy thanks to its increased user-friendliness, and it performs admirably on our leaf disease detection dataset. As a first step in order to extract and represent features, the YOLOv5 first uses a bottle and neck layer followed by a convolutional neural network for classification and detection. Its backbone network, CSPDarknet53, is built on CSP modules as the bottle layer. Since this backbone is utilized for all feature extraction tasks, when we input our dataset of leaf diseases, all the features required for detection and classification will be extracted in this network. To better capture our leaf disease features, it also includes a neck section with extra convolutional and up sampling layers. The model's last layer is where the last classification or detection is done using objectness scores, bounding boxes, and class predictions derived from the features extracted by the bottle and neck layers.

4.3 *YOLO-NAS*

You only look once—Neural Architecture Search (YOLO-NAS) is an innovative object detection model, which involve both classifying and localizing objects within an image. Developed by Deci (An AI research company) using their proprietary NAS technology. YOLO-NAS integrates re-parametrization and 8-bit quantization advantages through Quantization-Aware Selection (QSP) and Quantization-Aware Combination (QCI) blocks. These blocks are devised based on an approach proposed by Chu et al., ensuring a minimal accuracy loss during post-training quantization. YOLO-NAS is pre-trained using an advanced regimen, including automatically labeled data, self-distillation, and large datasets. The model training process follows a comprehensive and advanced approach, utilizing Deci's Super Gradients library.

The chosen YOLO-NAS architecture, pre-trained on the coco dataset, is fine tuned for object detection tasks. Training parameters are meticulously configured, employing a cosine learning rate schedule, Adam optimizer with weight decay, and mixed precision for efficiency. The PPYOLOELoss, a specialized loss function, is employed, considering the dynamic nature of assignments and allowing for superior

object detection. The entire training paradigm is designed to strike a balance between precision and efficiency, ensuring the YOLO-NAS model's optimal performance in detecting the grapevine and potato leaf diseases.

4.4 YOLOv8

YOLOv8 is a cutting-edge model that often performs with higher precision and speed than its predecessors. The architecture of YOLOv8 is:

Backbone—It employs a modified version of CSPDarknet53 and CNN as the backbone network to extract features from the input photos, which allows it to extract the features from our leaves image more quickly and correctly.

Neck—YOLOv8 uses a combination of SPPF and C2F modules at its neck, which serves as the connection between the backbone and the head.

Head—The head oversees forecasting the bounding box, or the area, that denotes the presence of disease, as well as the objectness score, or the specific leaf disease score, which informs us of the confidence of the forecasted image to its associated class. YOLOv8 used the head of the Path Aggregated Network.

The model generates a predicted/classified image of a leaf along with a confidence value that ranges from 0 to 1.

The actions entail:

- The leaf image dataset should be preprocessed, including being resized to standard sizes.
- Processed Leaf image will be sent to the model.
- Obtain the objectness score and predicted bounding boxes.
- Get the predicted bounding boxes and class score.
- Classify the diseases in the dataset using the class scores and the predicted bounding boxes.

5 Methodology

To increase the variety of training examples, the dataset was augmented, with three distinct transformations applied to each image. These included flips both horizontally and vertically, clockwise, and counterclockwise 90° rotations, and an upside-down rotation. Furthermore, different degrees of cropping were applied to the images, with a minimum zoom of 0% and a maximum zoom of 10%. To add variation in brightness, changes were made between -25% and + 25%. To enable thorough model evaluation, the dataset was carefully divided into training, testing, and validation sets after the augmentation process. Models were trained to learn and generalize from the various transformations applied to the images using the augmented training

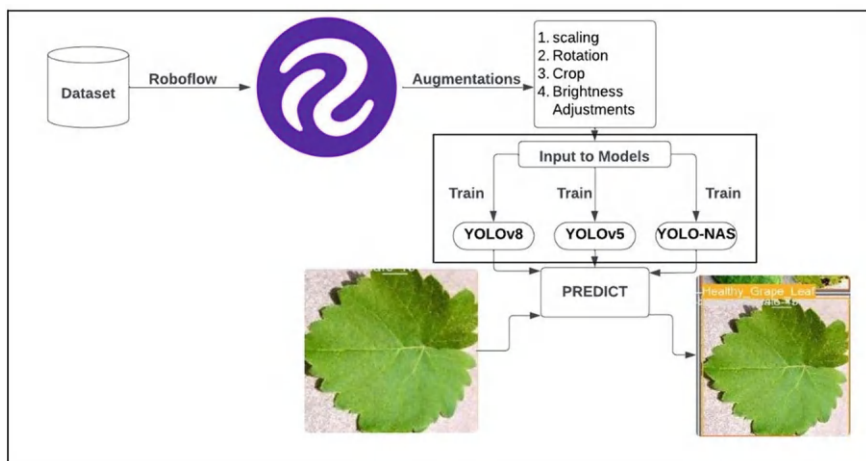


Fig. 2 Figure explains the model's complete workflow and the bounding box and model's prediction are shown in the last picture

data. After that, datasets were downloaded from Roboflow and given as input to the models. The models were then used to predict images from the test set, which made a thorough assessment of their performance possible. Several metrics were used to evaluate the models' performance, giving information about their recall, precision, and mAP. This methodical approach guarantees a thorough grasp of the models' capabilities and robustness in handling various image scenarios, from data augmentation to model training and evaluation (Fig. 2).

6 Performance Criteria

6.1 Classification Metrics

Classification metrics offer a comprehensive assessment of the model's performance on a specific dataset, furnishing valuable insights into metrics such as precision, recall, and mAP.

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}, \quad (1)$$

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}. \quad (2)$$

6.2 MAP Value

Mean average precision, or mAP for short, is one of the better metrics for evaluation since it assesses the model's total performance by accounting for both recall and accuracy. The formula for mAP is:

$$\text{mAP} = \int_0^1 P(R) \cdot dr. \quad (3)$$

Additionally, calculating the average precision for each class and averaging it across the total number of classes yield the mAP value.

6.3 Intersection over Union (IoU)

Intersection over union (IoU) is a crucial metric in evaluating the performance of object detection model models like YOLO family. In this family of models, IoU is commonly used to measure the accuracy of predicted bounding boxes concerning ground truth bounding boxes. IoU is calculated as the ratio of the area of the overlap between the predicted and ground truth bounding boxes to the area of their union. For a given predicted bounding box P and ground truth bounding box G, the IoU is computed using the following formula:

$$\text{IoU}(P, G) = \frac{\text{Area of overlap}}{\text{Area of union}}, \quad (4)$$

$$\text{IoU} = \frac{\text{True positive}}{\text{True positive} + \text{False positive} + \text{False negative}}. \quad (5)$$

7 Results and Discussion

In this study, we have used a dataset containing the grape and potato leaf images with its respective diseases labeled. To gauge the effectiveness of these models, we employed cross-validation techniques and a suite of performance metrics, providing an in-depth assessment of their accuracy and overall performance. The valuable insights derived from our analysis are encapsulated in a series of informative figures, namely Fig. 3, 4, 5, 6 and 7 which vividly illustrate the model's loss and mean average precision profiles, offering a visual snapshot of their respective capabilities in the context of leaf disease classify (Tables 1 and 2).

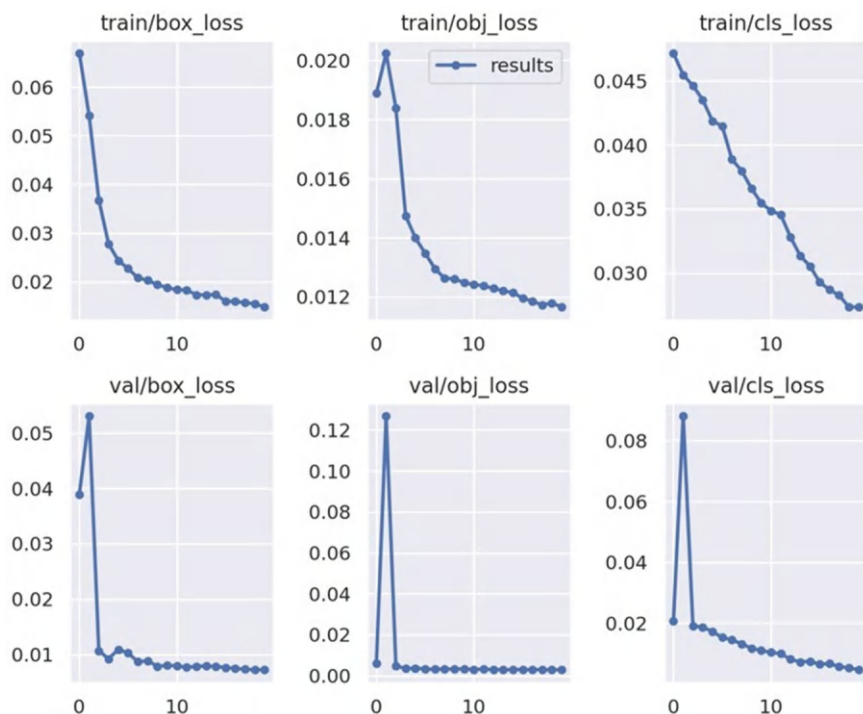


Fig. 3 Graphs plot the training and validation losses for three different types of loss: bounding box loss, objectness loss, and classification loss. The loss values for both training and validation decrease as the number of training steps increases

8 Conclusion

According to Eugene Khvedchenya and the Deci team's research [8], the YOLO-NAS model demonstrates superior performance compared to YOLOv8 and the other models based on the COCO dataset. But our specific investigation into leaf disease detection and classification underlines the expectational capabilities of YOLOv8, showcasing an impressive mAP value of 98.6, exceeding YOLO-NAS by 4.9%, and surpassing YOLOv5. YOLOv8 consistently excelled across all classes. This success stems from its unique proficiency in both detection and classification tasks, positioning YOLOv8 as the optimal choice for real-time leaf disease detection, providing a promising application for precise agricultural activities by enabling timely and accurate identification of plant diseases to enhance crop management and yield. The future work involves utilizing this technology to be deployed and been used in mobile phones to make it easy for the farmers to get access to this technology. Introducing the concept of Ensemble learning can yield even more greater mean average precision (mAP) and recall. The domain knowledge can be expanded by utilizing these models in detecting how these plant diseases can affect the neighboring plants.



Fig. 4 Training and validation losses for a YOLO-NAS object detection model. The graphs show that both the training and validation losses decrease over time

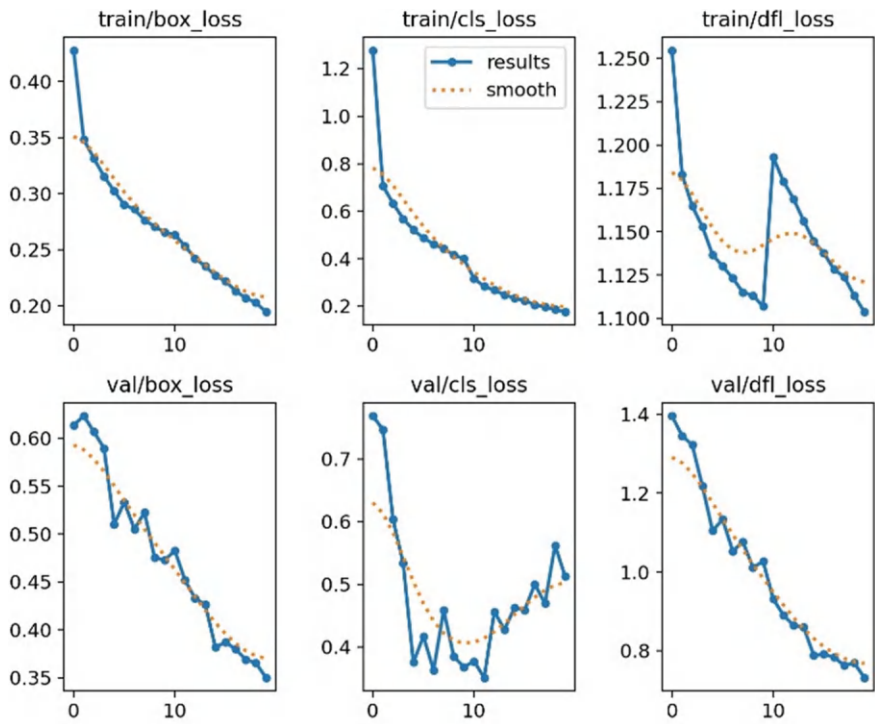


Fig. 5 Graph illustrates the YOLOv8 model’s training and validation losses

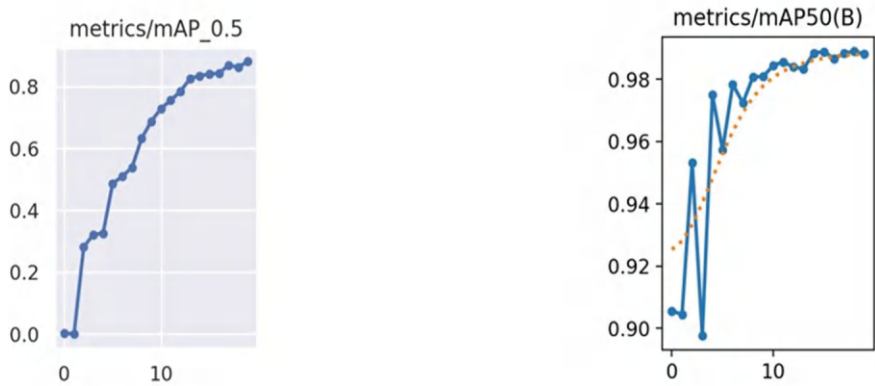


Fig. 6 Represents the value of mAP over 0.5 threshold of IoU for YOLOv5 and YOLOv8 models

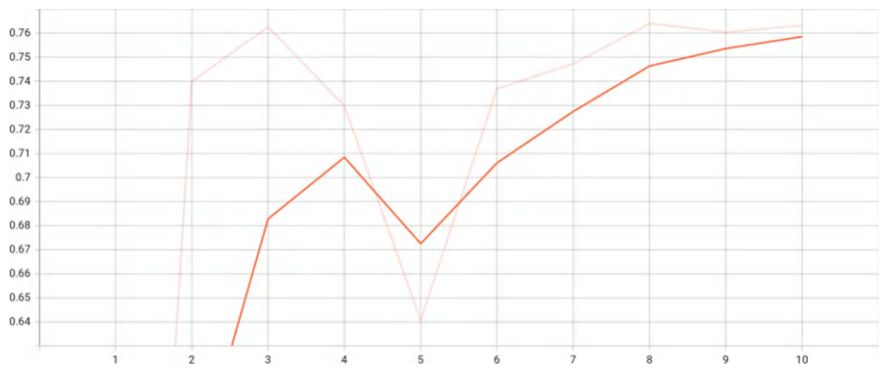


Fig. 7 Represents the value of mAP over 0.5 threshold of IoU for YOLO_NAS model

Table 1 Shows the mAP50 values of each model with YOLOv8 giving the best mAP compared to other models

Model	YOLOv5	YOLO-NAS	YOLOv8
mAP50 (%)	88.1	93.75	98.6

Table 2 Below depicts the performance metrics of the models. YOLOv8 outperforms the other models in terms of precision and recall for individual classes for the leaf disease dataset

Model	YOLOv5		YOLO-NAS		YOLOv8	
	Precision	Recall	Precision	Recall	Precision	Recall
Grape black rot	0.91	0.99	0.90	0.85	0.98	1.0
Grape ESCA	0.93	0.96	0.88	0.89	0.97	0.95
Grape leaf blight	0.89	0.97	0.92	0.87	0.99	0.97
Healthy grape leaf	0.31	0.68	0.91	0.90	0.94	1.0
Healthy potato leaf	0.98	0.83	0.89	0.92	0.95	0.99
Potato early blight	0.82	0.95	0.87	0.91	0.98	0.89
Potato late blight	0.61	0.91	0.93	0.88	0.92	0.98

References

1. Mousavi S, Farahani G (2022) A novel enhanced VGG16 model to tackle grapevine leaves diseases with automatic method. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=9923940>

2. Benfenati A, Causin P, Stefanello ROG (2023) Unsupervised deep learning techniques for automatic detection of plant diseases: reducing the need of manual labelling of plant images: <https://mathematicsinindustry.springeropen.com/articles/https://doi.org/10.1186/s13362-023-00133-6>

3. Pardede HF, Suryawati E, Zilvan V (2020) Plant diseases detection with low resolution data using nested skip connections: <https://journalofbigdata.springeropen.com/articles/https://doi.org/10.1186/s40537-020-00332-7>

4. Haque MdE, Rahman A et al (2022) Rice leaf disease classification and detection using YOLOv5: <https://arxiv.org/pdf/2209.01579.pdf>
5. Chen Z, lin Y et al (2022) Plant disease recognition model based on improved YOLOv5: <https://www.mdpi.com/2073-4395/12/2/365>
6. Panigrahi KP, Sahoo AK, Das H (2020) A CNN approach for corn leaves disease detection to support digital agricultural system. <https://ieeexplore.ieee.org/abstract/document/9142871>
7. Casas E, Bendek E, Rivas-Echeverría F (2023) Assessing the effectiveness of YOLO architectures for smoke and wildfire detection. <https://ieeexplore.ieee.org/abstract/document/10239390>
8. Khvedchenya E (2023) YOLO-NAS by Deci achieves SOTA performance on object detection using neural architecture search. <https://deci.ai/blog/yolo-nas-object-detection-foundation-model/object-detection-foundation-model/>
9. Alessandrini M, Calero Fuentes Rivera R, Falaschetti L, Pau D, Tomaselli V, Turchetti C (2021) A grapevine leaves dataset for early detection and classification of ESCA disease in vineyards through machine learning. <https://www.sciencedirect.com/science/article/pii/S2352340921000937>
10. Terven JR, Cordova-Esparza DM. A comprehensive review of yolo: from yolov1 and beyond. <https://arxiv.org/pdf/2304.00501.pdf>

A Meticulous Evaluation of Musculoskeletal Fracture Diagnosis and Categorization Methods



Shweta Dhareshwar , Manjula Gururaj Rao ,
and Piyush Kumar Pareek

Abstract Bone fractures are common in humans and can happen from a minor mishap or from extreme pressure being placed on the bone. Because of this, the right diagnosis of a fragmented femur is essential in the therapeutic area. Radiograph are used in this work to examine bone fractures. This paper attempts to develop an approach based on the image processing that can use information from x-ray images to classify bone fractures quickly and reliably. The goal of this present investigation initiative is to progress an efficient image processing system that uses data from an x-ray imagery to classify cracks promptly and reliably in bones. Prompt and precise evaluation of bone fractures is crucial for effective treatment, as fractures can result from a variety of illnesses and injuries. Numerous machine learning methods have been especially developed for bone fracture detection, including Naïve Bayes, Decision Tree, Nearest Neighbors, Random Forest, and SVM. The study utilized a range of algorithms, including Naïve Bayes, Decision Tree, Nearest Neighbors, Random Forest, and SVM, whose accuracy measures ranged from 0.64 to 0.92. This study's SVM accuracy was shown to be the greatest statistically, outperforming the majority of the analyzed studies.

Keywords Bone fracture · Machine learning · X-ray · Fracture detection · Preprocessing · Deep learning

S. Dhareshwar (✉)

Department of Master of Computer Applications, Nitte Meenakshi Institute of Technology,
Yelahanka, Bengaluru, Karnataka, India
e-mail: shwetashine.dhareshwar@gmail.com

Research Scholar, Nitte University (Deemed to Be University), Deralakatte, Mangalore,
Karnataka, India

M. G. Rao

Department of Information Science & Engineering, NMAMIT, Nitte, Karkala, Karnataka, India

P. K. Pareek

Department of AI&ML, Nitte Meenakshi Institute of Technology, Yelahanka, Bengaluru,
Karnataka, India
e-mail: piyush.kumar@nmit.ac.in

1 Introduction

Skeleton is one of the main components of the human body. These skeletons consist of bones. It gives the body structure and assistance while shielding internal organs. The bone injuries are common. This refers to bone fracture. Typically, the X-ray films are used to detect fractures. X-ray images are cost-effective and present in most of the hospitals. Regardless of this, the number of cases is rising. Chances may arise of misguiding the fracture. Due to the number of increased fractures and non-fractured cases doctors will have a lot of stress. This research work will allow the practitioners to distinguish fractured and non-fractured images. And also help them tell how severe the fracture is, and where the fracture is present. The doctors can concentrate on the fractured patients, rather than spend the time with non-fractured images.

Medical image processing is a fascinating and evolving field of study. To carry out an examination procedure in the biological sciences study images plays a vital part. The various mechanism in the X-ray images are: X-ray, Computed Tomography, US-Ultrasound, Optical Surface Scanning (OSS), Ifluorescence Endoscopy, Magnetic Resonance Imaging, etc. Content-Based Image Retrieval (CBIR) Data is produced from many sources. The raw data collected is named as information. The information can be either image or text. Accessing the text content is simple when comparing to the images. Content-based automatic detection and classification of bone fractures through artificial intelligence represents a promising advancement within the health-care industry. By combining the strength of expert system and computer vision with medical imaging, this technology can enhance the efficiency, accuracy, and accessibility of fracture diagnosis, ultimately improving patient care and outcomes. Bone fractures are common injuries that significantly impact the quality of life for individuals of all ages. Accurate and timely diagnosis of fractures is crucial for appropriate medical intervention and effective patient care.

Traditionally, radiologists have relied on visual analysis of remedial descriptions, such as X-rays, to identify and categories breaks. On the other hand, this procedure can take some time to complete, subject due to mistakes made by humans, and may not always be readily available in emergency situations. Content-based automatic detection and classification using AI techniques possess the capacity to transform the way ruptures are diagnosed and managed. This technology leverages expert systems algorithms, deep learning models, and computer vision to analyze medical images and provide rapid and accurate assessments of bone fractures.

2 Literature Survey

The literature review provides a concise overview of the critical aspects related to bone fracture severity. It highlights how important it is to comprehend fracture severity in order to guarantee proper diagnosis, suitable care, and the best possible outcomes for patients. The severity of bone fractures can be influenced by a variety of

factors, which collectively determine the extent of damage, potential complications, and the appropriate treatment approach.

Abbas, W et al. [1], 2020, suggested a Faster R-CNN deep learning model for transfer learning that uses the Region Proposal Network (RPN) for fracture classification and detection. It indicates that the proposed strategy is easy to use, productive, and insightful for dynamic fracture analysis and detection. Ahmed et al. [2], 2023, has done a comparative study on a number of intelligent retrieval methods, including Naive Bayes, Decision Tree, Nearest Neighbors, Random Forest have been specially working to handle bone fracture diagnosis. The study utilized a range of algorithms, whose accuracy measures ranged from 0.64 to 0.92. This study's support vector machine accuracy was shown to be the greatest statistically, outperforming many of the analyzed studies. D. P. Yadav and S. Rathor al [3]2020, A Deep Nets model has been created in the present study to establish a distinction between healthy and broken bone. Hardalaç et al. [4] 2022, The purpose of this work is to support medical professionals in the diagnosis of wrist X-ray fractures, especially in emergency services, by performing fracture detection using deep learning on these images. Twenty distinct fracture detection procedures were carried out on the dataset of wrist X-ray images from Gazi University Hospital using deep learning-based object detection models with different backbones, including SABL, RegNet, RetinaNet, PAA, Libra R-CNN, FSAF, Faster R-CNN, Dynamic R-CNN, and DCN. Five distinct ensemble models were created in order to further enhance these processes, and these models were then combined to create the novel detection model known as wrist fracture detection-combo (WFD-C). Indu et al. [5], explains the different forms of fractures, methods for detecting bone fractures, and categorization schemes. Joshi and Singh [6] 2020, numerous methods for fracture detection and classification have been put forth; these methods involve prepping the data, preprocessing the images, and then extracting and classifying the features. Automated fracture classification models could potentially alleviate the laborious and time-consuming process of expert radiologists interpreting and classifying radiographic images. According to numerous researchers cited in this survey, lack of tagged training datasets is the main obstacle to creating a high-performance classification model.

Karanam et al [7], 2023, has conducted a detailed study on researchers constructing models capable of automatically identifying and classifying human bone fractures by providing a basic guidance system. Karanam, Srinivas [8], 2023, has explored the methodology for detecting realignments between the bone organizations using predictive strategies is the main topic of this paper, and it is highlighted here. The results are evaluated using performance measures, and the developed model has produced notable outcomes in the binary and the division of bone fractures into numerous classes. To evaluate how well the confounding array appears proposed model, which is known as the FBGMM. Kuo et al. [9], 2022, presents a meta-analysis and comprehensive evaluation of AI applications in crack recognition. The study aims to evaluate the effectiveness and reliability of neural networks-based methods in identifying fractures from various medical imaging modalities, such as X-rays, CT scans, and MRI. The researchers conduct a comprehensive review of relevant literature and perform a meta-analysis to analyze the collective findings from multiple studies.

The results show that AI-based fracture detection systems exhibit high accuracy and sensitivity, indicating their potential as valuable tools in assisting healthcare professionals with fracture diagnosis. The paper concludes by highlighting the promising future of AI in fracture detection and its potential to improve patient care by expediting diagnosis and treatment. Pflüger et al. [10], (2022), evaluates different ankle fracture classification systems in a sample of 193 trimalleolar ankle fractures. The study aims to assess the reliability and consistency of these arrangement systems in accurately describing the severity and type of ankle fissures. Several widely used classification systems are compared, and their performance is analyzed based on their ability to provide consistent and reproducible results across multiple observers. The research emphasizes the importance of having a standardized and reliable classification system to guide treatment decisions and improve patient outcomes in ankle fracture cases. The research's findings offer insightful information about the benefits as well as the drawbacks of existing classification systems, helping to inform future developments in ankle fracture management.

Matrangolo et al. [11], (2023), have developed a deep learning-based universal fracture detection CAD system on X-ray images. We create a private fracture detection dataset called MURA-D, which includes 6000 bone X-ray images of various human body parts, and we suggest a novel deep learning-based system to detect universal fractures in the human body. Bagaria et al. [12], (2020), this paper provides a thorough explanation of bones, x-rays, fractures, and the various kinds of fractures. Techniques used in the various fractures are described in detail. Fractures can be categorized based on the various body parts they affect. For instance, upper limb fractures include radial head and neck fractures, olecranon fractures, scaphoid fractures, trapezium fractures, and clavicular fractures. Kapse et al. [13], (2023), describes image processing-based analytical methods for bone fracture identification based on X-ray/CT images. To remove the noise, preprocessing is required. With clever edge detection, edges are located. First, pictures are gathered from Kaggle, then the previously listed procedures are used, and LeNet architecture is used to compute the outcomes. Ninety percent accuracy yields good results, according to the research. Using this technology to locate the fracture region may be very challenging since certain X-ray/CT images have significant noise. Sahin [14], (2023), demonstrate an innovative approach to X-ray image-based broken bone evaluation and labeling by combining IP and ML algorithms. The objectives to automate the process of fracture identification, thereby assisting medical professionals in making faster and more accurate diagnoses. The proposed method involves image preliminary processing to enhance image quality, followed by feature extraction to capture relevant information from the X-ray images. Machine learning algorithms, such as or CNN, are then utilized to divide the fractures into different categories based on the extracted features. Tanzi et al. [15], 2020, has explored a variety of DL algorithms to categorize fractures, attempting to identify the advantages of each approach and draw a generalized plan. In terms of classifying bone fractures, the Shift Invariant or Space Invariant Artificial Neural Networks has produced outcomes that are on par with human performance. Yadav et al. [16], 2022, in order to distinguish between images of a fracture and healthy bone, researchers suggested a novel multi-scale

feature fusion using a convolution neural network (CNN) and an enhanced canny edge method. An innovative two-scale sequential deep learning model is the hybrid scale fracture network. Compared to other cutting-edge deep CNN models, this one is more computationally efficient and accurate in diagnosing bone fractures. Zhang et al. [17], (2023), paper focuses on the automated detection and classification of acute vertebral body fractures using a Convolutional Neural Network on CT scans. The study aims to develop an efficient and accurate system to automatically identify vertebral body fractures and classify them based on their severity. The results demonstrate that the CNN-based approach achieves high accuracy in detecting fractures and effectively categorizes them into different classes, aiding in efficient diagnosis and treatment planning.

3 Methodology

The suggested model has the potential to improve fracture diagnosis in clinical settings, which could lead to better patient outcomes and reduced healthcare costs. Detecting and classifying bone fractures typically involves a combination of medical imaging techniques, machine learning algorithms. X-ray imaging is commonly used in clinical practice to diagnose and assess bone fractures due to its ability to visualize bone structures and detect abnormalities. The architecture of the suggested work is shown in Fig. 1. An x-ray image is first provided as the system's input in the construction as it is displayed.

The modules of the suggested system for locating fractured bones in x-ray visualizations is explained as bellow.

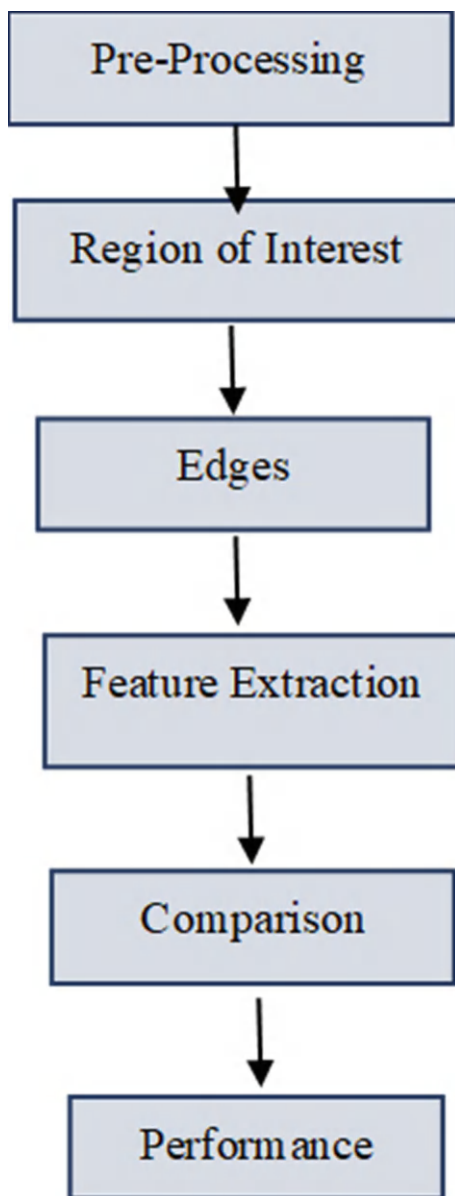
3.1 Preprocessing

The first stage in obtaining a picture with high precision is image preprocessing, which is followed by succeeding phases. The X-ray pictures have been reprocessed to enhance ability and reduce blare. Common preprocessing steps include resizing, normalization, and grayscale conversion.

3.2 Region of Interest (ROI) Extraction

In many cases, the bone of interest might be a small region within the image. ROI extraction methods can help isolate the relevant bone area, reducing computational complexity and focusing the analysis on the critical region.

Fig. 1 Modular schematic for the suggested system



3.3 *Edges*

Edges represent boundaries or transitions between different regions or objects in the image, and they provide crucial details regarding the underlying structures and shapes. Perimeters are crucial for numerous tasks related to component identification,

differentiation, and gathering features in visuals. Detecting edges in an image is often the primary phase in many computer vision algorithms because it helps identify the regions of interest and highlight important features.

3.4 Feature Extraction

Key attributes can be derived from the pre-processed X-ray images to represent the characteristics of fractures. These features can include edge information, texture patterns, and other structural attributes.

3.5 Comparison

Here, hybrid approach is used for combining both traditional methods, such as human visual inspection, with artificial intelligence (AI) techniques to improve the precision and effectiveness, and reliability of breakage recognition. This approach takes advantage of the strengths of both human expertise and AI algorithms to improve performance as a whole in detecting fractures.

3.6 Performance

Performance in fracture detection refers to the accuracy, sensitivity, specificity, and overall effectiveness of a system or method in correctly identifying fractures in medical images, such as X-rays.

4 Observations and Analysis

Machine learning techniques have been used in the development of automated systems for the detection of bone fractures. Fracture detection has demonstrated potential for machine learning methods like decision trees and support vector machines. The analysis is displayed in the Table 1.

SVM is a strong and adaptable machine learning method that may be applied to challenges involving regression as well as classification. It has been used in several distinct areas, such as image classification bone along with the percentage of the fracture in the bone. The k-nearest neighbor had been studied as a part of classifiers for detection of the fracture is studied.

Table 1 Extensive analysis of findings, success rates, and several AI techniques

Problem selected	Method	Observation	Success rate
Identification of skeletal lesions by machine learning techniques. Monitoring transducers	Several machine learning algorithms, such as Naïve Bayes, Decision Tree, Nearest Neighbors, Random Forest, and SVM	SVM accuracy was found to be the highest	Between 64 and 92%
Using X-ray images, image processing and machine learning are used for bone fracture diagnosis and classification	Computer-aided diagnosis system (CAD) and feature extraction and edge detection methods used	When test, training, and accuracy times for classifiers are compared, Linear Discriminant Analysis achieves the highest accuracy rate	88.67% and 89% AUC
Hip fracture automated identification and functional sub classification based on deep learning	A deep learning-based object detection model	After comparing this model's fracture detection accuracy to that of human observers, the model's accuracy was able to classify objects at the expert level	93.7%
A systematic approach for classifying and diagnosing bone fractures from images taken with an X-ray	Compared methodologies like KNN, SVM, RF, InceptionV3, and ResNeXt101	Bone fracture detection and classification system perform well-using ResNeXt101	87.75%
Clinicians diagnose bone fractures by comparing the SIFT algorithm with an intelligent bone fracture detection system	A sift algorithm is used in an intelligence bone fracture detection system to identify bone fractures	IBFDS and neural network	91.67% 83.56%
Blended SFNet model for ML/DL sensors for fractured bones observation and evaluation	The hybrid scale fracture network DL model	SFNet achieved the highest accuracy	73%
Implementing a canny edge algorithm enhances CNN's functionality	BayesNet classifier	BayesNet classifier outperforms every other classifier	86%

5 Results and Discussions

In all, 150 papers were found through the search. 100 abstracts were reviewed after duplicates were removed. Following full-text screening, 69 studies were chosen, leaving 18 unresolved. The utilization of machine learning and image classification methods presents an effective approach for the efficient detection and classification of bone fissure. This research paper explores diverse techniques that can be employed to

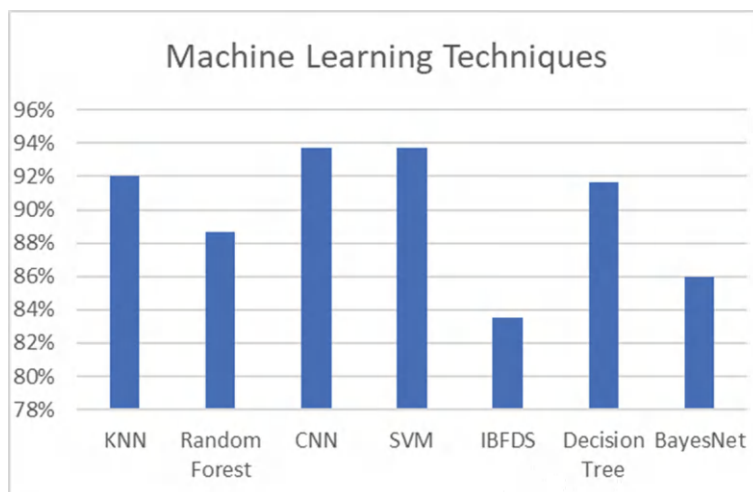


Fig. 2 Graphical representation machine learning techniques

achieve precise finding and procedure of fractures through the application of machine learning and image classification methodologies.

Frequently employed machine learning algorithms encompass LDA, SVM, Artificial Neural Networks, Naïve Bayes classifier, KNN, and deep learning algorithms is depicted in Fig. 2.

Preprocessing is a crucial step in machine learning, including applications in medical imaging such as bone fracture detection. Proper preprocessing can enhance the quality of the input data and improve the performance of machine learning models. It is presented in Table 2.

It is crucial that you keep in mind that the selection of preprocessing techniques be subject to the characteristics of your dataset and the specific requirements of your machine learning model. Experimentation and thorough evaluation are essential to determine the most effective preprocessing pipeline for bone fracture detection.

6 Conclusions

To assist medical professionals, particularly radiologists who frequently order X-rays to assess the type and degree of rupture. For fracture recognition, manual inspection, and conventional X-ray techniques, however, can be laborious and prone to human mistake. Fractures can occasionally be missed by radiologists because they can be confused with other normal pictures, which can cause weariness. It is especially important for rural people because there is no treatment for fractures, which could lead to bone dysfunction. The research is really important in the medical area and can benefit patients, physicians, and the healthcare system in several ways. This

Table 2 Different preprocessing techniques used in machine learning

Paper title	Preprocessing techniques used
An overview of methods for detecting fractures	Gaussian filter
Diagnosis and segmentation of lower leg bone fractures for X-ray images using Faster RCNN	Faster-RCNN
Deep learning-based classification of X-ray bone fractures: a foundation for building a trustworthy solution	Scale-invariant feature transform (SIFT)
Bone fracture identification using the edge detection method	Canny Edge Detection, Sobel operator
Automatic bone fracture detection with improved performance with a better filtering and neural network combination	BPNN with Canny edge detection
An analysis of the way X-ray pictures are used to identify and classify leg bone fractures	Harris corner detection, Decision Tree (DT), K th Nearest Neighbor
A review of finding and grading of limb mandible fractures	Unsharp Masking (USM)

study attempted to investigate the use of medical imaging in the identification and categorization of bone fractures using ML and DL approaches. Since classification offers a formal framework for comprehending, classifying, and reporting the nature and severity of fractures, it is crucial for detecting fractured bones and analysis. Making the right classification guarantees that the best course of treatment is selected. These innovations have enormous potential to enhance patient care and increase the precision of diagnoses. In conclusion, CNNs can be used to better diagnose fractures and direct treatment by identifying fractures and important fracture features on X-ray images.

References

1. Abbas W, Adnan SM, Javid MA, Majeed F, Ahsan T, Hassan SS (2020) Lower leg bone fracture detection and classification using faster RCNN for X-rays images. In: 2020 IEEE 23rd international multitopic conference (INMIC), pp 1–6. IEEE
2. Ahmed KD, Hawezi R (2023) Detection of bone fracture based on machine learning techniques. *Meas Sens* 27:100723
3. Yadav DP, Rathor S (2020) Bone fracture detection and classification using deep learning approach. In: 2020 International conference on power electronics & IoT applications in renewable energy and its control (PARC), Mathura, India, pp 282–285, <https://doi.org/10.1109/PARC49193.2020.236611>
4. Hardalaç F, Uysal F, Peker O, Çiçeklidağ M, Tolunay T, Tokgöz N, Mert F (2022) Fracture detection in wrist X-ray images using deep learning-based object detection models. *Sensors* 22(3):1285
5. Indu MA, Ms S. Survey on various techniques for bone fracture analysis using image processing methods. *J Educat Rabindrabharati University*. ISSN, 0972-7175

6. Joshi D, Singh TP (2020) A survey of fracture detection techniques in bone X-ray images. *Artif Intell Rev* 53(6):4475–4517
7. Karanam SR, Srinivas Y, Chakravarty S (2023) A systematic review on approach and analysis of bone fracture classification. *Mater Today: Proc* 80:2557–2562
8. Karanam SR, Srinivas Y, Chakravarty S (2023) A statistical model approach based on the Gaussian mixture model for the diagnosis and classification of bone fractures. *Int J Health Manag*, pp 1–12
9. Kuo RY, Harrison C, Curran TA, Jones B, Freethy A, Cussons D, Stewart M, Collins GS, Furniss D (2022) Artificial intelligence in fracture detection: a systematic review and meta-analysis. *Radiology* 304(1):50–62
10. Matrangelo MR, Smimmo A, Vitiello R, De Fazio A, El Motassime A, Noia G, Maccauro G (2023) Predictor of hip fracture type: a systematic review. *Acta bio-medica de l'ateneo Parmense* 94(S2):e2023047–e2023047
11. Pflüger P, Harder F, Müller K, Biberthaler P, Crönlein M (2022) Evaluation of ankle fracture classification systems in 193 trimalleolar ankle fractures. *Eur J Trauma Emerg Surg* 48(5):4181–8
12. Bagaria R, Wadhwani S, Wadhwani AK (2020) Different techniques for identification of a bone fracture in analysis of medical image. In: 2020 IEEE 9th International conference on communication systems and network technologies (CSNT), Gwalior, India, pp 327–332. <https://doi.org/10.1109/CSNT48778.2020.9115760>
13. Kapse R, Daware S, Bawane R, Cholkar T, Bambawale D (2023) Image processing for detecting bone fractures. In: 2023 4th International conference on electronics and sustainable communication systems (ICESC), Coimbatore, India, pp 1297–1302. <https://doi.org/10.1109/ICESC57686.2023.10193303>
14. Sahin ME (2023) Image processing and machine learning-based bone fracture detection and classification using X-ray images. *Int J Imag Sys Technol* 33(3):853–865
15. Tanzi L, Vezzetti E, Moreno R, Moos S (2020) X-ray bone fracture classification using deep learning: a baseline for designing a reliable approach. *Appl Sci* 10(4):1507
16. Yadav DP, Sharma A, Athithan S, Bhola A, Sharma B, Dhaou IB (2022) Hybrid SFNet model for bone fracture detection and classification using ML/DL. *Sensors* 22(15):5823
17. Yao L, Guan X, Song X, Tan Y, Wang C, Jin C, Zhang M (2021) Rib fracture detection system based on deep learning. *Sci Rep* 11(1):23513

Innovative Model of Brain Tumor Classification by Discovering the Potential of ENCO-MSRNet with MRI Imaging



S. Renukadevi, K. Rashmi, K. Asha, K. Ramya, C. S. Soumya, and V. Sunil Kumar

Abstract A brain tumor (BT), which may be benign or malignant, is an abnormal expansion of cells within the brain or central spinal canal. Because these tumors are close to important neurological regions, they may seriously harm an individual's health and impair normal brain function. Timely diagnosis and efficient treatment planning depend on the accurate classification of brain tumors using medical imaging data. This work presents a thorough method that uses cutting-edge methodologies to get better results in the classification of brain tumors. This work is based on the BraTS 2020 dataset, which offers a complex and varied collection of brain imaging data. An Adaptive Weighted Frost Filter (AWFF) is used as a preprocessing step to successfully remove quantum noise, improving the quality of the input images. Then, features are extracted using the Limited Receptive Field Mechanism (LRFM), which makes it possible to capture complex patterns that are essential for tumor categorization. By including a Multi-Scale Residual Network (MSRNet) model, the classification procedure is improved by leveraging the skip and depth connections inherent in residual networks. Using the European Night Crawler Optimization (ENCO) technique, the classification model's hyperparameters are optimized. This metaheuristic approach tunes important parameters in a methodical manner, improving the model's presentation and generalization. The experimental findings demonstrate the effectiveness of the suggested technique with a remarkable 99.8% classification accuracy. The

S. Renukadevi (✉)

Jain University, Bengaluru, Karnataka, India

e-mail: renuka.devi@jainuniversity.ac.in

K. Rashmi · K. Asha · K. Ramya

Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India

C. S. Soumya

RNS Institute of Technology, Bengaluru, Karnataka, India

V. Sunil Kumar

Nitte Meenakshi Institute of Technology, Bengaluru, Karnataka, India

suggested method is superior to existing models, as shown by the results of a comparison, underscoring its promise as a reliable solution for brain tumor classification problems.

Keywords Brain tumor • Adaptive weighted frost filter • Limited receptive field mechanism • Multi-scale residual network • European night crawler optimization

1 Introduction

Advances in medical imaging technology have brought about a complete transformation in the healthcare sector by enabling faster and more accurate diagnosis of a wide range of diseases [1]. One important area that benefits from these advancements is the diagnosis and classification of brain tumors. Brain tumors pose a significant risk to the public's health due to their diverse range of features and potential for catastrophic neurological effects [2]. The early and precise detection of these tumors is vital for prompt intervention and improved patient outcomes. Brain tumors are defined as abnormal tissue growths inside the central spinal canal or brain [3]. They can have benign or malignant origins and come from a variety of cell types. Because of their complex characteristics and the sensitivity of the organ, brain tumors are challenging to diagnose and classify. Traditional diagnosis methods, such as human interpretation of medical images, have accuracy and efficiency. As a result, research into novel technologies to enhance diagnostic abilities has been conducted [4, 5].

The significance of early and accurate brain tumor detection cannot be overstated. Delayed diagnosis may lead to advanced stages of illness, more complex treatments, and a lower chance of success [6]. Brain tumors that go untreated may also have extremely crippling or possibly fatal neurological effects. Novel methods that could expedite the discovery and organization of brain tumors are thus desperately needed in order to allow for prompt medical intervention [7]. Deep learning, a subset of artificial intelligence (AI), has developed into a powerful tool for automatically learn hierarchical features from complex datasets, it is particularly well-suited for tasks like brain tumor identification and classification [8, 9]. Deep learning can process and interpret medical imaging data quickly. CNNs are capable of identifying minute patterns that may indicate tumors [10].

Deep learning increases accuracy and streamlines the brain tumor diagnosis process by reducing the likelihood of human error [11]. These algorithms analyze enormous amounts of imaging data quickly, making it possible to detect anomalies accurately and quickly. Deep learning technology in medicine has the potential to improve patient outcomes by precisely and early identifying brain tumors [12, 13]. To improve the development of efficient diagnostic tools for better patient care, this work explores the application of deep classification.

The major contributions of this paper are;

- By effectively eliminating quantum noise, the AWWF preprocessing step greatly enhances the quality of input photos.
- LRFM is used to extract features, which makes it possible to identify complex patterns that are essential for correctly classifying brain tumors.
- By using the skip and depth connections that are naturally present in residual networks, the Multi-Scale ResNet model improves the classification process and offers a more reliable and accurate framework for the classification of brain tumors. By methodically optimizing important parameters, ENCO for hyperparameter tweaking improves the model's presentation and generalizability.

The remaining sections of the study are organized in the form of shadows: The relevant works are summarized in Sect. 2, the suggested model is briefly explained in Sect. 3, examination of the findings and validation are shown in Sects. 4 and 5 offers the summary and conclusion.

2 Related Works

Talukder et al. [14] presented a unique transfer learning-based DL technique to accurately categorize brain tumors. Their innovative approach combines significant pre-processing with reconstruction of the architecture of transfer learning and its fine-tuning. Several transfer learning techniques are used by them, include DenseNet201, ResNet50V2, and InceptionResNetV2. The research utilized the 3,064 image 'share MRI brain tumor dataset, and the accuracy scores for Xception, ResNet50V2, InceptionResNet2, and DenseNet201 were 99.40%, 99.68%, 99.36%, and 98.72%, respectively. According to their research, ResNet50V2 outperforms other models and has the MRI brain tumor dataset from Figshare had the highest accuracy percentage of 99.68%. As a result, the neurologists and clinicians who treat patients with brain tumors may benefit from their suggested model's ability to quickly and effectively categorize brain tumors.

In a research, Ali et al. [15] proposed wrapper-based learning networks (WBM-DLNet) feature optimization techniques with the goal of using magnetic resonance imaging to identify brain tumors. This research used sixteen pretrained deep learning networks to calculate its characteristics. The marine predator procedure, atom search optimization algorithm (ASOA), Harris hawks optimization procedure, butterfly optimization algorithm, whale optimization algorithm, gray wolf optimization algorithm (GWOA), bat procedure, and firefly algorithm are eight metaheuristic optimization algorithms that were used to evaluate the classification function. Using a selection process, the best deep-learning network was identified. To train the SVM model, all deep features from the best deep learning networks were integrated. A publicly available dataset was utilized to verify the proposed WBM-DLNet method. The classification accuracy was shown to be much higher when features selected using WBM-DLNet were used, as opposed to utilizing the whole collection of deep features.

The best results were obtained by EfficientNet-b0-ASOA, with a classification accuracy of 96.7%. The WBM-DLNet approach's results were also associated with those found in the literature.

Asad et al. [16] used a deep CNN with the stochastic gradient descent (SGD) optimization technique to identify tumors in brain cells. Using the publicly available Kaggle brain tumor dataset, the ResNet-50 model was used to multiclassify brain tumors. For training and testing, the method's accuracy was 99.82% and 99.5%, correspondingly. The experimental result showed that the recommended model worked better than baseline techniques, which is a strong argument for its application to other illnesses.

A lightweight, eight-layered classifier called (MBINet) made use of Hossain, A., et al's Operational Neural Network (Self-ONN) [17]. With an accuracy of 96.97%, MBINet fared better than other models in the classification of reconstructed microwave brain images. The precision of tumor classification was increased by preprocessing and data replenishment. MBINet demonstrated tumor detection potential inside a Sensor-based Microwave Brain Imaging (SMBI) scheme.

Brain tumor classification was achieved by Athisayamani et al. [18] using the Enhanced Chimpanzee Optimization Algorithm for feature selection and Canny Mayfly segmentation. On the Figshare dataset, the recommended method—which combined ResNet-152 with softmax—performed with 98.85% accuracy better than other approaches.

After doing research, Özkaraca et al. [19] developed a new modular deep learning model that retains the compensations of common transfer learning approaches for identifying MR images while eliminating their disadvantages. The Kaggle database's open-source images of brain tumors were utilized. For training purposes, two distinct splitting methods were used to the model. Initially, 80% of the MRI image dataset. The second technique was cross-validation ten times. The proposed transfer learning methods outperformed each other in classification on the same MRI dataset, however processing times rose.

2.1 Research Gaps

There is a notable research gap in the investigation of model generalization and robustness across diverse datasets, despite the notable progress made in brain tumor classification through the use of various techniques. Talukder et al. [14], Ali et al. [15], Asad et al. [16], Hossain et al. [17], Athisayamani et al. [18] and Özkaraca et al. [19] are among the studies that ignore the assessment of model adaptability to various data sources in favor of performance metrics on particular datasets. Further comprehensive studies that take into account variations in image quality, acquisition protocols, and demographic factors are necessary to determine the robustness and generalizability of these models. Closing these gaps could result in a more comprehensive understanding of the advantages and disadvantages of various strategies, which would make it easier to develop more reliable and broadly applicable brain tumor classification models.

3 Proposed Methodology

The proposed movement of the work is illustrated in Fig. 1 with the connections between the techniques that are used in this paper.

3.1 BraTS 2020 Dataset

In this study, we test the effectiveness of the projected network on the BraTS 2020 dataset [20]. There are 369 studies used for training, 125 for validation, and 169 for testing with multi-modal MR of the brain. As can be seen in Fig. 2, all studies incorporate not only T1-weighted (T1) but also post-contrast T1-weighted sequences. All of the MR pictures are exactly 240 by 240 by 155 pixels in size. The enhancing tumor (ET), the all annotated by specialists for each investigation. Annotations from public for online review and the final classification competition, whereas annotations from validation and test trials are kept secret.



Fig. 1 Block diagram of the projected work

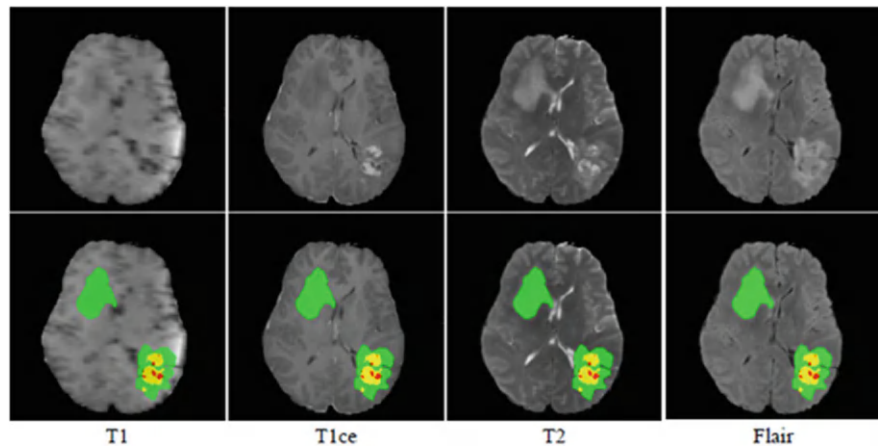


Fig. 2 Ground truth MRI scans and other modalities. The ED region is shown in green, the NET region in red, and the ET region in yellow

3.2 AFFF Preprocessing

Inconsistent photon impacts on the screen are the root source of quantum noise. This noise is typically created during image X-Ray photons [21]. Due to its signal-dependent character, noise is best represented by the Poisson distribution. The degree of noise is inversely related to the degree of unevenness in the quantum strike. If the number of shared quanta per pixel is sufficiently large, the electronic noise can be described by a Gaussian model. Quantum noise impacts every pixel in an image, but its effects are subtle and hard to see with the naked eye. In this research, the quantum noise in MRI scans of brain tumors is removed using a modified Frost filter. The frost filter is a well-known noise reduction filter. As shown in Eq. 1, the CV (coefficient of variation) of the present filter is defined as the ratio of the local standard deviation (SD) to the local mean (mean) of the deformed image. The problem with the current filter is that it does not create the correlation between pixels in an image. There is far less chance of core pixel corruption if neighboring pixels are nearly related. There will be no positive effects from using the filtered image.

$$DN = \sum_{n \times n} K \alpha e^{-\alpha t} \rightarrow (1)$$

$$\text{where } \alpha = \left[\frac{4}{n\bar{\sigma}^2} \right] \left[\frac{\sigma^2}{n\bar{I}^2} \right]$$

DN “Digital Number (values within the moving window)”

K “normalization constant”.

$\bar{I}$ “local weighted mean intended by using Eq. 3”

σ^2 “local variance”.

$\bar{\sigma}$ “image coefficient of dissimilarity charge”

$$|t| = |X - X_o| + |Y - Y_o|, \quad (2)$$

where $|X - X_o|$ mentions to regular charge in x direction and $|Y - Y_o|$ denotes to the difference of size. To fix this, we adjust the Frost filter by using a weighted local average of the distorted or degraded picture. Since the correlation between pixels is crucial, the current filter only utilizes a basic average, but the suggested adaptive weighted filter employs a window. To calculate the weighted average of an empirically selected 55 window, we use Eq. 2. Obviously, the best enhancement results would come from a filter that takes into account the association with the neighboring window. As a result, the suggested modified Adaptive filter that employs a weighted sum of values inside a NN window to replace the center pixel. When the connection between center and residual pixels weakens, the weighting factor similarly lowers and so on. Equation 2 specifies the updated Frost filter in detail for a 55 window size:

Fig. 3 Weighted window for weighted local mean

1	1	1	1	1
1	2	2	2	1
1	2	1	2	1
1	2	2	2	1
1	1	1	1	1

$$\text{weighted\_sum}^k = \sum_{i=-2, i \neq 0}^2 (w^i |\text{PIX}|/5), -2 \leq i \leq 2 \rightarrow (2) \quad (3)$$

where

$$W_i = \begin{cases} 2, & i = -1, +1 \\ 1, & i = -2, +2 \end{cases}$$

The proposed window shown in the Fig. 3 will be used to calculate the weighted mean [22].

In general, closer grayscale values exist between pixels with a smaller lag interval. Accordingly, as shown in Fig. 2, greater emphasis is placed on the pixels with a lag distance of 1, and lesser emphasis is placed on the pixels 2. Since the nearby pixels help well to restoring the core pixel of the damaged image, this strong correlation of related pixels is crucial. As a consequence, the filter window weighted mean has been used instead of the final pixel value.

3.3 LRFM Feature Extraction

In this work, we employ LRFM to features from processed MRI scans. The receptive field represents the exact region represented by picture. The field carries local feature info, whereas the large-size receptive field provides global context info [23]. It has been demonstrated that using a narrower receptive field increases resilience against adversarial patch assaults [24]. It takes its cue from the dual-path network (DPN) [25] to build an LRFM that is resistant to adversarial patches, allowing it to more accurately record the image's local properties while mitigating their negative impacts.

To construct the limited dynamic range of the receptive field, the first of LRFM's four sub-methods employs 11 convolution and 33 dilated convolution with an atrous constant r . The depth-wise convolution is utilized to preserve the local receptive field while extracting local features in the branches. The kk convolution kernel is broken down into smaller $1\ k$ and $k\ 1$ convolution kernels as part of the depth-wise decomposition. To preserve fine-grained feature information, the fourth branch employs 11 convolution, while the skip connection procedure facilitates feature transfer. Another branch feature map is fused using the $\text{concat}()$ and $\text{add}()$ fusion method. Formally, feature be $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$; the first branch is intended as shadows,

$$\mathbf{B}_1 = \mathcal{K}_{3 \times 3, r=2}(\mathcal{K}_{1 \times 1}(\mathbf{I})), \quad (4)$$

where $\mathcal{K}_{1 \times 1}(\cdot)$ and $\mathcal{K}_{3 \times 3}(\cdot)$ are a convolution involving a layer, a ReLU activation function, and an arousal coefficient of $r = 2$. The formulas for the forks are as follows.,

$$\mathbf{B}_2 = \mathcal{K}_{1 \times 1}(\mathcal{K}_{1 \times k}(\mathcal{K}_{k \times 1}(\mathbf{I}))) \quad (5)$$

$$\mathbf{B}_3 = \mathcal{K}_{1 \times 1}(\mathcal{K}_{k \times 1}(\mathcal{K}_{1 \times k}(\mathbf{I}))), \quad (6)$$

where $\mathcal{K}_{1 \times k}(\cdot)$ and $\mathcal{K}_{k \times 1}(\cdot)$ characterize $1 \times k$ and $k \times 1$ convolution. To retain the limited receptive field range, it set $k = 3$. The formula for the fourth fork and feature fusion is as shadows:

$$\mathbf{B}_4 = \mathcal{K}_{1 \times 1}(\mathbf{I}) \quad (7)$$

$$\mathbf{B}_{\text{cat}} = \mathcal{K}_{1 \times 1}(\mathcal{F}_{\text{cat}}(\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)) \quad (8)$$

$$\mathbf{B}_{\text{add}} = \tau(\mathcal{F}_{\text{add}}(\mathbf{B}_{\text{cat}}, \mathbf{B}_4)), \quad (9)$$

where $\mathcal{F}_{\text{cat}}(\cdot)$ characterizes the function, $\mathcal{F}_{\text{add}}(\cdot)$ characterizes function, and $\tau(\cdot)$ signifies the ReLU function.

3.4 MSRNet Classification

In this section, we break down the experimental network's model and architecture [26]. As can be seen in Fig. 4, the perfect consists of a total of 12 layers, all of which can be trained except for the input layer. All convolution that more information of the image can be maintained.

C_1's output feature maps can be concatenated by the operator Concat. Extending to the receptive field, layer. The 11, 33, and 55 convolution kernel sizes are used in

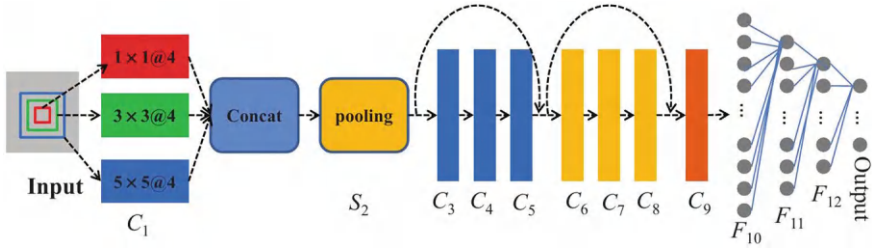


Fig. 4 The overall architecture of the projected MSRNet

the convolution procedure. The resultant feature maps are sent through a rectified linear unit (ReLU) function before being concatenated from multiple convolution modules, each with four kernels.

Maximum pooling layer S_2 consists of 12 feature maps. Due to the non-overlapping nature of the 22 receptive fields, the map in S_2 has half as many rows and columns as the feature map in C_1 . Convolutional layers each use a 33 kernel. The network's classification accuracy improves when it adds two residual blocks and uses more hidden layers. In total, 32 feature maps are generated by the C_9 convolutional layer. Having 120 and 84 units in total, respectively, layers F_{10} and F_{11} are both fully connected. The dropout procedure is used to lessen the possibility of overfitting. Layer F_{12} , the model's final and output layer, is also a fully connected layer. There is a correlation between the number of categories and the sum of neuron units.

3.5 Loss Function

The dice coefficient was chosen as the loss function due to the extreme variation in sample sizes between classes. When determining how similar two sets of data are, i.e., when the label is coefficient is employed.

$$D = 1 - \frac{2|X \cap Y|}{|X| + |Y|} \quad (10)$$

The network predictions p_i , which have k dimensions, go layer that returns the likelihood that each pixel belongs to a specific class. k represents the total sum of categories. It suggests a goal function based on the dice constant. This is the definition of the loss function:

$$L = 1 - \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}, \quad (11)$$

where p_i is the output slash and g_i is the score. N attitudes for the sum of pixels.

3.5.1 Hyper Parameter Tuning of MSRNet Using ENCO

In this study, we provide European Night Crawler Optimization (ENCO), an approach for hyper parameter tuning of MSRNet inspired by the behavior of real-world European Night Crawlers [27]. The algorithm is based on the reproduction habits of the Crawler. The European night crawler has two distinct reproductive strategies to sustain its population. The crawler has the most outstanding slightly alteration.

The European Night Crawler often has both sex organs, allowing it to reproduce independently and create juveniles. One possible mathematical version of the aforementioned strategy is,

$$En_{i1,j} = En_{\text{maximum } j} + E_{\text{minimum } j} - \alpha En_{i,j}; \alpha \in, \quad (12)$$

where $En_{i1,j}$ The European Night Crawler's offspring are placed at a distance from their parents based on a formula where the j th element is a factor of.

Cross over process has been realized by since crawler as $P_{En} = 2$ and adolescent crawler as $A_{En} = 1$. Then crawler P_{En1} and P_{En2} are selected by exactly uttered as,

$$P_{En} = \begin{bmatrix} P_{En1} \\ P_{En2} \end{bmatrix} \quad (13)$$

Then “2” off-springs En_{12} and En_{22} are shaped from the “2” parents and it distinct as,

If random > 0.50 , $En_{12,j} = P_{En1,j}$ and $En_{22,j} = P_{En2,j}$.

Or else

$$En_{12,j} = P_{En2,j} \text{ and } En_{22,j} = P_{En1,j} \quad (14)$$

Then En_{i2} is defined as,

$$En_{i2} = \begin{cases} En_{12} & \text{for random for } 1 < 0.50 \\ En_{22} & \text{otherwise} \end{cases} \quad (15)$$

The next generation crawler (En'_i) is defined after the cohort of En_{i1} and En_{i2}

$$En'_i = \beta En_{i1} + (1 - \beta) En_{i2} \quad (16)$$

Through the factor “ β ” proportion between En_{i1} and En_{i2} is attuned by complementary the search successfully.

$$\beta^{ct+1} = \gamma \cdot \beta^{ct} \quad (17)$$

where “ct” is the existing stage $ct = 0$ and $\beta = 1$

Cauchy mutation, here defined as, is incorporated into the process to prevent the solution from getting stuck at a local optimum.

$$\text{weight}_j = \left(\sum_{i=1}^{N \text{ population}} En_{i,j} \right) / N \text{ population} \quad (18)$$

Then the j th element is distinct as

$$En''_{i,j} = En'_{i,j} + \left(\text{weight}_j = \left(\sum_{i=1}^{N \text{ population}} En_{i,j} \right) / N \text{ population} \right) * G \quad (19)$$

where “ G ” is the random sum and it worn from the delivery $\tau = 1$ where τ scale parameter.

Algorithm 1: ENCO pseudocode.

1. Start
2. Setting the parameters in motion
3. Thirdly, cultivate a thriving European Night Crawler population
4. Evaluation of the European Night Crawler Population's Feasibility and Illusion
5. Value the objective function (5th step)
6. Create the progeny via reproduction 10: Make the progeny by incorporating the cross over procedure
7. Combine the relative importance of two progeny to acquire the modern European night crawler
8. Cauchy-mutate the modern European night crawler to save the species and ensure its survival for future generations
9. The viability of a modern populace is evaluated (13)
10. Continue from Step 9 through Step 13 until the desired population size is reached
11. Do step “6” 16 again, except this time replace the bad people with the best people
12. Repeat “4”, “5”, and “6” for step 17
13. If you don't get the best result by trying and trying again from “7” to “8,” then you aren't doing it right
14. End
Setting the parameters in motion
Thirdly, cultivate a thriving Crawler population
Evaluation of the European Night Crawler Population's Feasibility and Illusion

4 Results and Discussions

4.1 Experimental Arrangement

The following is a list of features and capabilities of the system: System requirements include a computer with Windows 11 (64-bit), the VSCode code editor, the Tensorflow-1.15.0 + keras-2.1.5 deep learning framework, the *Cudnn*7.4.15 + *CUDA*10.0 computing architecture, the NVIDIA GeForce RTX 3050 Ti Laptop GPU for output, the AMD Radeon (TM) Graphics processor, 64 GB of RAM, and the OpenCV-python 4.5.5.62 vision processing language.

4.2 Performance Metrics

The researchers used a number of evaluation criteria, including accuracy (AC), F1-score, precision (PR), sensitivity (SE), specificity (SP), recall (RE), and confusion matrix (CM). It is possible to get these metrics by perfect to a dataset containing 613 MRIs and numeration the True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN). While FP represents cases where non-tumors were mistakenly classified as tumors, TP represents tumor occurrences that the model accurately recognized and named. Unrecognized tumors are those that the diagnostic procedure overlooked. True Negatives (TN) are those that occurred exactly as predicted. Each model's sensitivity, specificity, accuracy, precision, and F1-score are calculated using Eqs. (20)–(24) to assess its overall performance.

$$PR = \frac{TP}{TP + FP} \quad (20)$$

$$RE = \frac{TP}{TP + FP} \quad (21)$$

$$SP = \frac{TN}{TN + FP} \quad (22)$$

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \quad (23)$$

$$F1 - score = \frac{2(TP)}{2(TP) + FP + FN} \quad (24)$$

From Table 1 and Fig. 5, in the training phase, the comparative analysis of several models for brain tumor classification, different designs perform at different levels. LeNet-5 produces an F1-score of around 86% and an accuracy of 86.8% with balanced recall, precision, and specificity. AlexNet achieves an F1-score of 87.7% with a slightly better accuracy of 87.4% while keeping a steady balance in precision, recall, and specificity. GoogLeNet shows much more improvement, with an F1-score of almost 88% and an accuracy of 88.9% with noteworthy precision, recall, and specificity. With notable precision, high recall, specificity, and an F1-score of 89.6%, VGGNet maintains its upward trend and achieves an accuracy of 89.9%. With an accuracy of 93.4%, the Proposed ENCO-MSRNet model performs very well, outperforming all other designs. With an outstanding F1-score of 92.7%, this model demonstrates exceptional accuracy, recall, specificity, and efficacy, highlighting its promise as a reliable solution for brain tumor classification tasks. The findings demonstrate the gradual gains made possible by more complex architectures, with the suggested ENCO-MSRNet model serving as a potential example of this development.

From Table 2 and Fig. 6, a new Proposed ENCO-MSRNet model was compared with four well-known designs, namely LeNet-5, AlexNet, GoogLeNet, and VGGNet, in this study comparing several CNN models for a testing phase of brain tumor classification. The assessment metrics provide a thorough picture of the models' performance and include accuracy, precision, recall, specificity, and F1-score. LeNet-5 achieved a balanced F1-score of 96.7% with an impressive accuracy of 96.8%, along with good precision, recall, and specificity of 96.9%, 96.7%, and 96.7%, respectively. With an accuracy of 96.9% and noteworthy precision, recall, and specificity scores of 97.2%, 97.5%, and 97.4%, respectively, AlexNet showed a small improvement in overall performance. GoogLeNet showed even more progress, with a 97.4% accuracy rate along with excellent recall, precision, and specificity scores of 97.9%, 97.7%, and 97.6%. VGGNet achieved an F1-score of 97.9% with an impressive accurateness of 97.6%, along with exceptional precision and specificity of 98.1% and 97.6%, respectively. The Proposed ENCO-MSRNet model achieved an amazing accuracy

Table 1 Training analysis of the projected model

Models	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1-score (%)
LeNet-5	86.8	86.2	86.7	86.3	86.4
AlexNet	87.4	87.5	87.3	87.4	87.7
GoogLeNet	88.9	88.6	88.9	88.2	88.7
VGGNet	89.9	90.4	91.4	89.8	89.6
Proposed ENCO-MSRNet model	93.4	92.6	93.1	92.5	92.7

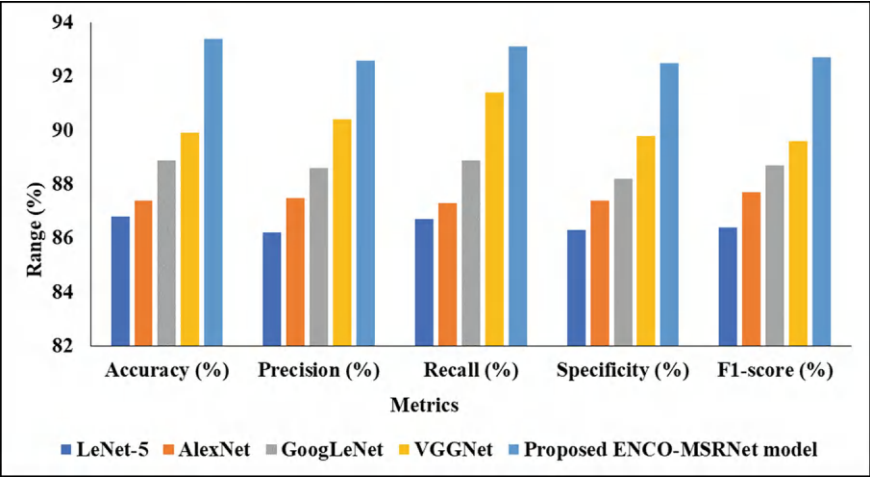


Fig. 5 Training validation of MSRNet model

Table 2 Testing analysis of the projected perfect

Models	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1-score (%)
LeNet-5	96.8	96.9	96.7	96.7	96.7
AlexNet	96.9	97.2	97.5	97.4	96.3
GoogLeNet	97.4	97.9	97.7	97.4	97.6
VGGNet	97.6	98.1	98.0	97.6	97.9
Proposed ENCO-MSRNet model	99.8	99.4	99.3	99.4	99.5

of 99.8%, outperforming all of its predecessors. Strong discrimination skills are shown by its accuracy, recall, and specificity scores of 99.4%, 99.3%, and 99.4%, respectively. The remarkable balance between accuracy and recall shown by the suggested model’s F1-score of 99.5% highlights its effectiveness in the assessed task. This thorough comparison highlights the proposed ENCO-MSRNet model’s enhanced performance and positions it as a potential development in the field of convolutional neural networks for the brain tumor classification.

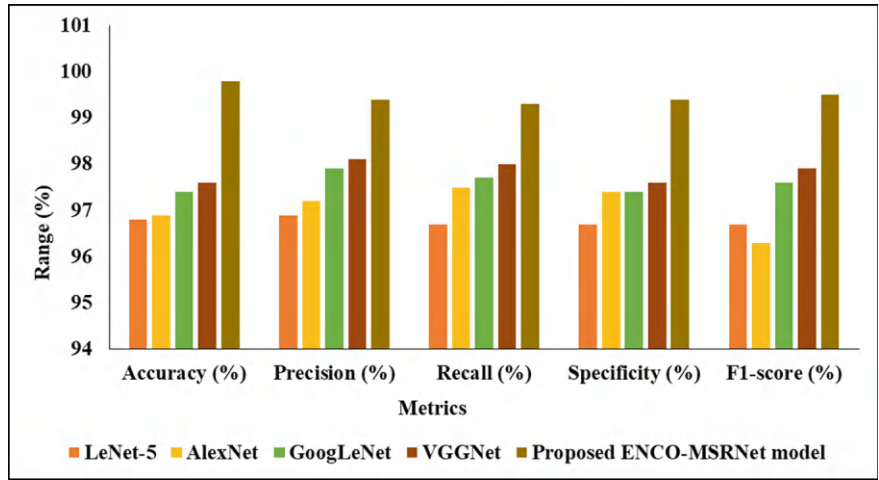


Fig. 6 Testing validation of the MSRNet model

5 Conclusion

Finally, this work presents a thorough and novel strategy to brain tumor categorization that makes use of cutting-edge techniques to provide excellent outcomes. A strong classification system is based use of the BraTS 2020 dataset, which comprises a variety of intricate brain imaging data. Adaptive Weighted Frost Filter (AWFF) preprocessing effectively reduces quantum noise and improves the quality of input pictures, which is important for precise tumor classification. The mechanism known as the Limited Receptive Field Mechanism (LRFM) is helpful in identifying complex patterns that are necessary for classifying tumors, which helps the model identify various tumor kinds. By using the skip and depth connections included in residual networks, the Multi-Scale Residual network (MSRNet) model significantly enhances the classification process and strengthens the model’s overall performance. By means of the European Night Crawler Optimization (ENCO) strategy, hyperparameters are systematically optimized, leading to a methodical improvement in the model’s performance and generalization abilities. The experimental consequences validate the effectiveness of the suggested technique, obtaining a remarkable 98.8% classification accuracy. Analyses in comparison with current models validate the superiority of the proposed strategy, highlighting its potential as a dependable solution for classification problems involving brain tumors. Future deep learning systems will be more compact, and this paper will use an improved feature fusion procedure for tumor detection.

References

1. Deepak S, Ameer PM (2019) Brain tumor classification using deep CNN features via transfer learning. *Comput Biol Med* 111:103345
2. Ayadi W, Elhamzi W, Charfi I, Atri M (2021) Deep CNN for brain tumor classification. *Neural Process Lett* 53:671–700
3. Das S, Aranya ORR, Labiba NN (2019) Brain tumor classification using convolutional neural network. In: 2019 1st International conference on advances in science, engineering and robotics technology (ICASERT), pp 1–5. IEEE
4. Swati ZNK, Zhao Q, Kabir M, Ali F, Ali Z, Ahmed S, Lu J (2019) Brain tumor classification for MR images using transfer learning and fine-tuning. *Comput Med Imag Graph* 75:34–46
5. Kang J, Ullah Z, Gwak J (2021) Mri-based brain tumor classification using ensemble of deep features and machine learning classifiers. *Sensors* 21(6):2222
6. Raza A, Ayub H, Khan JA, Ahmad I, S. Salama A, Daradkeh YI, Hamam H (2022) A hybrid deep learning-based approach for brain tumor classification. *Electronics* 11(7):1146
7. Abiwinanda N, Hanif M, Hesaputra ST, Handayani A, Mengko TR (2019) Brain tumor classification using convolutional neural network. In: World congress on medical physics and biomedical engineering 2018: June 3–8, 2018, Prague, Czech Republic vol 1, pp 183–189. Springer Singapore
8. Ahmad I, Liu Y, Javeed D, Ahmad S (2020) A decision-making technique for solving order allocation problem using a genetic algorithm. *IOP Conf Ser Mater Sci Eng* 853:012054
9. Nikam RD, Lee J, Choi W, Banerjee W, Kwak M, Yadav M, Hwang H (2021) Ionic sieving through one-atom-thick 2D material enables analog nonvolatile memory for neuromorphic computing. *Adv Electron Mater* 17:2100142
10. Ye F, Yang J (2021) A deep neural network model for speaker identification. *Appl Sci* 11:3603
11. Panda B, Panda CS (2019) A review on brain tumor classification methodologies. *Int J Sci Res Sci Technol*, pp 346–359
12. Kumar RL, Kakarla J, Isunuri BV, Singh M (2021) Multi-class brain tumor classification using residual network and global average pooling. *Multimed Tools Appl* 80:13429–13438
13. Kaplan K, Kaya Y, Kuncan M, Ertunç HM (2020) Brain tumor classification using modified local binary patterns (LBP) feature extraction methods. *Med Hypotheses* 139:109696
14. Talukder MA, Islam MM, Uddin MA, Akhter A, Pramanik MAJ, Aryal S, Almoyad MAA, Hasan KF, Moni MA (2023) An efficient deep learning model to categorize brain tumor using reconstruction and fine-tuning. *Expert Syst Appl*, p 120534
15. Ali MU, Hussain SJ, Zafar A, Bhutta MR, Lee SW (2023) WBM-DLNet: wrapper-based metaheuristic deep learning networks feature optimization for enhancing brain tumor detection. *Bio Eng* 10(4):475
16. Asad R, Rehman SU, Imran A, Li J, Almuhaimeed A, Alzahrani A (2023) Computer-aided early melanoma brain-tumor detection using deep-learning approach. *Biomedicines* 11(1):184
17. Hossain A, Islam MT, Abdul Rahim SK, Rahman MA, Rahman T, Arshad H, Khandakar A, Ayari MA, Chowdhury ME (2023) A lightweight deep learning based microwave brain image network model for brain tumor classification using reconstructed microwave brain (RMB) images. *Biosensors* 13(2):238
18. Athisayamani S, Antonyswamy RS, Sarveshwaran V, Almeshari M, Alzamil Y, Ravi V (2023) Feature extraction using a residual deep convolutional neural network (ResNet-152) and optimized feature dimension reduction for MRI brain tumor classification. *Diagnostics* 13(4):668
19. Özkaraca O, Bağrıaçık Oİ, Gürüler H, Khan F, Hussain J, Khan J, Laila UE (2023) Multiple brain tumor classification with dense CNN architecture using brain MRI images. *Life* 13(2):349
20. Saeed MU, Ali G, Bin W, Almotiri SH, AlGhamdi MA, Nagra AA, Amin RU (2021) RMU-net: a novel residual mobile U-net model for brain tumor segmentation from MR images. *Electronics* 10(16):1962

21. Dharmalingam M, Sathyamoorthy S (2021) Adaptive frost filtered quantile regressive artificial deep structure learning framework for image quality assessment. In: Computer networks and inventive communication technologies: proceedings of third ICCNCT 2020, pp 317–336. Springer Singapore
22. Widiastuti S, Omer HS, Mohsen E, Evgenievich DA, Abed JM, Hasan JA, Turki JA (2022) Noise reduction and mammography image segmentation optimization with novel QIMFT-SSA method. *Компьютерная оптика* 46(2):298–307
23. Chen X, Li Z, Jiang J, Han Z, Deng S, Li Z, Fang T, Huo H, Li Q, Liu M (2021) Adaptive effective receptive field convolution for semantic segmentation of VHR remote sensing images. *IEEE Trans Geosci Remote Sens* 59:3532–3546
24. Xiang C, Bhagoji AN, Sehwal V, Mittal P (2021) PatchGuard: a provably robust defense against adversarial patches via small receptive fields and masking. *arXiv* 2021, [arXiv:2005.10884](https://arxiv.org/abs/2005.10884)
25. Jiang K, Wang Z, Yi P, Lu T, Jiang J, Xiong Z (2022) Dual-path deep fusion network for face image hallucination. *IEEE Trans Neural Netw Learn Syst* 33:378–391
26. Wang ZY, Xia QM, Yan JW, Xuan SQ, Su JH, Yang CF (2019) Hyperspectral image classification based on spectral and spatial information using multi-scale ResNet. *Appl Sci* 9(22):4890
27. Kanagasabai L (2021) Real power loss reduction by blue noddly and European night crawler optimization algorithms. *Informatica* 45(6)

Intelligent Application for Dog Breed Identification Using Deep Learning and Machine Learning



Pradnya Narkhede , Anuradha Thakare , Ayush Dongardive, Sakshi Shitole, and Vrushali Guldagad

Abstract This research study addresses the pivotal task of precise dog breed identification, a crucial pursuit with far-reaching implications, including disease control, genetics research, and personalized care strategies. A sophisticated dog breed identification system is introduced that harnesses the power of Convolutional Neural Networks (CNNs), such as InceptionV3, VGG16, Xception, and ResNet, for robust feature extraction. The extracted features are subsequently subjected to classification via the Support Vector Machine (SVM) algorithm. These CNN models are meticulously trained on a diverse and comprehensive dataset known as the Stanford Dogs Dataset, which comprises a wide array of dog images. This dataset facilitates effective feature extraction and enables our system to classify diverse dog breeds accurately. Through iterative training, our model adeptly learns intricate, breed-specific patterns embedded in the images, culminating in an impressive classification accuracy of 98.8%. This research not only advances the state of the art in dog breed identification but also offers a versatile framework that can be applied to a multitude of real-world scenarios where accurate breed recognition is indispensable. Focusing on precision and versatility, our system opens up new avenues for the broader field of canine research and care, with far-reaching implications for various domains, including biology, veterinary science, and personalized pet management.

P. Narkhede (✉) · A. Thakare · S. Shitole · V. Guldagad
Pimpri Chinchwad College of Engineering, Pune, India
e-mail: pradnya.narkhede@pccoepune.org

A. Thakare
e-mail: anuradha.thakare@pccoepune.org

S. Shitole
e-mail: sakshi.shitole21@pccoepune.org

V. Guldagad
e-mail: vrushali.guldagad22@pccoepune.org

A. Dongardive
Springer Heidelberg, Heidelberg, Germany
e-mail: ayush.dongardive21@pccoepune.org

Keywords Dog breed · CNN · SVM · InceptionV3 · Xception · VGG · Deep learning · Machine learning

1 Introduction

Dogs are now integral to human society, acting as laborers, friends, and even therapists [1]. Accurately recognizing dog breeds is crucial in many situations because there are more than 120 breeds, each with unique physical qualities and behaviors. This involves population control, veterinary clinics, animal shelters, and dog research. The ability to accurately identify dog breeds can have significant consequences. Modern solutions are required since traditional dog breed identification relies on human evaluation that is vulnerable to bias. Convolutional neural networks (CNNs), a subclass of deep learning, are excellent at classifying images and have the potential to revolutionize breed identification through accuracy and objectivity.

The primary challenge is establishing reliable automated dog breed identification. Current methods, reliant on manual assessment and predefined rules, struggle to encompass breed diversity. This challenge extends beyond practical implications, with accurate breed identification impacting disease management, genetics research, animal welfare, and legal aspects. Progress can optimize shelters, customize veterinary care, and encourage responsible breeding practices.

2 Literature Survey

A method for classifying dog breeds using convolutional neural networks (CNNs) was proposed [2] and [11]. The authors use the TensorFlow and Keras frameworks to train a CNN on a dataset of dog images in [2]. CNN is able to achieve an accuracy of 88.4% on the test set. This paper is a good introduction to the use of CNNs for dog breed classification. It provides a detailed overview of the methodology and the results. A deep learning approach to identify dog breeds [3] uses transfer learning by retaining pre-trained CNNs. It is basically an image classification model that achieves a promising accuracy of 89.92% on the published dataset with 133 dog breeds. This paper is a good example of how transfer learning can be used to improve the performance of a CNN for dog breed classification. The authors show that by retaining a pre-trained CNN, they can achieve a higher accuracy than if they had trained the CNN from scratch.

Furthermore, a Convolutional Neural Network and transfer learning model was used in [4] to create an android application that uses picture analysis to determine the breed of a dog. The user of the Android application can click or upload a dog photo. After that, the picture is preprocessed to extract the features needed for testing. With CNN and transfer learning, dog breed predictions are made with 94% accuracy. In [5] the main principle was to find a deep learning approach to identify dog breeds. Its

main outcome was an image classification model which has an accuracy of 89.92% on a dataset of 133 dog breeds. The authors show that ResNet50 is a good choice for dog breed classification because it is able to learn more complex features from the images.

An interesting research on implementing an Image classification model with Deep learning focusing Dog breed was proposed [6] which identifies the breed of the dog from an image by using CNN. The authors show that their model is able to learn the features of dog breeds from the images and classify them with high accuracy. Among deep learning approach to identify various breeds of dogs, InceptionV3 is a good choice [7] for dog breed classification since it is able to learn more complex features from the images.

A method for classifying dog breeds using the size and position of each local part of the dog [8] was proposed. The authors use a CNN to extract features from the images and then use a classifier to classify the dog breeds based on the features. The model is able to achieve an accuracy of 90.6% on the test set. An android application to classify dog breeds using a deep learning approach [9] can be a best utility for dog lovers. In this CNN architecture is used to achieve the accurate results similar to the VGGNet, ResNet50, and InceptionV3 architectures. The authors show that Xception is a good choice for dog breed classification because it is able to learn more complex features from the images.

A Dog Breed Classification model Based on Deep Learning with transfer learning by retaining pertained CNNs was proposed in [10] and [13]. The authors use transfer learning to train the model. It start with a pre-trained CNN, such as ResNet50, Inception-ResNet-V2 and then fine-tune the model on the dataset of dog images. This allows the model to learn from the features that have already been learned by the pre-trained CNN. In [12], the authors used YOLOv7 object detection algorithm for real-time dog breed classification. Table 1 depicts comparative analysis of various approaches for dog breed identification.

3 Existing Methods for Dog Breed Identification

Existing methods and technologies for dog breed identification encompass a range of approaches that leverage various types of data, including visual, genetic, and behavioral information. Here are some of the common methods and technologies used for identifying dog breeds:

3.1 Visual Identification

1. **Visual Inspection:** This is the simplest and most traditional method, relying on the visual assessment of a dog's physical characteristics, such as size, coat color,

Table 1 Comparative analysis of existing approaches for dog breed identification

Research paper	Algorithms/Methods	Strength	Limitations	Percentage accuracy	Dataset size
Dog breed identification and age detection using neural networks	CNN, KNN and SVM	Achieved optimal performance	Less accuracy for age detection	97.8%	90 breeds
Dog Breed identification using pre-trained models	ResNet50,	Achieved good accuracy using pre-trained Res Net 50 model	System specification are not sufficient to work on complete dataset	82.7%	133 Dog breeds
Hybrid deep learning Algorithms for dog breed identification—a comparative analysis	Xception, VGG19, EfficientNetV2, Inception, ResNet152V2, EfficientNetV2M, NASNetMobile	Comparative study gives better understanding to apply algorithm	Due to time and CPU limitations, many iterations were not able to conduct	Xception- 91.9%, VGG19- 55%, EfficientNetV2M- 89.05%, NASNetMobile-83.47%, ResNet152V2-87.38%, Hybrid of Inception-v3 &Xception—92.4%	120 breeds
Dog identification using soft biometrics and neural networks	Deep Neural Network, CNN Xception model	Provide good idea based on soft biometrics		90.80% and 91.29%	120 Dogs breed
Review paper on dog breed classification using convolutional neural network	CNN, VGG16	Achieved good Accuracy		–	180 dog breeds

(continued)

Table 1 (continued)

Research paper	Algorithms/Methods	Strength	Limitations	Percentage accuracy	Dataset size
Dog breed identification using deep learning	Inception-ResNet-v2 and NASNet-A	The paper presents a detailed exploration of the use of convolutional neural networks for fine-grained image recognition, specifically in determining the breed of a dog in a given image		NASNet-A mobile—80.72% Inception-ResNet-v2-90.69%	120 different breeds of dogs, with a varying number of images per breed
Dog Breed classifier for facial recognition using Convolutional Neural Networks	PCA, CNN	Experiments performed on large dataset of images and achieved an accuracy of 95.5% using the CNN algorithm	Lack of discussion on the specific challenges faced in the classification of dog breeds using CNN	95.5%	13,233 human images and 8351 dog images from 103 classes
Dog breed Identification using convolutional neural networks on Android	VGGNet, ResNet50, Xception and Inception-Resnet V2	Predicts the breed of a dog by uploading or clicking an image	Not efficient with images captured from another image	Xception-93%, Inception-94%, validation data-2000 images and test data-9381 images	Train data- 9199 images, validation data-2000 images and test data-9381 images
Dog breed identification with fine tuning of pre-trained models	VGG16, Xception, InceptionV3, Logistic Regression, Transfer learning	It provides better solution for different image classification and identification problems.	Out of 120 breed only 16 breeds images were covered	VGG16 - 91% Xception - 93% InceptionV3-94%	1400 images and 16 breeds

(continued)

Table 1 (continued)

Research paper	Algorithms/Methods	Strength	Limitations	Percentage accuracy	Dataset size
Dog breed identification using deep learning	Inception-ResNet-v2 and NASNet-A	The paper presents a detailed exploration of the use of convolutional neural networks for fine-grained image recognition, specifically in determining the breed of a dog in a given image		NASNet-A mobile—80.72% Inception-ResNet-v2—90.69%	120 different breeds of dogs, with a varying number of images per breed
Dog breed classifier for facial recognition using convolutional neural networks	PCA, CNN	The authors have conducted experiments with a large dataset of images and achieved an accuracy of 95.5% using the CNN algorithm	Lack of discussion on the specific challenges faced in the classification of dog breeds using convolutional neural networks	95.5%	13,233 human images and 8351 dog images from 103 classes
Dog breed identification using convolutional neural networks on Android	VGGNet, ResNet50, Xception and Inception-Resnet V2	Predicts the breed of a dog by uploading or clicking an image	Not efficient with images captured from another image	Xception-93%, Inception-94%,	train data has 9199 images, validation data has 2000 images and test data has 9381 images
Dog breed identification with fine tuning of pre-trained models	VGG16, Xception, InceptionV3, Logistic Regression, Transfer learning	It provides better solution for different image classification and identification problems	Out of 120 breed only 16 breeds images were covered	VGG16—91% Xception—93% InceptionV3-94%	1400 images and 16 breeds

- pattern, and body shape. Dog breed standards set by kennel clubs can be used as references.
2. **Photographic Analysis:** Software applications and algorithms have been developed to compare images of dogs with a database of known breeds. These systems analyze various visual features, like coat texture, ear shape, and facial features.
 3. **3D Scanning and Imaging:** Advanced technologies like 3D scanning and imaging can provide a detailed representation of a dog's physical characteristics, enabling a more accurate breed identification.

3.2 Machine Learning and Deep Learning

A variety of machine learning models can be used for dog breed identification. Some of the most popular models include:

Convolutional neural networks (CNNs): CNNs are a type of neural network that is well-suited for image classification tasks. CNNs work by extracting features from images and then using those features to classify the images. There are three layers in a convolutional neural network: input, hidden, and output. Convolutional neural network layers are found in the hidden layers. This usually involves a layer that uses the layer's input matrix to perform a dot product of the convolution kernel. Figure 1 represents basic architecture of CNN model.

Convolutional layer: Convolutional layers are the core of a CNN. They extract features from the input image by applying a series of convolution operations.

Pooling layers: Pooling layers reduce the size of the feature maps output by the convolutional layers. This contributes to the network's decreased parameter count and increased computational efficiency. Max pooling 2D can be used to improve the performance of CNNs for dogs breed identification by reducing the size of the feature maps, which can help to reduce the number of parameters in the CNN and to make it more computationally efficient. Max pooling 2D can also help to prevent overfitting, which is a problem that can occur when a CNN is trained on a large dataset.

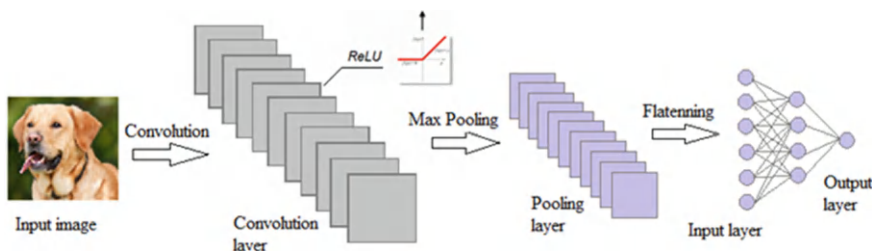


Fig. 1 Basic architecture of a CNN model

Flatten Layer: When identifying dog breeds, a convolutional neural network's (CNN) output is transformed into a one-dimensional vector using a flatten layer. The dog breed is then classified by feeding this vector into a fully linked layer. A multidimensional tensor is the result of a CNN, so the flatten layer is required. This tensor would be too large to be processed by a fully connected layer, which is why it is flattened into a one-dimensional vector.

Support vector machines: SVMs are another type of machine learning model that can be used for classification tasks. SVMs work by finding a hyperplane in the feature space that separates the data points into two classes.

Random forests: Random forests are an ensemble learning model that consists of a collection of decision trees. Random forests work by averaging the predictions of the individual decision trees to produce a final prediction.

Computer Vision and Image Recognition: Machine learning algorithms, such as convolutional neural networks (CNNs), can be trained to recognize dog breeds based on images. These models learn to detect breed-specific patterns and features in photographs.

3.3 Mobile Apps and Online Platforms

Various mobile apps and online platforms are available to help individuals identify dog breeds using photographs. These platforms use pre-trained models or user-generated data to provide breed suggestions.

4 Proposed Intelligent Approach for Dog Breed

In the proposed method, the project undergoes five main phases: Load dataset, data preparation, Generate CNN Features, model training, and testing as shown in Fig. 2. The data preparation phase involves importing and preprocessing images and labels. In The CNN Feature Generation Phase Features will extract from the various Deep learning algorithms like InceptionV3, VGG16, Xception, RasNet, etc. The model training phase encompasses creating, compiling, and training a SVM model on extracted data from deep learning algorithms for breed classification. The testing phase involves selecting an image, preprocessing it, and using the trained model to predict the breed. The original and predicted breed labels are then compared for evaluation.

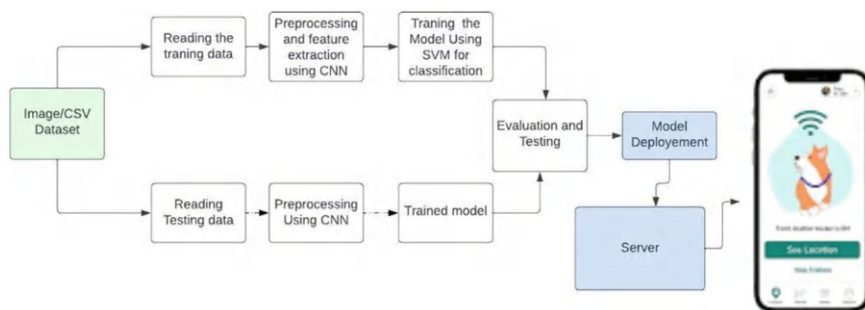


Fig. 2 Block diagram for dog breed classification and display result

4.1 Load Dataset

We used a dataset of Stanford's Dog breed identification from Kaggle. This dataset consists of a training set and a test set of images of dogs. There are 120 various breeds, image data is given with approximately 80–100 pictures of each breed. Each image has a filename that is its unique id. There is one csv file named as label.csv which contains the breed name and image's unique id.

4.2 Data Preparation

Using Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs) together for dog breed identification involves a two-step process: First, training a CNN to extract features from dog images, and then using these features as input to an SVM classifier. Here's a step-by-step guide on how to use CNN and SVM on a dog breed dataset:

Data Collection. Gather a dataset of dog images, ideally labeled with their respective breeds.

Data Preprocessing. Resize the images to a consistent size (e.g., 224×224 pixels) and normalize the pixel values to a common scale.

4.2.1 CNN Feature Extraction

In this section, we outline the critical steps involved in the extraction of informative features from the input images using Convolutional Neural Networks (CNNs). As shown in Fig. 3, the goal is to leverage pre-trained models, namely Xception, ResNet, VGG16, and Inception, to facilitate the extraction of discriminative features for subsequent image classification tasks.

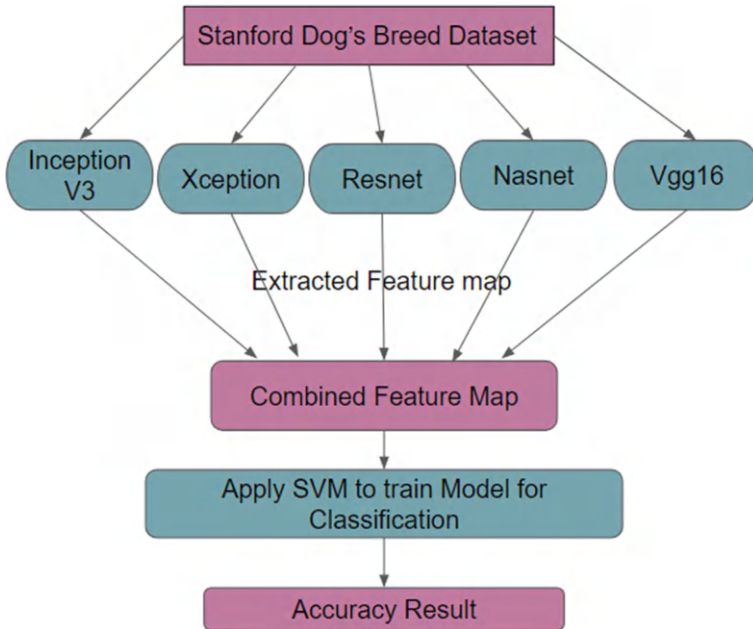


Fig. 3 Workflow diagram for CNN feature extraction model and SVM classification model

4.2.2 Building and Training a CNN

To harness the power of deep learning for image feature extraction, we begin by constructing a CNN architecture. We rely on models pre-trained on large-scale image datasets such as ImageNet for efficiency and effectiveness. Pre-trained models like Xception, ResNet, VGG16, and Inception have shown remarkable performance across various computer vision tasks. Transfer learning allows us to benefit from the extensive knowledge and feature representations these models have learned during their training on vast and diverse image datasets.

4.2.3 Generation of CNN Features

The final stage of our CNN-based feature extraction involves the application of the pre-trained models, namely Xception, ResNet, VGG16, and Inception, to our dataset. We pass each image through the network and capture the output from the last convolutional layer, yielding a feature vector for each image.

Xception Feature Extraction

Xception, a deep convolutional neural network architecture, is employed to extract features from the images. The architecture is designed to approximate an ideal local sparse structure, thereby allowing the utilization of multiple filter sizes within a single image block. This enhances the model's ability to capture complex and diverse image characteristics.

ResNet Feature Extraction

ResNet, a state-of-the-art deep learning architecture, is applied to extract image features. Its deep residual connections enable the effective capture of essential image features, even in cases of very deep networks, contributing to improved feature extraction performance.

VGG16 Feature Extraction

VGG16, a classic CNN architecture, is utilized for feature extraction, offering a balance between model complexity and performance. It excels at capturing both low-level and high-level image details, making it a valuable component in our feature extraction pipeline.

Inception Feature Extraction

Inception, characterized by its innovative inception modules, is integrated into our feature extraction process. These modules allow the incorporation of a diverse range of filter sizes within a single image block, fostering the extraction of a wide spectrum of abstract features critical for image classification.

In summary, the feature extraction process using pre-trained CNN models, including Xception, ResNet, VGG16, and Inception, empowers our system to capture and encode a rich set of discriminative features from input images. These features form the basis for subsequent image classification tasks, contributing to the overall effectiveness of our proposed method.

This structured explanation of the CNN feature extraction process using pre-trained models will help readers understand the methodology and its significance within your research.

4.2.4 SVM Classification

In this section, we outline the methodology for employing Support Vector Machines (SVMs) as a classification technique. SVMs have proven to be powerful tools for

Table 2 Result of feature extraction from various deep learning

Dataset	Feature extraction algorithm	Features map	Final features map (X)	Accuracy after classification model
Stanford dogs image dataset and label [Breed] csv	Inception	(9691, 2048)	(9691, 9664)	98.80
	Xception	(9691, 2048)		
	Vgg16	(9691, 512)		
	NasNet	(9691, 4032)		
	ResNet	(9691, 1536)		

image classification tasks, and our approach comprises three crucial steps: Data Preparation, SVM Classifier Training, and Hyperparameter Tuning.

Prepare Data for SVM

To enable effective image classification using SVMs, we first create feature vectors for all the images in our training and testing datasets by combining features of feature extraction deep learning algorithms as shown in Table 2. These feature vectors serve as essential inputs to the SVM classifier. CNNs have established themselves as efficient feature extractors for image data, allowing us to represent complex visual information compactly and discriminatively.

Train an SVM Classifier

The integration of CNN-based feature extraction with SVM classification, forms a robust framework for image classification that is effective in handling a diverse range of image datasets. This approach is underpinned by the synergy of deep learning and traditional machine learning techniques, harnessing the best of both worlds for accurate and reliable image classification.

5 Result and Discussion

5.1 Dataset Description

The dataset used to perform breed identification is Stanford’s Dogs Breed Identification dataset from Kaggle. In this dataset, there are 120 breeds of dogs with the two sets for training and testing. In the training set, there are 10,224 images present and in testing also 10,220 images are available. We also used the label CSV file to map with the training image dataset.

5.2 Result of Feature Extraction

The features map of various Feature extraction deep learning algorithms are as shown in Table 2. All extracted features are combined together and will be applied to the SVM to train the classification model.

5.3 Validation and Testing

The model is compiled with appropriate loss function, optimizer, and metrics. The dataset is split into training, validation, and testing sets using train test split. The model is trained on the training set, and the training progress is monitored.

Model evaluation. Once the model is trained, it will need to be evaluated on a held-out test set. The test set should be a set of images that were not used to train the model. The accuracy of the model will be calculated on the test set.

Predict Dog Breed. The trained SVM classifier is used to predict the breeds of dogs in your testing dataset. The SVM takes the CNN-extracted feature vectors as input and provides breed predictions.

Evaluation Metrics. The performance of your hybrid CNN-SVM model is assessed using evaluation Metrics Accuracy and log loss.

The trained model is tested on unseen data to evaluate its performance and make predictions. An example image is selected for testing purposes. The selected image is preprocessed to match the input shape expected by the model. The model's predictions are generated for the preprocessed image. The predicted breed label is extracted based on the highest probability output from the model. The original breed label of the image is obtained from the dataset. The original and predicted breed labels are displayed to compare. By applying this model to the existing image dataset from Kaggle, we can achieve an accuracy of 98.80%. The training and testing validation accuracy is represented in Fig. 4.

After the training of the model, some predictions were made on the testing data using the model for breed identification. Figures 5, 6, 7, and 8 show the given outputs of the predictions of the CNN model. In Fig. 5, it is observed that a picture of a Maltese dog is given to the model, and the model identifies the breed correctly. In Fig. 6, it is observed that a picture of Bernese Mountain is given to the model, and the model identifies the breed correctly. Figure 7 shows the predictions of dog breeds. The picture of a Scottish deerhound is given to the model, and the model predicts the image correctly. Then, in Fig. 8, the picture of the entlebucher is predicted correctly using the CNN model.

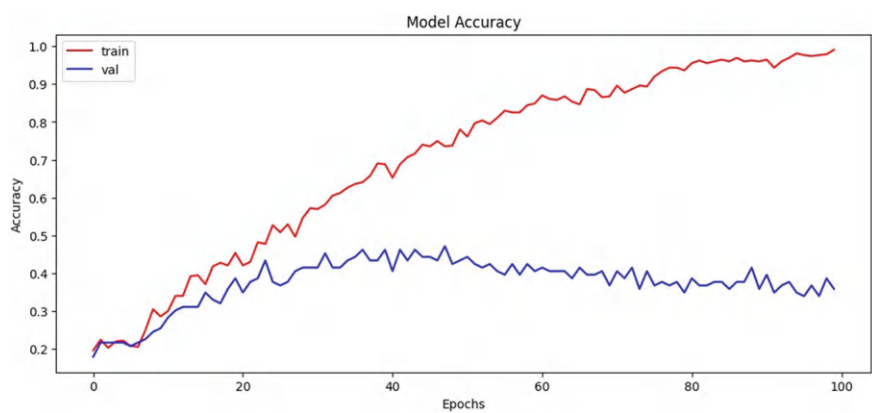


Fig. 4 Training and validation accuracy

Fig. 5 Original dog breed
Maltese dog and predicted
dog breed by our model is
also Maltese

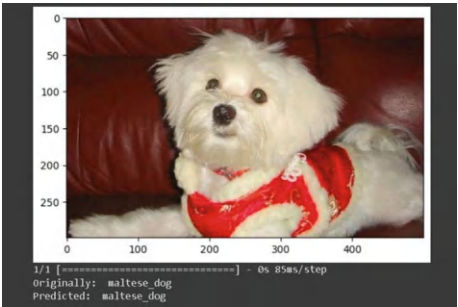


Fig. 6 Original dog breed
Bernese mountain and
predicted dog breed by our
model is Bernese mountain



Fig. 7 Original dog breed Scottish deerhound and predicted dog breed by our model is also Scottish deerhound

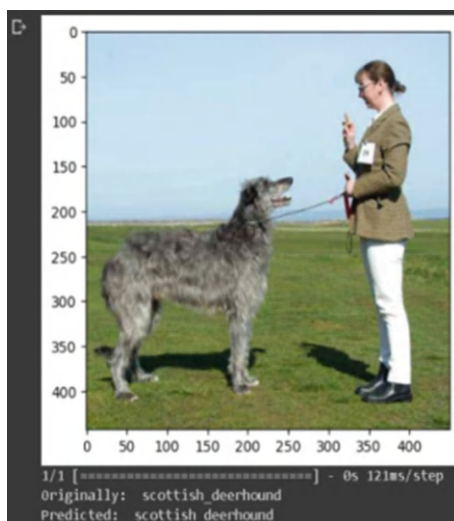
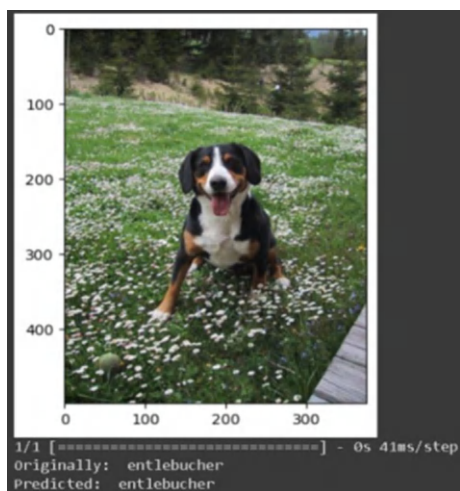


Fig. 8 Original dog breed entlebucher and predicted dog breed by our model is also entlebucher



6 Conclusion

In this proposed method, CNN model with SVM classification algorithm will help in identifying the various breeds of dogs on the basis of their images. By applying this model to the existing image dataset from Kaggle we can achieve accuracy of 98.80%. In this dataset, there are 120 breeds of image data. Feature extractions by various CNN applications are performed on this data to achieve more accuracy. The SVM Classification is applied on combined features of various CNN Models. The main purpose behind this project is to improve animal welfare, veterinary Care, Behavioral

training, provide nutritious and safe food and proper medical treatment. In Future we are planning to develop an application which provides breed accuracy, emotion detection, and medical services at one tip.

References

1. Holland KE, Mead R, Casey RA, Upjohn MM, Christley RM (2022) Why do people want dogs? A mixed-methods study of motivations for dog acquisition in the United Kingdom. *Front Vet Sci* 9:877950
2. Haswar P, Iyer A, Godhane P, Jadhav A, Malhotra A (2022) Dog breed identification and age detection using neural networks. *Int J Innov Res Technol* 8(11):125–131
3. Sadanandan T, Thaha Mohammad SY, Subhash Chandar TJ, Vaidhya GK (2020) Dog breed identification using pre-trained models. *Int J Comput Sci Mob Comput* 9(4):76–82
4. Valarmathi B, Gupta NS, Prakash G, Reddy RH, Saravanan S, Shanmugasundaram P (P.) Hybrid deep learning algorithms for dog breed identification—a comparative analysis. *IEEE Access* 11:77228–77239
5. Lai K, Yanushkevich S, Tu X (2019) Dog identification using soft biometrics and neural networks. In: International joint conference on neural networks (IJCNN) 2019. IEEE Xplore, Hungary
6. Suyash SB, Rishikesh PP, Rohit PW, Kaustubh PJ, Bodke B (2021) Review paper on dog breed classification using convolutional neural network. *Int J Adv Res Sci Commun Technol* 4(2):343–346
7. Ráduly Z, Sulyok C, Zolde A (2018) Dog breed identification using deep learning. In: IEEE 16th international symposium on intelligent systems and informatics 13(15):0271–0276
8. Shah BK, Kumar A, Kumar A (2020) Dog breed classifier for facial recognition using convolutional neural networks. In: Third international conference on intelligent sustainable systems on proceedings. IEEE Xplore, pp 508–513
9. Durga Bhavani D, Quadri MHS, Ram Reddy Y (2019) Dog breed identification using convolutional neural networks on android. *CVR J Sci Technol* 17(8):62–66
10. Vijaya Kumar B, Bhavya K (2019) Dog breed identification with fine tuning of pre-trained models. *Int J Recent Technol Eng* 8(2S11):3677–3680
11. Iyer SC, Shekokar NM (2023) Identifying the optimal deep learning based image recognition technique for dog breed detection. In: 14th international conference on computing communication and networking technologies (ICCCNT), Delhi, India, pp 1–10
12. Sudharson S, Deena S, Reddy SN (2023) Efficient real-time breed classification using YOLOv7 object detection algorithm. In: 14th international conference on computing communication and networking technologies (ICCCNT), Delhi, India, pp 1–6
13. Manivannan S, Venkateswaran N (2023) Dog breed classification using inception-ResNet-V2. In: International conference for advancement in technology (ICONAT), Goa, pp 1–5

A Survey on Machine-Learning Models Used for Detecting Brain Tumors



K. S. Supriya, Anil Kumar , S. Sree Priya, R. Supriya, and K. S. Tejaswini

Abstract Brain tumor is a potentially lethal illness that can significantly lower a patient's quality of life and obstruct regular brain function. Early diagnosis and identification are essential to successful treatment. Lately, machine learning has image analysis and detecting techniques have become essential function in the medical industry, presenting more precise and reliable performance. In this study, our goal is to review several methods for detecting brain tumors, where we compare several techniques in machine learning models like ResNet50, InceptionV3, VGG16 and DenseNet201 with respect to precision, recall, accuracy and F1-score.

Keywords Brain tumor · CNN · ResNet50 · InceptionV3 · VGG16 and DenseNet201

K. S. Supriya (✉) · S. Sree Priya · R. Supriya · K. S. Tejaswini
Department of ISE, BMS Institute of Technology and Management, VTU, Bengaluru, India
e-mail: 1by20is175@bmsit.in

S. Sree Priya
e-mail: 1by20is164@bmsit.in

R. Supriya
e-mail: 1by20is176@bmsit.in

K. S. Tejaswini
e-mail: 1by20is183@bmsit.in

A. Kumar
Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal
Academy of Higher Education, Manipal, Karnataka, India
e-mail: kumar.anil@manipal.edu

1 Introduction

The International Association of Cancer Registries (IARC) reports that there are more than 28,000 brain tumor cases reported in India each year, and the disease is thought to be the primary cause of over 24,000 fatalities [1]. However, it has proven to be extremely difficult to identify and treat some conditions affecting the central nervous system, such as brain tumors. Cognitive health is seriously at risk from brain tumors, which are collections of abnormal cells growing inside the limited space of the brain. They are a major cause of mortality for both adults and children, and their frequency has been rising throughout the world.

An abnormal mass of tissue in which cells grow and replicate uncontrollably, seemingly unchecked by the processes that control normal cells, is called an intracranial tumor, or brain tumor. Brain tumor is classified as Glioma, Meningioma, Pituitary [2]. Conventional techniques for diagnosing brain tumors, such MRIs and CT scans, can be expensive and time-consuming and mostly depend on the expertise of radiologists [3]. It is required to have a quicker and more automated approach to detect a brain tumor. We examine the outputs of different CNN models in an attempt to determine which model gives better results in terms of generating trustworthy and accurate findings for brain tumor diagnosis in order to solve the issue of brain tumor identification.

The precise identification of brain tumors is critical in the medical image analysis field. This study conducted a thorough survey and analysis of four well-known Convolutional Neural Networks (CNN) models: ResNet [4], VGG16 [5], Inception V3 [6], and MobileNet, in order to address this pressing healthcare issue. We assessed these models' ability to identify brain cancers using recall, accuracy, precision, and F1-score, among other metrics. The study used a dataset representative of real-world scenarios to determine which CNN model performed best for this life-saving task. The findings provide insightful information for the medical community.

This paper will give a brief information about the various machine learning models used in detecting brain tumors and it also specifies various evaluation metrics used to evaluate the machine learning algorithms. It helps and give a clear idea for the researchers to choose the machine learning algorithms before doing the research in the similar area (Table 1).

Table 1 Confusion matrix [10]

TP (True Positive)	FN (False Negative)
FP (False Positive)	TN (True Negative)

2 Literature Survey

Surveys conducted in the past have revealed that Convolution Neural Networks (CNN) [7] performed better than when compared to other machine learning models in terms of outputs [8]. The other various models in machine learning includes ResNet50, InceptionV3, VGG16, DenseNet201 [9]. This survey is assessed utilizing 4 kinds of factual measurements such as precision, accuracy, recall, and F1-score in two different tables, i.e., Tables 2 and 3. The above four mentioned measurements can be calculated with the help of confusion matrix as shown in Table 1.

Accuracy: The ratio of patients who were successfully classified to all patients indicates the model’s accuracy.

$$(TP + TN)/(TP + FN + FP + TN)$$

Table 2 Comparison of different models in machine learning with respect to precision, recall and F1-score

Ref. paper	Machine learning models	Precision	Recall	F1-score
[11]	ResNet50	0.98	1.00	0.99
	InceptionV3	0.98	1.00	0.99
	VGG-16	0.98	1.00	0.99
	MobileNetV2	1.00	1.00	1.00
[10]	ResNet50V2	0.99	0.99	0.99
	DenseNet201	0.98	0.98	0.98
	InceptionResNetV2	0.99	0.99	0.99
	Xception	0.97	0.98	0.97
[12]	ResNet50	0.70	0.68	0.69
	ResNet18	0.64	0.52	0.57
	ResNet101	0.73	0.67	0.70
	DenseNet201	0.73	0.67	0.70
	InceptionResNetV2	0.98	0.99	0.99
	Xception	0.98	0.99	0.98
	nceptionV3I	0.93	0.94	0.93
	MobileNetV2	0.81	0.80	0.80
[13]	ResNet50	0.966	0.968	0.967
	VGG-16	0.974	0.977	0.975
	InceptionV3	0.967	0.971	0.969
	InceptionResnetV2	0.962	0.966	0.964
	Xception	0.957	0.959	0.958
[14]	ResNet50	0.84	0.75	0.75
	ResNet152	0.84	0.78	0.75
	VGG-16	0.82	0.75	0.72
	VGG-19	0.09	0.29	0.13
	DenseNet201	0.84	0.77	0.74
	InceptionV3	0.62	0.60	0.56
	Xception	0.68	0.61	0.57
[15]	InceptionV3	0.98	0.98	0.98

Table 3 Comparison of different models in machine learning with respect to accuracy

[18]	MR images	CNN	88.46
		ResNet50	84.62
		InceptionV3	92.31
[19]	MR images	MobileNet	97.19
		ResNet50	93.60
		InceptionResNet	97.03
[20]	MR images	ResNet50	99.10
		VGG16	96.78
		DenseNet201	97.00
		Xception	98.20
[21]	CT scan	VGG16	93.00
		ResNet50	96.50
		MobileNetV2	97.60
[22]	253 samples (155 malignant and 98 benign)	VGG-16	90.00
		Inception-V3	55.00
		ResNet-50	95.00
[22]	253 samples (155 malignant and 98 benign)	VGG-16	60.00
		ResNet-50	80.00
		Inception-V3	50.00

Precision: The precision of the model is indicated by the ratio of patients with the condition that are correctly classified to the overall number of patients predicted.

$$TP/(TP + FP)$$

Recall: The recall metric of a model is the proportion of patients with the disease who have been correctly diagnosed to all patients with the disease.

$$TP/(TP + FN)$$

F1-score: Score is a measure that strikes a balance between recall and precision; it is also called as the F Measure.

$$(2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall})$$

In the paper [11], The main aim was to classify the brain tumors using transfer learning techniques. The best accurate of four pre-trained models—ResNet50, VGG16, MobileNet-V2, and Inception-V3—was determined via comparison. Without applying augmentation methods, the study also examined the impact of dataset balancing on classification accuracy. After dataset balancing, the outcomes demonstrated that ResNet50 outperformed the other models, with the maximum accuracy rate of 99.06%. Future studies of this paper, should look into the potential benefits of augmentation techniques for raising the resilience and the accuracy

of models. Furthermore, techniques like Generative Adversarial Networks (GANs), active learning, and deep learning architectures which are advanced can be taken into consideration to improve accuracy.

In the paper [10], they used the Brain Tumor Image dataset to demonstrate the high diagnostic accuracy of their suggested model for brain tumors. For brain tumor classification, Xception scored 98.40%, ResNet50V4 scored 99.68%, InceptionResNetV2 scored 99.36%, and DenseNet201 scored 98.72%. Subsequent analysis showed that ResNet50V2 performs more precisely when compared to other models and current techniques.

In the paper [12], Inceptionresnetv2 yielded the best results when it came to classifying brain tumors. Inceptionresnetv2, which has an accuracy rate of 98.71%, is a reliable technique for classification of brain tumor. The best model (Inceptionresnetv2) has been shown to be superior to other hybrid approaches that used DL models for deep feature extraction and SVM for brain tumor classification, with a 98.91% classification accuracy. The fact that many more models have not yet been investigated and that data models can be enhanced even more by training them on larger data sets is one of the paper's limitations, despite the fact that they compared various models.

In the paper [13], Using the available data, they obtained the accuracy of 99.5% in their analysis, which has achieved the proposed methodology. Their work uses five different convolutional models and transfer learning architectures to address the research focus on identification of brain tumor in medical imaging. Still, there's space for more research and advancement in this area. Future research should aim to address the complexity of real-world diagnostic challenges by incorporating a more diverse and clinically relevant set of brain lesions.

In the study [14], on a dataset of 3264 MRI images with (tumor types) and without tumors, pre-trained VGG-16, VGG-19, ResNet-50, ResNet152, DenseNet-201, Xception, and Inception V3 models are used to classify brain tumors. ResNet-50 outperformed the others with 84% precision, 75% F1 score, 75% recall, and 80% accuracy. The VGG-19 model, in contrast, has the lowest accuracy rate. This works future enhancement is to classify and analyze the MR images more thoroughly.

In the paper [15], comparison to related works, their proposed model demonstrated the best classification accuracy with minimal preprocessing of the data. When assessed based on additional performance metrics, their suggested model outperformed every other state-of-the-art method on the same dataset by a significant margin. The following is the graphical representation of the comparison of various machine learning models in terms of recall, F1-Score, accuracy and precision. In Fig. 1, when four distinct convolutional neural networks (CNN) models are compared, MobileNet comes out on top with 89% accuracy, 90% precision, 88% recall, and 90% F1-score. DenseNet comes in second with an F1-score of 87%, recall of 88%, accuracy of 88%, and precision of 85%. With a high recall of 89%, accuracy of 91%, precision of 90%, and F1-score of 88%, VGG19 performs admirably. With the highest accuracy (93%), precision (94%), recall (92%), and F1-score (92%), ResNet performs better than the others. ResNet is the most suitable model among

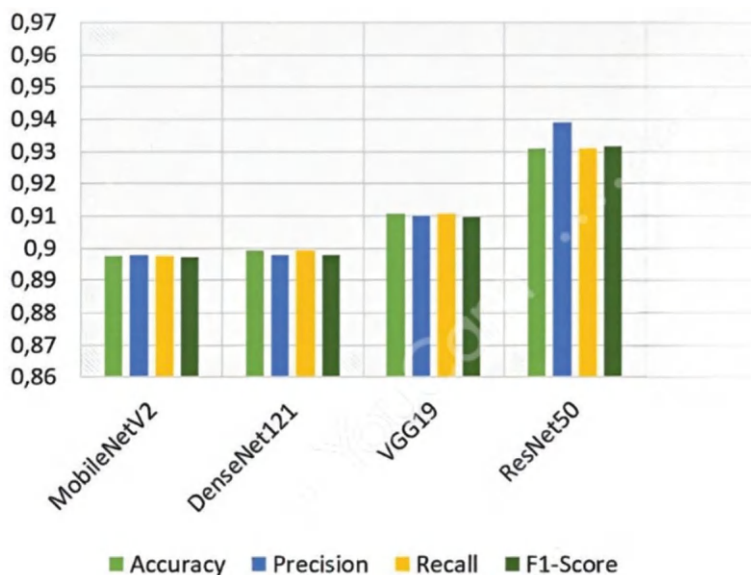


Fig. 1 Comparison of different CNN models [16]

the ones mentioned because it exhibits the best overall performance based on these metrics, providing a combination of high precision, accuracy, recall, and F1-score.

With respect to Fig. 2 VGG19 is the best technique for the classification among the four CNN models that were evaluated; it consistently performs better than ResNet50, VGG16, and Inception V3 across multiple critical metrics. With the accuracy of 97%, ResNet50 and VGG19 are the most accurate, followed closely by VGG16 at 95%, and Inception V3 at 80%. Recall is highest for VGG19 (97%), lowest for Inception V3 (70%) and highest for ResNet50 (90%). VGG16 (96%), and VGG19 (97%), are the closest competitors. Precision is another area where VGG19 shines, coming in at 97%, closely followed by ResNet50 and VGG16 at 95%, and Inception V3 at 65%. VGG19's superiority is further cemented by the F1-score, which strikes a balance between precision and recall, coming in at 96%, closely followed by VGG16 at 95%, ResNet50 at 93%, and Inception V3 at 77%.

In paper [18], they considered Br35H and MR Images datasets with accuracy for various models as shown in the Table 3. They also mentioned about how the cost-sensitive application of these models yielded higher accuracy than standard one for unbalanced datasets. A promising accuracy of 92.31% and a recall value of 1 are shown by the cost-sensitive InceptionV3 (CS-InceptionV3) and CNN (CS-CNN). However, more study is required to improve the accuracy of various models and make them appropriate for use in the real world.

In the paper [19], the discussed model is assessed with various training sizes. 99.6% classification accuracy is recorded by the suggested model. Considering the Adam optimizer is used, the suggested model learns quickly and avoids overfitting

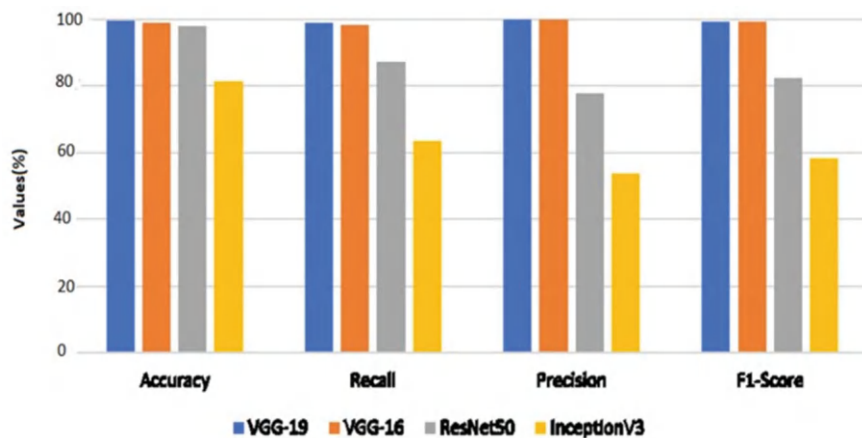


Fig. 2 Comparison of accuracy, recall, precision and F1-score of CNN models [17]

issues. Additionally, the suggested model performs comparably on noisy picture data. Their next research will concentrate on cutting the suggested model's parameter count and computation time without sacrificing functionality.

In paper [20], The authors discussed the proposed model in which they used brain tumor MR image data and integrated data augmentation and transfer learning techniques to classify various tumor types, including meningioma, glioma, and pituitary. In future, they plan to diagnose brain tumors using additional datasets and deep learning techniques.

In paper [22], Finding the appropriate algorithm in the field of deep learning for small datasets is the aim of this work. They considered 253 samples (155 malignant and 98 benign) as datasets where ResNet50 showed the highest accuracy. The accuracy scores of the VGG-16, Inception-V3, and MobileNet-V2 are 79.00%, 84.00%, and 86.00%, respectively, by considering the test results. This project's main goal is to address the dataset's small size, which prevents enough learning from the ostensibly huge number of image samples. Nevertheless, MobileNet-V2 is found to be the best of the three algorithms in managing these constraints. The most beneficial features will be chosen for this work in future by applying a few well-known feature selection methods. The feature selection methods have the potential to enhance the DL algorithm's performance.

Regarding Fig. 3, the accuracy scores of several CNN models that are used to identify brain tumors are shown in the information. ResNet has the lowest accuracy among the options at 65%, Inception V3 is the highest at 55%, VGG 16 is the lowest at 85%, VGG 19 is the highest at 95%, and DenseNet is the lowest at 90%. The accuracy metric, which indicates how well the model can classify brain tumors, is a crucial consideration when assessing the performance of these models. With a 95% accuracy rate, VGG 19 is clearly the best option in this situation. This implies that,

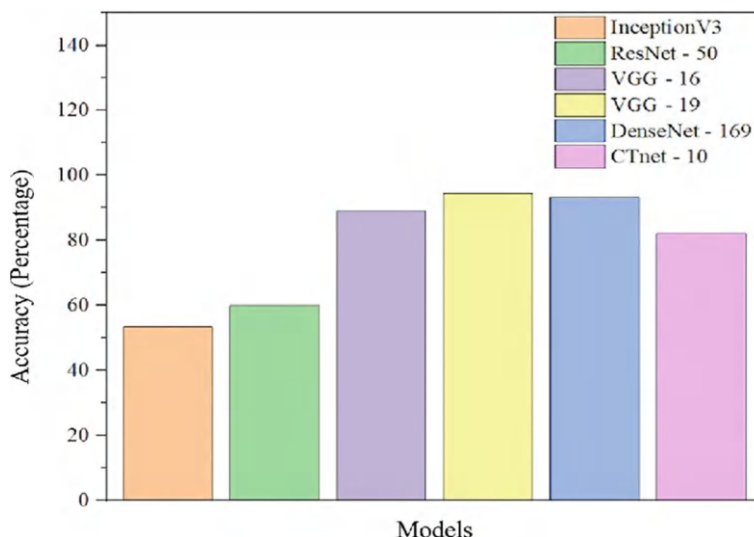


Fig. 3 Comparison of accuracy scores of different CNN models [23]

of the architectures listed, VGG 19 is probably the most trustworthy for detecting brain tumors.

The given chart or Fig. 4 shows a line plot illustrating the efficacy of several Convolutional Neural Network (CNN) models in the context of brain tumor detection: ResNet, VGG16, Inception V3, and MobileNet. It tracks the accuracy of these models for both training and testing data over multiple training epochs. The accuracy values, which span from 0 to 1, can be plotted against the number of epochs to show how capable the models are of learning and generalization. This diagram functions as a vital performance evaluation tool, offering information on how well the models identify brain tumor photos and how they change over training. It draws attention to the models' capacity to identify relevant features in the data—a crucial component of medical imaging analysis.

Figure 5 illustrates how well Convolutional Neural Networks (CNN) models—ResNet, VGG16, Inception V3, and MobileNet—perform when it comes to brain tumor detection. The number of training epochs is plotted on the x-axis between 0 and 40, and the accuracy values are plotted on the y-axis between 0.70 and 1.00. These accuracy trends are significant performance indicators that reveal details about the models' ability to learn and generalize. This visual representation is essential for assessing how well the models perform during training and for figuring out how to classify brain tumor images. It demonstrates how the models can recognize and extract significant features from the data, which is critical for medical image analysis.

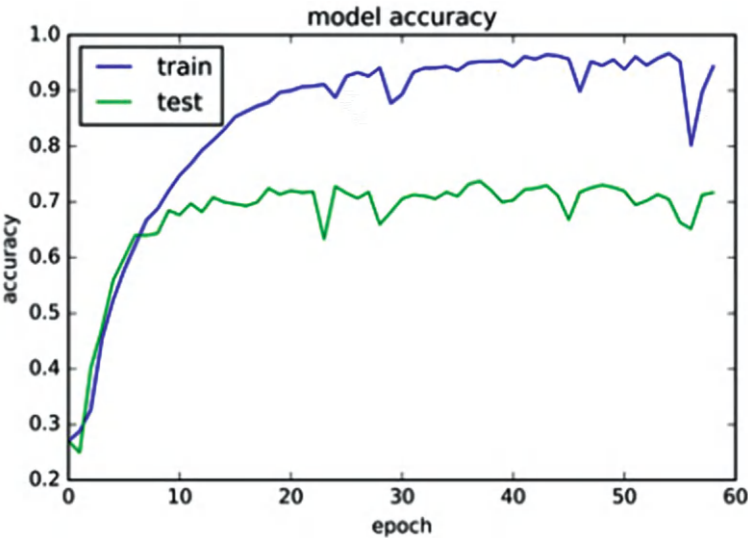


Fig. 4 Efficacy of training and testing data over multiple training epochs [24]

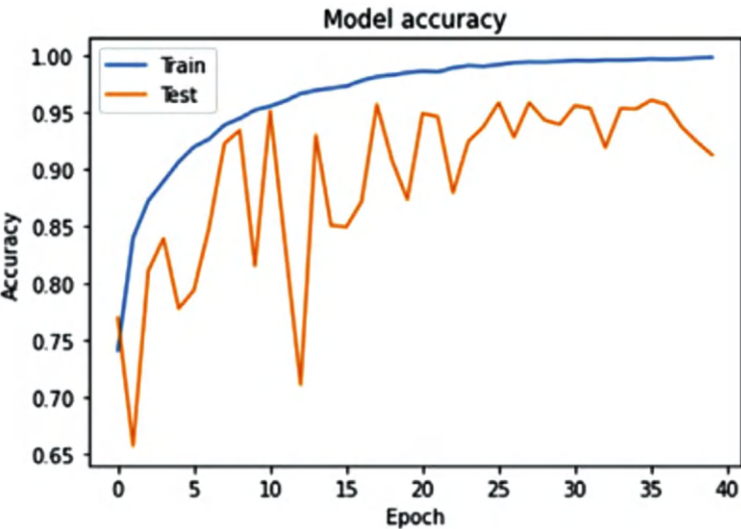


Fig. 5 Performance of CNN models over different epochs [25]

3 Conclusion

Four CNN algorithms—ResNet50V2, InceptionResNetV2, VGG-16, and DenseNet201—were used in this study and thoroughly assessed using a range of metrics, including f1 score, accuracy, recall, and precision. The results displayed impressive accomplishments, with ResNet50V2 emerging as the top performer, outperforming other models and current techniques, especially in terms of precision. Moreover, the study's focus on the appropriateness of convolutional neural networks and various machine learning methods for brain tumor identification produced positive results. The customized CNN outperformed the others with an accuracy of 98.21%. Expanding the study's scope to identify different types of brain tumors is one possible direction for future research, which could result in the creation of more accurate and comprehensive diagnostic tools for the medical imaging industry. Additionally, with the improved accuracy noted in our survey when taking into account all the references, one can concentrate on identifying the tumor's stages that could progress to brain cancer. These advancements may significantly increase patient and physician confidence and trust in the detection of brain tumors.

References

1. Dasgupta A, Gupta T, Jalali R (2016) Indian data on central nervous tumors: a summary of published work. *South Asian J Cancer* 5(3):147
2. Usman K, Rajpoot K (2017) Brain tumor classification from multimodality MRI using wavelets and machine learning. *Pattern Anal Appl* 20:871–881
3. Afshar P, Plataniotis KN, Mohammadi A (2019) Capsule networks for brain tumor classification based on MRI images and coarse tumor boundaries. In: *Proceedings of the ICASSP 2019–2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Brighton, UK, 12–17 May 2019, pp 1368–1372
4. Wang H, Li K, Xu C (2022) A new generation of ResNet model based on artificial intelligence and few data driven and its construction in image recognition model. *Comput Intell Neurosci* 2022(1):5976155
5. Younis A, Qiang L, Nyatega CO, Adamu MJ, Kawuwa HB (2022) Brain tumor analysis using deep learning and VGG-16 ensembling learning approaches. *Appl Sci* 12(14):7282
6. Turk O, Ozhan D, Acar E, Akinci TC, Yilmaz M (2022) Automatic detection of brain tumors with the aid of ensemble deep learning architectures and class activation map indicators by employing magnetic resonance images
7. Khan MSI, Rahman A, Debnath T, Karim MR, Nasir MK, Band SS, Mosavi A, Dehjangig I (2022) Accurate brain tumor detection using deep convolutional neural network. *Comput Struct Biotechnol J* 20:4733–4745
8. Mahmud MI, Mamun M, Abdelgawad A (2023) A deep analysis of brain tumor detection from MR images using deep learning networks. *Algorithms* 16(4):176
9. Huang G, Liu Z, van der Maaten L, Weinberger K (2017) Densely connected convolutional networks. In: *Conference paper-2017*
10. Talukder MA, Islam MM, Uddin MA, Akhter A, Pramanik MAJ, Aryal S, Almoayad MAA, Hasan KF, Moni MA (2023) An efficient deep learning model to categorize brain tumor using reconstruction and fine-tuning. *Expert Syst Appl* 230:120534
11. Masum AKM, Badhon N, Badhon SMSI, Ria NJ, Abujar S, Syed MM, Mahmud N (2023) Comparative evaluation of transfer learning for classification of brain tumor using MRI

12. Ullah N, Khan JA, Khan MS, Khan W, Hassan I, Obayya M, Negm N, Salama AS (2022) An effective approach to detect and identify brain tumors using transfer learning. *Appl Sci* 12(11):5645
13. Abdusalomov AB, Mukhiddinov M, Whangbo TK (2023) Brain tumor detection based on deep learning approaches and magnetic resonance imaging. *Cancers* 15(16):4172
14. Arshed MA, Shahzad A, Arshad K, Karim D, Mumtaz S, Tanveer M (2022) Multiclass brain tumor classification from MRI images using pre trained CNN model. *VFAST Trans Softw Eng* 22–28
15. Soumik MFI, Hossain MA (2020) Brain tumor classification with Inception network based deep learning model using transfer learning. In: 2020 IEEE region 10 symposium (TENSYP), Dhaka, Bangladesh
16. Akinyelu AA, Zaccagna F, Grist JT, Castelli M, Rundo L (2023) Brain tumor diagnosis using machine learning, convolutional neural networks, capsule neural networks and vision transformers, applied to MRI: a survey. *J Imaging* 8(8):205
17. Aslam A, Khan E, Beg MMS, Chanchlani A, Chaudhari M (2022) Brain tumor detection and segmentation by intensity adjustment. *J Med Syst* 43(8):282
18. Shawon MTR, Shibli GM, Ahmed F, Joy SKS (2023) Explainable cost-sensitive deep neural networks for brain tumor detection from brain MRI images considering data imbalance
19. Kokkalla S, Kakarla J, Venkateswarlu IB, Singh M (2021) Three-class brain tumor classification using deep dense inception residual network
20. ul Haq A, Li JP, Khan S, Alshara MA, Alotaibi RM, Mawuli CB (2022) Deep learning approach for classification of brain tumors using MRI data in IoT healthcare environment. *Sci Rep* 12(1):15331
21. Dawood NM, Abou El-Magd LM, Abdel-Aty A-H, Awad WS, Elgarayhi A, Sallah M (2023) Brain tumors detection using computed tomography scans based on deep neural networks. *Inf Sci Lett* 12(4)
22. Zailan ZNI, Mostafa SA, Abdulmaged AI, Baharum Z, Jaber MM, Hidayat R (2022) Deep learning approach for prediction of brain tumor from small number of MRI images. *Int J Inform Vis* 6(2–2):581–586
23. Vidyavihar S, Shah J, Mehendale N (2022) Diagnosis of brain tumor using CT scan images and machine learning learning techniques
24. Gaur L, Zhao Z, Bhandari M, Razdan T, Mallik S (2023) Explanation-driven deep learning model for prediction of brain tumour status using MRI image data. *Front Genet* 13:822666
25. Shivdikar A, Shirke M, Vodnala I, Upadhaya J (2022) Brain tumor detection using deep learning 2022

An Efficient Swarm Optimized Based Machine-Learning Technique for Improving the Prediction of Chronic Diseases



Sandeep Kumar Hegde, Rajalaxmi Hegde, and Thangavel Murugan

Abstract Heart Disease is considered as one of the deadliest chronic disease across the globe. In spite of the advancement in the healthcare technology, the mortality rate due to the cardiovascular disease increasing every day. The heart disease will claim at least 18 million lives each year. The manual based treatment to diseased patient by physician is one of the solution to reduce the mortality rate. But one of the drawback of this approach is, it will consume more time as sign of the disease will not be known in advance. In order to overcome with these problems in the proposed paper machine learning based algorithmic model has been suggested to early predict the chronic disease. In the proposed paper machine learning algorithms such as Random Forest, Decision Tree, SVM, and Naive Bayes classifier and stacked classifier is applied on the heart disease dataset to early predict the disease. The performance of the machine learning algorithms are further optimized using proposed Hybrid Dragonfly-Genetic Algorithm. Further the performance of the suggested model is subsequently validated by using the various performance metrics. The experimental results shows that the combination of Hybrid Dragonfly-Genetic Algorithm and stacked classifier attained higher accuracy compared to the existing algorithms.

Keywords Dragonfly · Machine learning · Naive Bayes · Chronic diseases · SVM

S. K. Hegde · R. Hegde (✉)

NITTE (Deemed to be University), Department of Computer Science and Engineering, NMAM Institute of Technology, Nitte, Karnataka, India
e-mail: laxmi123.prabhu@gmail.com

T. Murugan

Department of Information Systems and Security, College of Information Technology, United Arab Emirates University, Al Ain, UAE

1 Introduction

Machine learning refers to the machines that have been genetically configured to behave and process information like humans to a model for human intelligence. Because intelligent technology is based on the same principles as human intelligence, machines can easily mimic human intelligence and carry out tasks of all complexity levels. Machine learning has three main objectives: instruction, deductive thinking, and perception. The main objective of this paper is to early predict the chronic heart disease. The heart disease also known is cardiovascular disease (CVD) which causes artery damage to organs like the renal system, heart, eyes, and brain. Thus, it puts people's health in danger [1]. One of the main causes of young people's mortality is CVD in the majority of developed and developing countries worldwide. Given the complexity of CVD, handling it requires caution. Without it, the heart's condition could deteriorate and sudden death could happen [2]. One of the remedy would be applying the variety of machine-learning techniques to early predict the diseases. The severity of heart disease in patients has been evaluated using a variety of data analysis and machine-learning techniques [3]. The performance of the machine-learning algorithms always depends on feature selection strategies and optimization mechanism. The dataset required to conduct the experiment was gathered from the various hospitals of Karnataka, India. In the proposed paper the informative features are extracted from the heart disease dataset and machine-learning models are further optimized using Hybrid Dragonfly-Genetic Algorithm. The proposed Hybrid Dragonfly-Genetic Algorithm is applied in combination with various machine-learning algorithms such as Random Forest, Decision Tree, SVM, Naive Bayes classifier and stacked classifier on the considered heart disease dataset. The experimental results demonstrates that the Hybrid Dragonfly-Genetic Algorithm in combination with stacked classifier obtained higher accuracy compared to existing approaches.

2 Literature Survey

An expanding field related to medical computer vision is the usage of artificial intelligence or ML algorithms for the diagnosis of heart disease. The increased use of these procedures is due to how accurately they can diagnose the illnesses. Several methods for diagnosing heart disease have been developed by researchers utilizing algorithms such as Supported Vector Machine (SVM), Decision Trees(DT), Random Forest(RF) and Naive Bayes Classifier (NB). In this paper, a method utilizing KNN, NB, and Decision Tree classifiers was proposed. In [4] Decision Tree algorithm was applied to predict the heart disease. The algorithm achieved an accuracy of 85%. Using the R attribute selector and the Naive Bayes machine learning algorithm, author [5] proposed a method for heart disease diagnosis. The approach obtained an accuracy of 86% while predicting the CVD. In [6] Neural Network based algorithm is applied on CVD dataset for the early prediction of disease. The algorithm achieved

an accuracy of 78%. The author [7] attempted to forecast the heart disease using KNN algorithm. The model predicted the disease with an accuracy of 68%. In [8] Deep Neural Networks (DNN) based approach is utilized to predict the chronic heart disease. However the DNN algorithm predicted the disease with only 75% accuracy. In [9] SVM algorithm was used to predict the heart disease. The results showed that SVM algorithm predicted the disease with 82% accuracy. The development of ML and improvements in computing power have opened up a number of research possibilities in the field of healthcare [10]. To improve the precision of disease prediction, numerous researchers have suggested machine-learning (ML) algorithms [11, 12]. In [13] authors used Logistic Regression based machine learning model on Cleveland heart disease dataset to predict the disease. The missing values and irrelevant features in the Cleveland dataset were eliminated using correlating Pearson coefficients. A KNN imputation method was put forth in [14] for the forecasting the CVD disease where dataset had both continuous variables (average of the nearest neighbors) and categorical factors (most frequent). After cleaning the dataset and locating missing values during pre-processing, an ensemble based machine-learning classifier was imposed on the disease dataset. The approach achieved an accuracy of 87.1% [15].

3 Proposed Methodology

Implementation of an efficient machine-learning model for the early prediction of chronic disease is very much essential to provide appropriate treatment in right time which can cure diseases relatively faster hence it improves the quality of life.

Machine-learning algorithms is also capable of forecasting and tracking the growth of the diseases by looking at information collected. The gathering of data and selection of the most crucial attributes is the first step in the implementation of machine learning model. Different stages involved in implementing machine-learning algorithms for prediction of diseases are dataset gathering, selecting the data, data pre-processing, data balancing, and health forecasting. The dataset related to CVD disease was gathered from the various hospital of Karnataka, India over the course of almost two months, is used for implementing model. The feature selection process is considered one of the important stage of machine learning which includes choosing the traits or characteristics from the given dataset to make predictions about presence or absence of disease. The feature selection process is done to improve the efficiency of the implemented machine learning model. In the proposed paper Hybrid Dragonfly-Genetic Algorithm is used to select optimal features from the disease dataset. The flowchart plotted in Fig. 1 illustrates the various stages applied in Hybrid Dragonfly-Genetic Algorithm. As first step the dragonfly position parameters or population parameters are initialized. The pseudocode of the proposed Hybrid Dragonfly-Genetic Algorithm is shown in Fig. 2. The Eq. (1) shown indicates the computation of population parameters. Once the population parameters are determined step vectors are calculated for all the populations of Z_a . As next step, the objective functions of all the populations are derived through the computed step

vectors. The next step is to determine the near hood radius of dragonflies T_p , B_a , D_1 , B_r and E_r respectively, and then modify the near hood radius based on the positions of the dragonfly.

The positions of dragonfly is analyzed closely, in case present dragonfly has at least one near dragonfly, then crossness is chosen and mutation process is carried

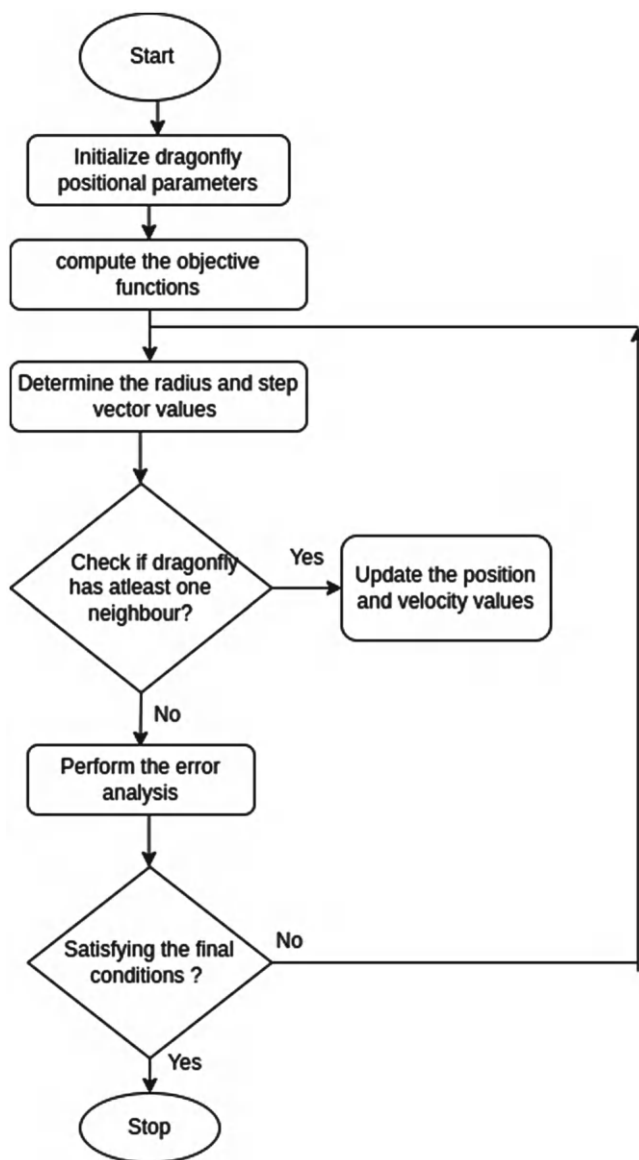


Fig. 1 Work flow of proposed methodology

```

Start: Computing population of dragonflies as  $Z_a = (e_a^1, e_a^d, \dots, e_a^N)$  (1)
Computing the step vectors
While the ending step is not met
    Determine the Objective function for all  $Z_a$ 
    Determine  $T_p, B_a, D_l, B_r$  and  $E_r$ 
    Modify the near hood radius
    If the present dragonfly has at least one near dragonfly
        Choose two crossness
        Carryout mutation
    Else
        Update the step of the vector using equation
         $\Delta M_{t+1} = (sT_p + qB_l + cD_h + fB_r + eE_r) + W\Delta Z_t$  (2)
        Modify the state of the vector using
         $Y_{t+1} = Y_t + \Delta Y_{t+1}$  (3)
    End if
    Evaluate and perform error the new position on new boundaries of
    variables
End While
End

```

Fig. 2 Pseudocode of hybrid dragonfly-genetic algorithm

out. If the dragonfly has does not have another dragonfly nearby then step vector is updated using Eq. (2) and modify the state of the vector using Eq. (3). The process is continued for all the new position occupied by dragonfly on new boundaries. The most optimal features are extracted from the dataset using the proposed Hybrid Dragonfly-Genetic Algorithm. The optimal features of the disease dataset are used to build machine learning model such as SVM, KNN, Naïve Bayes, Decision Tree and Random forest. The performance of the machine learning model is compared using the various parameter such Accuracy, ROC curve, F Score and confusion Matrix. The section below illustrates the result obtained through the proposed approach.

4 Results and Discussion

The performance of the proposed algorithms are analyzed using the various datasets pertaining to chronic diseases. Accuracy, loss, precision, recall, and F-Score were used to gauge the performance of the machine learning model. Testing and Training data were both used with the models and the dataset were split in ratio of 70:30 where 70% of data belongs to training the model and 30% of data utilized the examine the accuracy of the implemented machine-learning model.

Exploratory data analysis was carried out on the considered disease dataset to analyze how attributes affect a particular disease. Figure 3 represents the results of exploratory data analysis carried out on chronic kidney disease which illustrates

patient with Pheochromocytoma (PCC) and its impact on CKD disease. Figure 4 indicates the analysis on patient with Coronary Artery Disease (CAD) having CKD.

Figure 5 represents the exploratory data analysis result on frequency of heart disease on sex attributes. The result demonstrates that on considered dataset, the disease is more prevalent among female than male.

Figure 6 represents the exploratory data analysis result on heart disease frequency with respect to different chest pain. The result demonstrates that the chest pain type

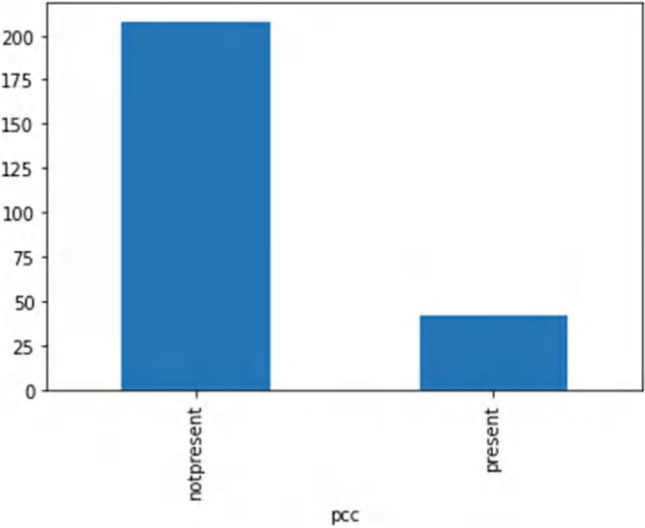


Fig. 3 Analysis of people with pheochromocytoma (PCC) having CKD

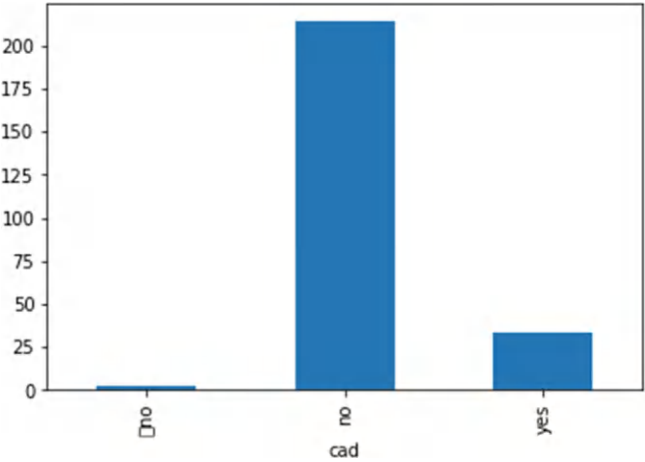


Fig. 4 Analysis of people with coronary artery disease (CAD) having CKD

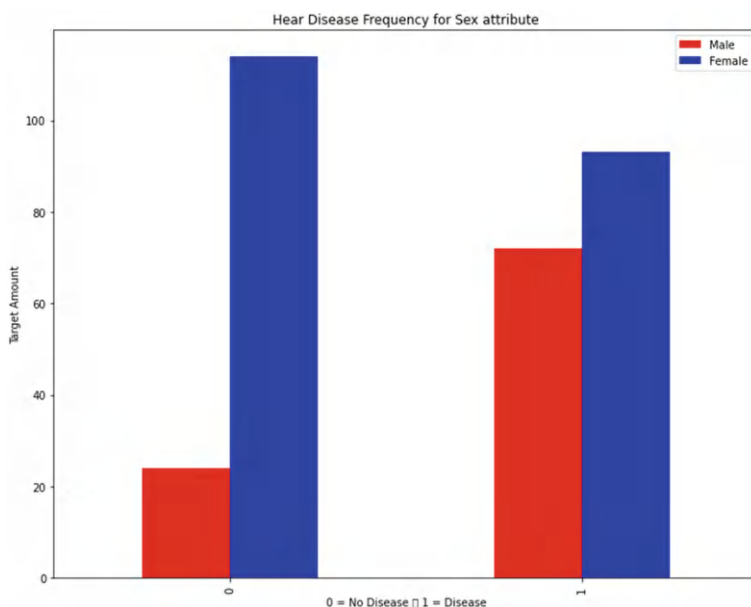


Fig. 5 Analysis on heart disease frequency for sex attribute

0 will not contribute much toward heart disease whereas type 1, 2 and 3 are the chest pain related to heart disease itself which cannot be neglected. As part of prediction of disease KNN, SVM, Naive Bayes, Random Forest and Decision Tree and stacked classifier which is combination these four classifier are applied in combination with the proposed Hybrid Dragonfly-Genetic Algorithm algorithms. The performance of the algorithms are analyzed using various parameters such as precision, F-Score, ROC, Confusion Matrix and accuracy.

Figure 7 illustrates the result obtained using KNN algorithms on heart disease dataset. It can be analyzed from the figure that the algorithm achieved accuracy of 78% on test data when the number of neighbors are 11.

The experimental results also revealed that Naive Bayes model achieves 82.56% accuracy and Support Vector Machine, Decision Tree and random forest algorithms achieved 84.72%, 85.23%, and 88.71% respectively. The stacking model which is combination of these four model achieved an higher accuracy of 97.35%. Figures 8 and 9 shows the Confusion Matrix generated through Naïve Bayes Model and SVM model on heart disease dataset.

In order to increase the accuracy of various machine learning model on Test Dataset, the performance of the model is evaluated on the unseen data and efficiency of various model is validated by applying K fold cross validation technique, where value of K was chosen as 10.

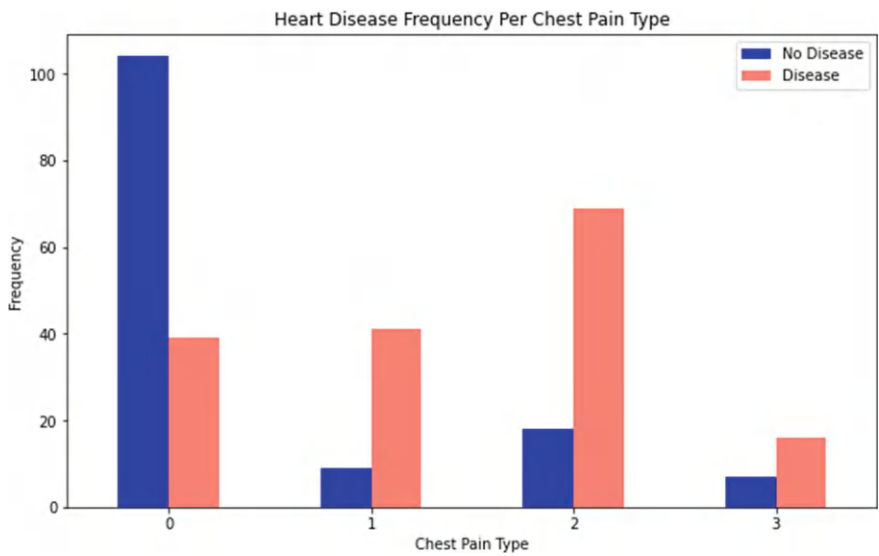


Fig. 6 Analysis on heart disease frequency based on chest pain

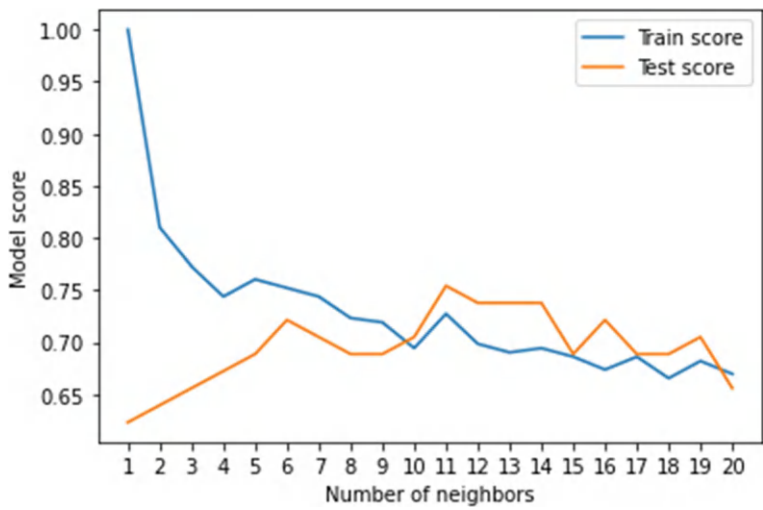


Fig. 7 Result obtained using KNN algorithm

By applying the cross-validation technique, the generalizability of the model can be enhanced. Figure 10 represents the cross-validation accuracy results of several classifiers.

Figure 11 illustrates the result obtained through stacked classifier which are implemented by using combination of algorithms such as SVM,KNN, Naïve Bayes,

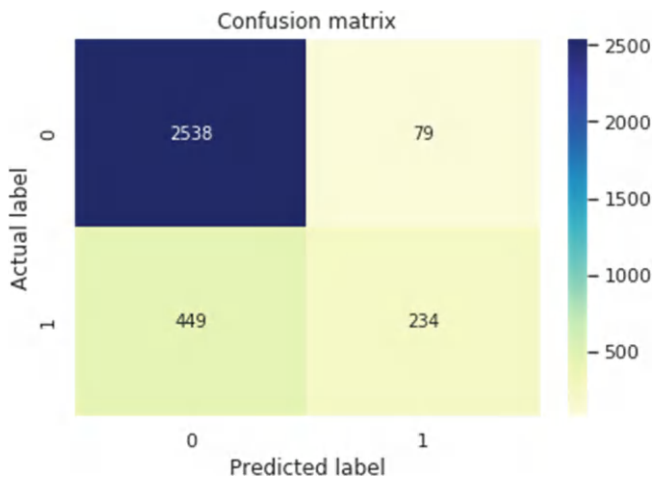


Fig. 8 Confusion matrix of Naive Bayes model

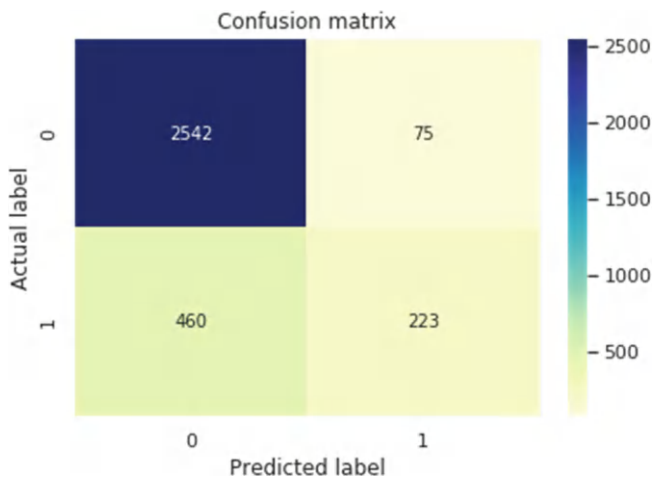


Fig. 9 Confusion matrix of SVM model

Random Forest and Decision Tree. The stacked algorithm achieved AUC rate of 0.97. Figure 12 illustrates the result obtained after applying various machine learning model on heart disease dataset. Table 1 demonstrates the Accuracy and F Score results achieved through various machine learning models. Experimental results illustrates that the proposed stacked classifier in combination with Hybrid Dragonfly-Genetic Algorithm attained highest accuracy of 97.35% compared to the existing models.

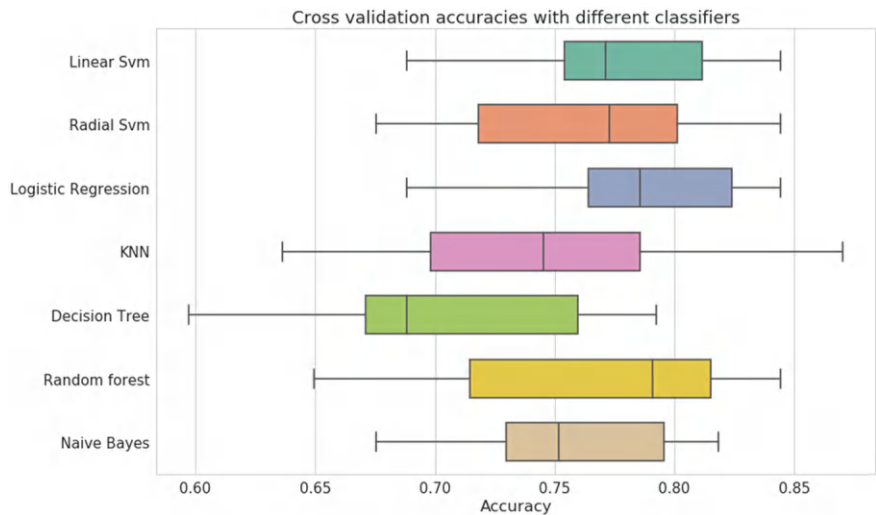


Fig. 10 Cross validation accuracy results of different classifiers

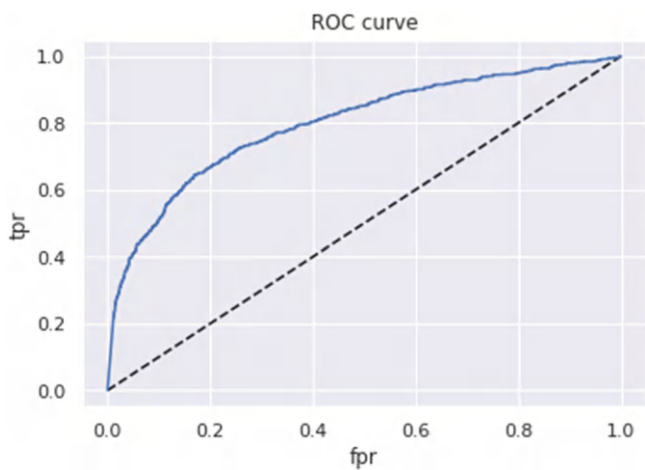


Fig. 11 ROC curve generated using stacked classifier on heart disease data

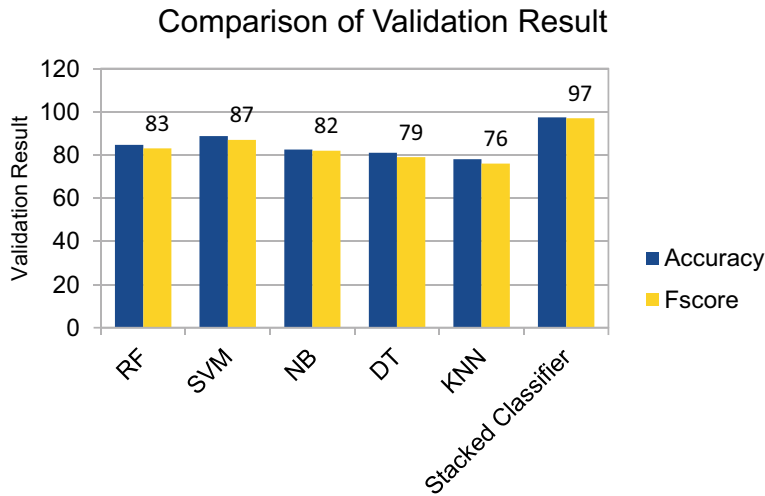


Fig. 12 Comparison of validation results

Table 1 Validation results of various ML model

ML model	Accuracy	F score
RF	84.72	0.83
SVM	88.71	0.87
NB	82.56	0.82
DT	81	0.79
KNN	78	0.76
Stacked classifier	97.35	0.97

5 Conclusion

Heart diseases are among the most common causes of death in India and throughout the world. Early identification of heart disease can aid high-risk patients in making decisions about modifying their lifestyles to decrease their obstacles which can be an important advancement in medical care. More people are given cardiac disease diagnoses each year. This necessitates a quick diagnosis and plan of action. In this paper a Hybrid Dragonfly-Genetic Algorithm is proposed in combination with various machine learning algorithms namely SVM, Naïve Bayes, Decision Tree, KNN, Random Forest and stacked classifier to early predict the chronic heart disease. The proposed Hybrid Dragonfly-Genetic Algorithm increased efficiency of machine learning algorithm by optimizing the attributes from the heart disease dataset. In order to accurately predict the disease, it is intended to use a variety of evaluation metrics, including the Confusion Matrix, ROC, Precision, Recall and F Score. From

the experimental results it can be concluded that the Hybrid Dragonfly-Genetic Algorithm in combination with stacked classifier achieved an highest accuracy of 97.35% compared to other traditional techniques.

References

1. Cao Q et al (2023) Phenome-wide association study and precision medicine of cardiovascular diseases in the post-COVID-19 era. *Acta Pharmacol Sin* 44(12):2347–57
2. Aboud MA et al (2022) Anesthetic considerations for adult down's syndrome patients with congenital heart disease for noncardiac surgery: a review article. *J Cardiothorac Vasc Anesth* 37(4):613–626
3. Ali MM et al (2021) Heart disease prediction using supervised machine learning algorithms: performance analysis and comparison. *Comput Biol Med* 136:104672
4. Ozcan M, Peker S (2023) A classification and regression tree algorithm for heart disease modeling and prediction. *Healthc Anal* 3:100130
5. Rustamov Z et al (2023) Enhancing cardiovascular disease prediction: a domain knowledge-based feature selection and stacked ensemble machine learning approach
6. Wankhede J, Sambandam P, Kumar M (2022) Effective prediction of heart disease using hybrid ensemble deep learning and tunicate swarm algorithm. *J Biomol Struct Dyn* 40(23):13334–13345
7. Absar N et al (2022) The efficacy of machine-learning-supported smart system for heart disease prediction. In: *Healthcare*, vol 10(6). MDPI
8. Hassan D, Hussein HI, Hassan MM (2023) Heart disease prediction based on pre-trained deep neural networks combined with principal component analysis. *Biomed Signal Process Control* 79:104019
9. Saputra D, Dharmawan WS, Irmayani W (2022) Performance comparison of the SVM and SVM-PSO algorithms for heart disease prediction. *Int J Adv Data Inf Syst* 3(2):74–86
10. Alanazi A (2022) Using machine learning for healthcare challenges and opportunities. *Inform Med Unlocked* 30:100924
11. Ahsan MM, Luna SA, Siddique Z (2022) Machine-learning-based disease diagnosis: a comprehensive review. In: *Healthcare*, vol 10(3). MDPI
12. Krishnamoorthi R et al (2022) A novel diabetes healthcare disease prediction framework using machine learning techniques. *J Healthc Eng* 2022
13. Ayon SI, Islam MM, Hossain MR (2022) Coronary artery heart disease prediction: a comparative study of computational intelligence techniques. *IETE J Res* 68(4):2488–2507
14. Saranya G, Pravin A (2023) A novel feature selection approach with integrated feature sensitivity and feature correlation for improved prediction of heart disease. *J Ambient Intell Humaniz Comput* 14(9):12005–12019
15. Tiwari A, Chugh A, Sharma A (2022) Ensemble framework for cardiovascular disease prediction. *Comput Biol Med* 146:105624

Unsupervised Resume Evaluation and Information Extraction: Leveraging Data-Driven Strategies for Applicant Ranking and Data Retrieval



V. Jaivignesh  and V. UmaRani 

Abstract Due to the overwhelming volume of job applications they get, recruiters today confront a difficult task. The necessity for automatic resume evaluation has grown to address this problem. This study presents a thorough methodology that uses Latent Dirichlet Allocation (LDA) for resume assessment and SpaCy for object detection in answer to this need. Our goal with this analysis is to transform the hiring process by optimizing the resume evaluation process and making it more effective and efficient. The suggested methodology extracts important elements from resumes, such as education, experience, and skills, using Named Entity Recognition (NER) from SpaCy. The LDA model then uses these discovered entities to give each a topic likelihood score, producing an overall assessment of the resume. Our study not only presents a full analysis of entity detection along with an extensive set of evaluation criteria, in addition, concentrates on the creation of this novel method. The methodology's results show that it can greatly improve the hiring process by assuring a more accurate assessment of candidates' qualifications while saving time and resources. To automate resume evaluation, our research presents a novel approach that blends entity detection with LDA. This strategy can change the hiring procedure. Our research indicates that this strategy is quite promising and could help recruiters overcome the increasing difficulties they experience in the contemporary employment market.

Keywords Resume evaluation · Entity detection · SpaCy · Latent Dirichlet allocation · NER · Recruitment · Automated assessment

V. Jaivignesh (✉)

Saveetha Engineering College, Thandalam, TamilNadu, India

e-mail: jaivignesh_jv@outlook.com

V. UmaRani

Department of CSE, Saveetha Engineering College, Thandalam, Chennai, India

1 Introduction

In the realm of modern recruitment and employment, the importance of effectively evaluating job applicants' qualifications cannot be overstated. The efficiency and accuracy with which this task is performed have a profound impact on an organization's ability to select the right talent and optimize its workforce. This paper embarks on a journey to present an innovative solution: a Resume Rater system that serves as a robust tool for assessing applicants based on the content of their resumes. This approach delves into both supervised and unsupervised methodologies, offering a comprehensive perspective on resume evaluation. The choice to explore a dual approach is rooted in the inherent complexities of determining how best to rank resumes. In this pursuit, we aim to bridge the gap between the dynamics of a job applicant's interaction with potential employers, mirroring the intricate dance that unfolds during the recruitment process. The two primary approaches adopted in this endeavor are designed to decipher the intricacies of the resume evaluation process. The first methodology involves the use of predefined keywords, offering a supervised path to resume analysis. The second approach, conversely, harnesses unsupervised techniques, particularly Latent Dirichlet Allocation (LDA), to unveil the hidden keywords embedded within the resumes themselves. This reverse-engineering strategy is driven by the assumption that resumes are crafted based on specific keywords, a hypothesis we explore in the context of a probabilistic model. In the resume evaluation process, it is imperative to assess the relevance of keywords within the content. For this purpose, two unique matching techniques are included in our Resume Rater system: Keyword Match and Within Match. These techniques assess the importance of the keywords and the degree of coherence in each resume's content. Following the completion of these evaluations, a rating reflecting the applicant's suitability for the role is assigned. To enhance the reliability of the unsupervised approach, a proposed ensemble model is introduced to mitigate variance, ensuring consistency across ratings. Categorizing and extracting relevant information from resumes are fundamental to the hiring process. While the paper highlights the practicality of a rule-based approach, it also recognizes the potential for further advancements through Named Entity Recognition (NER) models. However, the challenges associated with labeling, diverse document formats, multilingual entities, and various other complexities underscore the continued relevance of the rule-based approach. This paper aims to unravel the complexities of resume evaluation and information extraction in the context of modern recruitment practices. It underscores the strengths of a multifaceted approach that can serve as an efficient first-pass screening tool. It also draws attention to the prospects of a more supervised model for comprehensive applicant ranking, and the importance of leveraging NER models for information extraction, particularly in the context of larger organizations with extensive applicant databases. For smaller businesses, the rule-based approach remains valuable, offering high recall and ensuring that no pertinent details are overlooked during the screening process. The journey through this paper explores the intricate intricacies of resume evaluation and information extraction, shedding light on their significance

and the potential for innovation and enhancement in these critical aspects of the employment ecosystem.

2 Literature Survey

The evaluation of resumes has evolved significantly through various methodologies and technologies introduced in academic research. These studies have contributed immensely to the enhancement of the hiring process by leveraging techniques from NLP, machine learning, keyword extraction, and context-aware analysis. Bae and Kim's study [1] in 2018 was pivotal in advocating for text mining's utilization to classify job resumes. Their emphasis on leveraging NLP methods underscored the importance of efficiently categorizing resumes for employment by scrutinizing their content. This approach not only streamlined the process but also demonstrated how text mining could substantially improve the effectiveness of initial resume screenings. Automatic categorization based on content emerged as a viable strategy, showcasing the potential for significant time savings in the recruitment process.

Building upon this, Smith and Brown [2] 2019 delved into machine learning-based resume screening. Their exploration illuminated the capabilities of machine learning algorithms in analyzing resumes to identify pertinent skills and qualifications. Algorithms such as support vector machines and decision trees were discussed as effective tools for automating the initial screening process, enabling HR departments to save valuable time and resources. The integration of machine learning brought forth a data-driven approach that revolutionized resume evaluations. Wang and Zhao's review [3] in 2020 expanded the discussion by examining various keyword extraction methods. This comprehensive review delineated the significance of keyword extraction techniques in the resume review process. Their analysis encompassed diverse approaches, including machine learning, linguistic, and statistical methods, emphasizing their relevance in extracting crucial information from resumes. This emphasized the role of structured data extraction in enhancing the efficiency of resume assessments.

In the realm of specialized techniques, Blei et al.'s groundbreaking work on Latent Dirichlet Allocation (LDA) [4] in 2003 introduced a topic modeling technique capable of identifying thematic content within documents, including resumes. LDA's ability to unearth hidden topics in a collection of documents made it a valuable tool for identifying thematic content within resumes, enhancing the understanding of contextual relevance in job applications. Furthermore, advancements in Named Entity Recognition (NER) were highlighted by Lample et al. [5] in 2016. Their exploration of neural approaches for NER was pivotal in extracting specific data like names, addresses, and qualifications from resumes. Techniques such as conditional random fields (CRFs) and bidirectional LSTMs demonstrated promising avenues for accurately extracting structured information from unstructured resume data. Singh et al.'s extensive survey [6] in 2021 provided crucial insights into job matching and resume parsing techniques. Their emphasis on structured data extraction from resumes underscored its pivotal role in expediting the matching process. Covering

a wide array of techniques, from machine learning-driven approaches to rule-based parsing, their survey highlighted the opportunities and challenges within this domain.

In tandem, Rokach's study [7] in 2010 delved into ensemble-based classifiers and their benefits in resume rating systems. This examination of combining several models to augment prediction accuracy showcased the potential of ensemble techniques like boosting and bagging in improving the consistency and dependability of resume evaluations, especially when multiple evaluation criteria were at play. However, challenges persisted in resume assessment, as highlighted by Zhang et al. [7] in 2021. Despite their primary focus on public health literature, their discussion on text data mining challenges resonated with resume assessments. Named entity recognition, handling varied document formats, and managing unstructured data were identified as critical roadblocks, emphasizing the necessity of overcoming these challenges for efficient resume assessments.

Si and Kim [8] in 2018 offered a comprehensive review of machine learning's application in human resource management, particularly in talent acquisition and recruitment. Their insights underscored how machine learning methods, including NLP and predictive modeling, could significantly enhance the efficiency and data-driven nature of the hiring process. Moreover, recent research in 2022 introduced a context-aware approach to resume analysis. This innovative approach considered specific job requirements or company needs, aiming to enhance the matching process by accounting for contextual nuances. By aligning resume analyses with the particularities of job postings or organizational demands, this methodology aimed to optimize the hiring process by ensuring a more precise fit between candidates and positions.

In conclusion, the collective insights from these diverse studies have immensely enriched the landscape of resume evaluation methodologies. Leveraging NLP, machine learning, keyword extraction, and context-driven analyses has redefined the paradigms of efficient and effective resume assessments, paving the way for more streamlined and insightful hiring processes.

3 Methodology

3.1 Resume Evaluation

For the task of resume evaluation, we consider both supervised and unsupervised approaches, guided by the recognition of the unique challenges within this domain. The choice between these methods depends on the predefined knowledge of keywords, which is typical in supervised approaches, or the exploration of latent topics and keywords within the documents, as employed in unsupervised approaches.

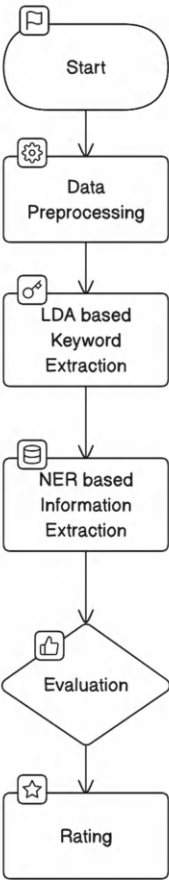
In the supervised approach, we rely on predefined sets of keywords, obtained from domain experts and existing job descriptions. These keywords act as initial parameters for the evaluation. Measuring how closely the resume content and these

keywords match forms the basis of the evaluation. The goal is to identify how well an applicant’s qualifications match the expected keywords, providing a familiarity rating.

In the unsupervised approach, we embrace the challenge of reverse engineering the latent keywords that candidates expect organizations to see. This is accomplished using a generative model called LDA. That uncovers hidden topics within the documents. The LDA model provides probabilities for keywords based on topics. The keywords are then ranked by their relevance to each document, allowing us to dynamically determine the expected keywords (Fig. 1).

Fig. 1 Methodology diagram

Unsupervised Resume Evaluation Process



3.2 Information Extraction

Rule-based approaches have a relatively high recall, they are used for information extraction, ensuring that relevant information is extracted from resumes—even those with complex designs and formatting. To identify important entities like names, email addresses, phone numbers, and job experiences, the algorithm adheres to a set of rules. Furthermore, by identifying category boxes, the system can classify and identify data into segments like education, skills, and work experience. To address challenges such as diverse document formatting, multilingual entities, and idiosyncratic content presentation, we have incorporated specific rule-based logic. This logic accounts for potential variations in text placement and formatting discrepancies, allowing for improved extraction accuracy. Resumé rating is approached more adaptively and data-drivenly in the unsupervised method. Rather than depending solely on pre-selected keywords, our goal is to ascertain which keywords a candidate anticipates a company reviewing their resume for. To do this, we use a generative model called Latent Dirichlet Allocation (LDA), which assumes that each document is a combination of a few different topics, with each word being assigned to one of these topics. This enables us to deduce the candidate's expected keywords in reverse engineering. As if we are attempting to ascertain the keywords that the candidate anticipates the employer seeing (for a specific global space of keywords K and documents D , the candidate is attempting to do.

$$\text{Document } (d) = \text{argmax } ((d), \text{ keyword1} \dots \text{keywords}) * P(D|K) \quad (1)$$

Next, we apply a generative model called Latent Dirichlet Allocation (LDA), which assumes that every document d is a combination of a few topics z and that every word in the document is associated with one of the documents. To put it another way, it presumes that d is produced by keywords drawn from a variety of topics. It calculates:

1. $P(Z|D = d)$, the likelihood that d is produced for each z in Z , Random CV, for instance, is [R1: 0.1, R2: 0.3, R3: 0.2, R4: 0.4]
2. $P(K|Z)$, the Likelihood that a Word from K is Selected from Z 's Topic Z . For Example,
 z_1 [software: 0.15, engineering: 0.5, leadership: 0.3, computer: 0.1,...] == 1
 z_2 : [asset: 0.08, leadership: 0.1, quantitative: 0.21, finance: 0.33,...] == 1
 z_3 : [lead: 0.05, report: 0.07, general: 0.1, management: 0.12,...] == 1
 z_4 : [award: 0.08, recruitment: 0.1, resource: 0.21, human: 0.22,...] == 1.

The keywords are then found using the Bayesian network.

$$\begin{aligned} k'_{1\dots n} &= \text{nargmax}_{k \in K} P(K|D = d) \\ &= \text{nargmax}_{k \in K} \sum_{i \in Z} \frac{P(K \cap Z \cap D)}{P(D)} \end{aligned}$$

$$\begin{aligned}
&= \operatorname{argmax}_{k \in K} \sum_{i \in Z} \frac{P(K|Z \cap D)P(Z|D)P(D)}{P(D)} \\
&= \operatorname{nargmax}_{k \in K} \sum_{i \in Z} \frac{P(K|D|Z)P(Z|D)P(D)}{P(D)} \\
&= \operatorname{nargmax}_{k \in K} \sum_{i \in Z} P(K|DZ)P(Z|D = d)
\end{aligned}$$

where $\operatorname{nargmax}$ is the argmax for the top n keywords and $P(K|D = d)$ is the probability that for each k in K , d is generated. Since the probability of K is only directly conditioned on Z , we factorize $P(K|ZD) \setminus = P(K|Z)$. We can easily compute $P(K|D = d)$ for all k since LDA provides $P(Z|D = d)$ and $P(K = k|Z = z)$.

4 Results and Discussion

Evaluation and information extraction utilizing LDA and SpaCy's NER techniques will be covered in this section. To build a strong resume analysis and rating system, the project makes use of several tools and libraries. To set up a web server that can receive resume files, use Flask. Upon file upload, the application employs Apache Tika for parsing PDF documents, extracting text content, and organizing it for further processing. SpaCy, a natural language processing library, is used to recognize entities within the resume, such as names, emails, and contact details. Regular expressions are then leveraged to identify critical sections in the resume, like work experience and education, which are subsequently analyzed to compute the total experience in months or years. Skills listed in the resume are also extracted and presented to the user. A predefined dictionary of technical roles and their corresponding skills is used to match the applicant's skills and suggest potential job roles. Furthermore, a linear regression model rates the resume based on various factors, including experience, skills, and education. Finally, the system returns the resume rating, extracted information, and suggested job roles, facilitating streamlined resume analysis and efficient job role matching for candidates and employers.

4.1 Information Extraction Results

Rule-Based Method

The information extraction method based on rules has shown to be dependable in obtaining important entities like names, phone numbers, email addresses, and employment histories. The method showed a high recall rate, guaranteeing that pertinent data was taken out of resumes with intricate formatting. Particular rule-based


```
Top terms for topic 0:
['student', 'fluent', 'chinese', 'english', 'collect', 'administration', 'mandarin', 'business administration', 'native', 'china']
Top terms for topic 1:
['strategy', 'target', 'focus', 'asia', 'structure', 'market', 'proposal', 'negotiation', 'lead', 'term']
Top terms for topic 2:
['coordinate', 'involve', 'department', 'credit', 'set', 'request', 'review', 'control', 'bank', 'report']
Top terms for topic 3:
['leader', 'value', 'plan', 'development', 'team', 'communication', 'include', 'best', 'management', 'service']
Top terms for topic 4:
['risk', 'relationship', 'house', 'credit', 'proficient', 'lead', 'client', 'management', 'develop', 'team']
Top terms for topic 5:
['engagement', 'audit', 'audits', 'control', 'finding', 'assurance', 'procedure', 'accountants', 'entity', 'lead']
Top terms for topic 6:
['expect', 'notice', 'computation', 'forecast', 'salary', 'tax', 'job', 'consolidation', 'taxation', 'expect salary']
Top terms for topic 7:
['meeting', 'law', 'matter', 'board', 'regulation', 'reference', 'advisory', 'draft', 'advise', 'personal']
Top terms for topic 8:
['budget', 'honours', 'science', 'software', 'bachelor', 'accounting', 'function', 'list', 'block', 'auditor']
Top terms for topic 9:
['banking', 'base', 'basis', 'position', 'senior', 'transaction', 'monitor', 'maintain', 'datum', 'industry']
Top terms for topic 10:
['international', 'master', 'china', 'equity', 'security', 'capital', 'corporation', 'fund', 'investment', 'scholarship']
Top terms for topic 11:
['event', 'vice', 'organize', 'day', 'call', 'execution', 'kyc', 'open', 'markets', 'aim']
Top terms for topic 12:
...
Top terms for topic 48:
['liaise', 'exchange', 'administration', 'requirement', 'update', 'enquiry', 'book', 'provide', 'system', 'assist']
Top terms for topic 49:
['trade', 'calculation', 'broker', 'fee', 'reconciliation', 'pricing', 'hedge', 'equity', 'action', 'query']
```

Fig. 3 Top terms in each topic

4.2 Resume Evaluation Results

In the supervised approach, we observed that the use of predefined keywords led to a structured and relatively straightforward evaluation process. The Keyword Match (KM) strategy provided a familiarity rating based on the degree of keyword matches. This approach offered consistent results and was particularly effective for resumes that closely matched the predefined keywords. However, it needed to be more adaptable to variations in language and the dynamic nature of resume content. The unsupervised approach, using LDA to dynamically determine keywords, revealed promising results. LDA allowed us to identify latent topics within resumes and extract keywords based on these topics. The Within Match (WM) strategy, which evaluated the relevance of words matched by keywords, further improved the evaluation process. This approach adapted well to varying resume content and provided more flexibility in handling diverse language usage. Both the supervised and unsupervised approaches resulted in final ratings that could be normalized to a common scale (0–10). Normalization helped ensure that the rating system was consistent and could be applied uniformly to all resumes. It prevented extreme outliers and allowed for fair comparison across the dataset. example of how our resume evaluation model ranks a specific resume, along with the rating received, and displays the content of the ranked resume as an image (Figs. 4 and 5).

Sample Resume Ranking

Resume Title: Network Engineer Resume

Rating Received: 8.8/10

Resume Title: Full Stack Developer Resume

Rating Received: 8.9/10

The ranked resume content will be displayed as an image, highlighting the key strengths and qualifications of the applicant. This example showcases how our resume

```

1 {
2   "rating": 8.8,
3   "info": {
4     "Education": [
5       "Saveetha Engineering College, Chennai, India, 2020 - Present",
6       "Vani Vidyalaya Senior Secondary School, West KK Nagar, Chennai, India, 2020"
7     ],
8     "Education Experience": "3 years 10 months",
9     "Work Experience": "0 months",
10    "Works": "Technical Volunteer\nCollaborated with a team to design a visually captivating magazine layout\nInterests",
11    "city": "India",
12    "email": "jyoutlook.com",
13    "name": "Jaivignesh",
14    "number": "+91-9499924412",
15    "skills": [
16      "Routing", "System", "Linux", "Data analysis", "Interactive", "Python", "C", "Visual", "Database", "UI",
17      "Windows", "Github", "Email", "Assembly", "Tv", "R", "Analytical", "Technical", "Pandas", "Seaborn", "Content",
18      "Security", "Spreadsheets", "Analytics", "Api", "Computer science", "English", "Tableau", "Mobile", "Training",
19      "Engineering", "Design", "Pillow", "Programming", "Matplotlib", "Os", "Networking", "Troubleshooting", "Testing",
20      "C++", "Excel", "Hardware", "Ux", "Java", "Analysis", "Sql", "Kpis"
21    ]
22  },
23   "curated": "Our analysis says you are looking for Network Engineer Jobs. (4 keywords found) Suggested roles:- Software Developer -"
24 }
25

```

Fig. 4 Results for network engineer resume

```

1 {
2   "rating": 8.9,
3   "info": {
4     "Education": [
5       "Saveetha Engineering College, Chennai, India, 2020 - Present",
6       "Vani Vidyalaya Senior Secondary School, West KK Nagar, Chennai, India, 2020"
7     ],
8     "Education Experience": "3 years 10 months",
9     "Work Experience": "0 months",
10    "Works": "Technical Volunteer\nCollaborated with a team to design a visually captivating magazine layout\nInterests",
11    "city": "India",
12    "email": "jyoutlook.com",
13    "name": "Jaivignesh",
14    "number": "+91-9499924412",
15    "skills": [
16      "Routing", "System", "Linux", "Interactive", "Python", "C", "Visual", "Database", "Css", "Ui",
17      "Windows", "Architecture", "Github", "Email", "Assembly", "Tv", "R", "Technical", "Content",
18      "Security", "Analytics", "Api", "Computer science", "English", "Html", "Mobile", "Javascript",
19      "Training", "Engineering", "Design", "Pillow", "Programming", "Matplotlib", "Os", "Networking",
20      "Troubleshooting", "Testing", "C++", "Excel", "Hardware", "Ux", "Analyze", "Analysis", "Coding"
21    ]
22  },
23   "curated": "Our analysis says you are looking for Full Stack Developer Jobs. (4 keywords found) Suggested roles:- Software Developer -"
24 }
25

```

Fig. 5 Results for full stack developer resume

evaluation model operates, providing a specific rating to an applicant's resume and offering insights into its content.

5 Conclusion

With the use of rule-based methods and Latent Dirichlet Allocation (LDA), we have provided an extensive unsupervised resume evaluation and information extraction approach in this study. By streamlining the resume assessment procedure and improving information extraction, our study aims to revolutionize the recruitment process and solve the difficulties recruiters encounter when managing a large number of applications. Key findings and takeaways from our research include the Dual Approach: We explored both supervised and unsupervised methods for resume evaluation, offering a dual approach to adapt to the varying nature of resume content. Supervised Approach: The supervised approach, using predefined keywords, provided a structured evaluation process but had limitations in adapting to dynamic language and content changes. Unsupervised Approach: Leveraging LDA for dynamic keyword extraction, the unsupervised approach demonstrated adaptability and promising

results in evaluating resumes with varying content. Information Extraction: The rule-based approach for information extraction proved reliable, with a high recall rate, while the prospects of Named Entity Recognition (NER) models were highlighted for larger organizations and diverse document formats. To sum up, our study presents novel approaches to the problems associated with contemporary resume assessment and data extraction. Combining the dual approach with machine learning and NER models has the potential to completely transform the hiring process by increasing its accuracy, efficiency, and flexibility in response to changing market demands. Recruiters can ensure a more accurate assessment of candidates' qualifications while saving time and money by utilizing data-driven strategies.

Future Directions

Future research may explore the integration of machine-learning algorithms for more accurate resume evaluation and information extraction. Further investigation into contextual analysis, taking into account job posting requirements, could enhance the matching process. Including the integration of machine-learning algorithms for enhanced evaluation and the contextual analysis of resumes to align with specific job posting requirements.

References

1. Bae H, Kim H (2018) Job resume classification system using text mining. In: Proceedings of the 13th international conference on machine learning and data mining
2. Smith J, Brown A (2019) Automated resume screening using machine learning. *J Artif Intell Hum Resour Manage* 3(2):61–74
3. Wang X, Zhao S (2020) A review on keyword extraction methods. *Int J Data Min Knowl Manage Process* 10(2):13–28
4. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. *J Mach Learn Res* 3:993–1022
5. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. arXiv preprint [arXiv:1603.01360](https://arxiv.org/abs/1603.01360)
6. Singh D, Kumar R, Singla R (2017) Resume parsing with named entity clustering and recognition. In: 2017 IEEE international conference on data mining and advanced computing (SAPIENCE)
7. Valdez-Almada R, Rodriguez-Elias OM, Rose-Gomez CE, Velazquez-Mendoza MDJ, Gonzalez-Lopez S (2017) Natural language processing and text mining to identify knowledge profiles for software engineering positions: generating knowledge profiles from resumes. In: 2017 5th international conference in software engineering research and innovation (CONISOFT), Merida, Mexico, pp 97–106
8. Zhang J et al (2021) Challenges and opportunities in text data mining of public health literature

Enhancing Hydroponic Efficiency Using IoT and Machine Learning Algorithms



R. Punith Raj, R. Shushank, S. Vaibhav, and M. Shahina Parveen

Abstract Agriculture plays a vital role in the economic development of nations. Hydroponics, a soil-less cultivation technique utilizing nutrient solutions, has gained popularity as an innovative agricultural strategy. This research focuses on enhancing hydroponic plant growth through the integration of machine learning algorithms, such as neural networks and Bayesian networks (Banerjee T, Roy S in Deep neural network fitting model for hydroponics system, IEEE, pp 1–6 (2019); Deepak B, Singh R in IoT-based smart agriculture: a review on smart hydroponic systems. IEEE, pp 1–6 (2021)) [1, 2]. Additionally, leveraging the Internet of Things (IoT), we aim to develop an intelligent system capable of autonomous control based on diverse input data. Hydroponics offers the potential for soil-less plant cultivation, provided specific requirements are met. By optimizing nutrient solutions and environmental variables, plants can thrive without traditional soil-based methods. Machine learning algorithms, such as neural networks and Bayesian networks, have been extensively investigated to improve hydroponic growth conditions. These algorithms analyze various input data, identify patterns, and predict optimal control actions for maintaining an ideal hydroponic environment. The IoT enables machine-to-machine communication, allowing for independent and intelligent control of hydroponic systems. By integrating IoT devices, sensors, and actuators into the hydroponic setup, real-time data collection and feedback can be achieved. This data is then processed by machine learning algorithms to make informed decisions regarding nutrient composition, pH levels, temperature, humidity, and lighting conditions.

Keywords Hydroponics · Soil-less cultivation · Nutrient solutions · Innovative agricultural strategy · Hydroponic plant growth · Machine learning algorithms · Internet of Things (IoT)

R. Punith Raj (✉) · R. Shushank · S. Vaibhav · M. Shahina Parveen
Department of Computer Science and Technology, Dayananda Sagar University, Bengaluru,
Karnataka, India
e-mail: rajpunith469@gmail.com

1 Introduction

An entire country's economy is impacted by agriculture. Because of modern agricultural practices that enable plants to be cultivated without soil by using supplements, hydroponics is on the rise. Currently, some investigation has been made into using machine learning methods like neural networks and Bayesian networks to manage the growth of hydroponic plants. In cases where only a small amount of research has been done, AI has assisted in automating plant development. The soil is not taken into consideration when growing plants using standard methods. However, as long as these conditions are met, plants can grow, which opened the door to hydroponics cultivation [3, 4].

In hydroponic cultivation, plants are raised without the use of soil. Homesteads are able to go with the cultivators to the urban area using this method. The hydroponics system can be independently controlled and intelligently monitored thanks to the Internet of Things [5, 6]. Our goal is to create an intelligent system that, given a variety of information inputs, determines the appropriate control strategy for a hydroponic environment.

1.1 Problem Statement

The addition of pesticides to the soil for farming and agricultural purposes makes soil erosion a significant difficulty in contemporary agriculture and one of the key requirements for agriculture to survive. Soil erosion also has an indirect impact on air and water pollution. Farming is hampered by soil-borne pests, and employing pesticides can have negative effects. Pesticides used in traditional soil-based farming techniques wash into rivers and streams, polluting the water and harming the living things that live there. For agriculture and farming to continue, a solution that strives to reduce this pollution and the problem of soil erosion is essential. Modern technology and the advent of artificial intelligence-based smart systems could prove to be a fantastic answer to such a problem.

1.2 Proposed Solution

The proposed methodology aims to achieve optimal plant development by combining IoT and machine learning in tandem. Both elements are essential and have separate functions.

Farmers may now completely automate the hydroponics process thanks to the Internet of Things. IoT allows for automated control of many duties requiring the monitoring of PH level, water level, temperature, and light intensity. An ARM 7 Microcontroller will receive the information gathered by the hydroponics tank's

sensors and undertake ongoing monitoring to ensure optimal plant growth. By remotely controlling this system via a connection to our phones, other features can be enabled. An Arduino can be interfaced with the hydroponics task's sensors. The interface between the cloud and microcontroller is this thing called a Pi3. In order to control the hydroponic environment, data from sensors is used.

In the topic of synthetic agriculture in general, ML algorithms have a variety of applications. To regulate plant growth and optimize EC values in the nutrient solution, it is employed in hydroponics. Popular models like the Bayesian network can be used for precise environmental event control and observation. These models may do predictive analysis on a dataset gathered over the course of a month. Important output decisions for the system's autonomous control are made using this analysis.

As a result, IoT focuses on managing the hydroponic environment using a microcontroller and sensor data gathered from hydroponic tanks. On the other side, machine learning focuses on algorithmic level output prediction using collected parameters. IoT and ML collaborate in real-time to forecast the result and manage the system as a result.

1.3 Advantages of Hydroponics

Modern agricultural demands are met by soil-less plant cultivation, particularly through hydroponics, which has several advantages over traditional soil-based methods. Because hydroponics uses significantly less water than conventional growing methods, it offers itself as a resource-efficient option [7–10]. Water resources are conserved because the precise distribution of nutrients to plant roots reduces evaporation and guarantees maximum nutrient uptake. Furthermore, vertical farming is made possible by its space-efficient design, which makes it feasible in urban areas and places with limited land. This hydroponics feature encourages crop production all year round, regardless of seasonal variations, which boosts agricultural resilience. By controlling variables like pH levels, nutrient concentrations, temperature, and humidity, the technology may create a controlled environment that lowers the risk of disease and pests, reducing the need for chemical pesticides and herbicides.

Hydroponics provides a platform for innovation in innovative agriculture practices by enabling experimentation with different crop varieties and growth circumstances. Because of its adaptability, it can be used for year-round crop production, sustainable farming in places with poor soil quality, and agricultural exploration in urban areas. Furthermore, the precision and productivity of hydroponics are further improved by the integration of technologies like machine learning and the Internet of Things, which opens up new possibilities for agricultural innovation [11–15].

2 Methodology

Three main sections make up the complete system:

The Arduino is the first component. It transmits data to the microcontroller from sensors that measure pH, humidity, light intensity, temperature, and water level in hydroponic tanks. The Raspberry Pi, water tank, hydroponics tank, fertilizer tank, and every other component are all connected to the Arduino, which serves as the system's brain.

The Raspberry Pi3's Deep Neural Network Fitting Model, which was trained in the cloud using the gathered data set, is the second component. An intelligent choice would be made by the Pi3's fitting model to determine the output, which would then be communicated to Arduino, who would turn on the proper control mechanism. A power source will be used to power the Arduino and Raspberry Pi simultaneously.

The hydroponics tank would be the third component. A sustainable environment for plant growth would be maintained by this part. It will have a number of sensors, and a directed link will be used to send data from those sensors to Arduino [16–18].

Neural networks and Bayesian networks, in particular, are essential machine learning methods for optimizing hydroponic plant development. Neural networks are very good at pattern recognition and predictive modeling. They can determine the best course of action for control by using large datasets that include environmental variables and plant growth indicators. These algorithms improve decision-making processes to maintain optimal hydroponic conditions by continuously adapting and learning from incoming data. On the other hand, Bayesian networks are useful for evaluating the probabilistic correlations between different environmental conditions and the results of plant growth. By using historical data analysis to anticipate the ideal nutritional composition and growth parameters, these models help with accurate control and observation of environmental occurrences [19, 20].

These machine learning algorithms play a major role in improving connections between environmental parameters and plant growth in hydroponic environments by helping to comprehend complex data patterns. This enhances the effectiveness of hydroponic systems by maximizing resources and enabling real-time decision-making and predictive modeling (Fig. 1).

2.1 Use Case Diagram

See Fig. 2.

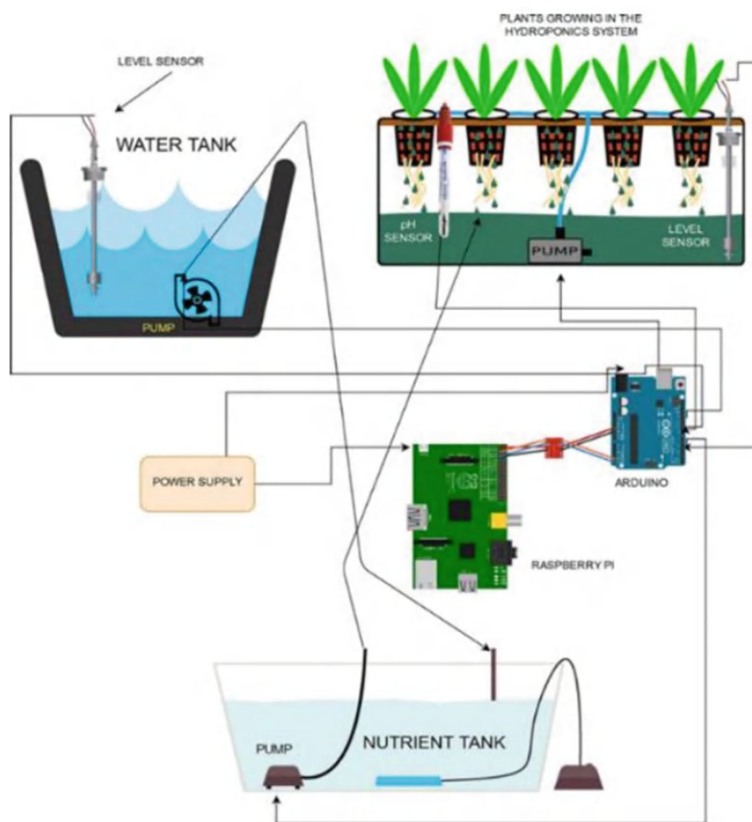


Fig. 1 System architecture of IoT based hydroponics

2.2 Data Flow Diagram

In Figure 3, the Raspberry Pi, the external water tank, the nutrient tank, and the hydroponics tank are the four primary components of the data flow diagram. Together, these elements control how this system is organized as a whole. The task of filling the tank with water will fall to the external water tank, which will then flush water into the hydroponics tank. The same hydroponics tank will simultaneously receive the proper nutrients from the nutrient tank. Collectively, these occurrences aid in controlling the pH of the tank's solution. Raspberry Pi is in charge of applying machine learning algorithms to recognize plans from the other end, which will then control plant growth and ultimately again

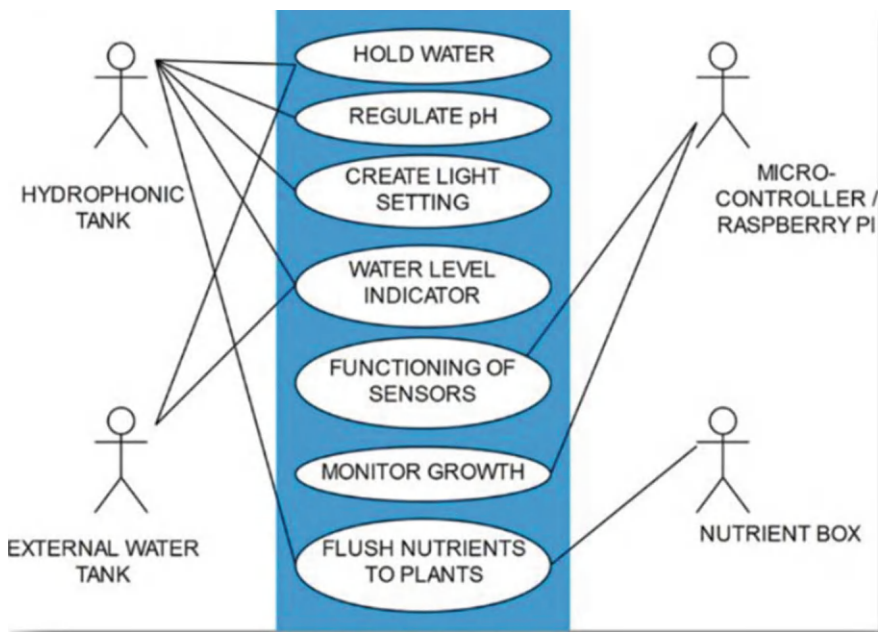


Fig. 2 This diagram demonstrates the responsibilities of different components in the system and their contribution

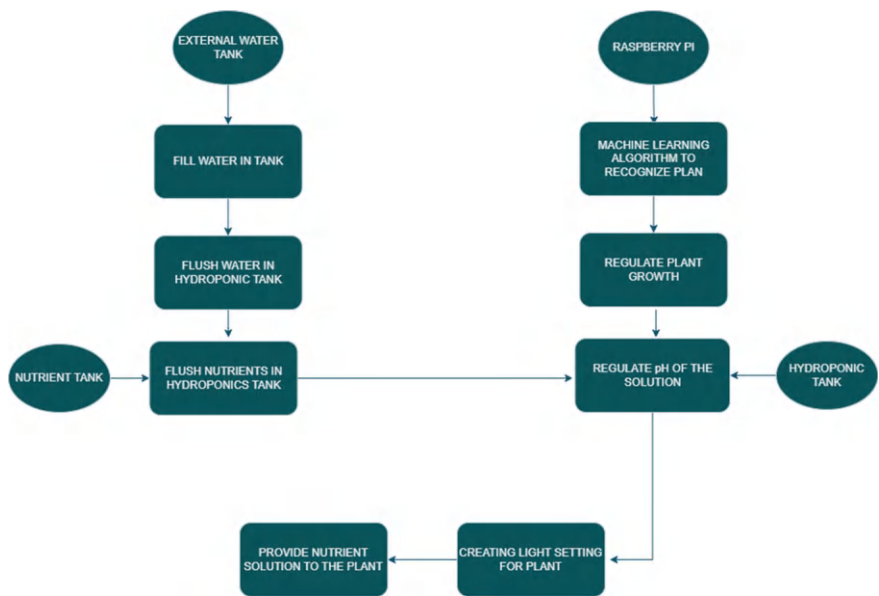


Fig. 3 Data flow diagram

2.3 Integration of IoT for Intelligent Hydroponics System

The field of intelligent farming tactics is revolutionized by the Internet of Things (IoT) integration into hydroponic systems. Important environmental characteristics like pH, temperature, humidity, and light intensity may be continuously monitored thanks to Internet of Things devices, which include sensors and actuators. The automated modifications that are made to hydroponic setups to maintain ideal growing conditions for plants are made possible by this real-time data collection. These data streams are transmitted to cloud-based platforms or centralized systems, where machine learning algorithms are able to analyze trends and patterns and make well-informed decisions for the best possible plant growth methods.

Furthermore, hydroponic system administration and accessibility from a distance are made possible by IoT connectivity. Growers may monitor and manage the system remotely via smartphones or other devices, allowing for timely interventions even when they are not physically there. Growers can guarantee accurate resource usage by supplying the precise amounts of light, water, and nutrients required for plant growth with IoT connectivity. In hydroponic settings, this improved precision reduces waste and optimizes yield potential.

3 Result and Analysis

Results from the research on improving hydroponic efficiency with IoT and machine learning algorithms were encouraging. Valuable insights were achieved by using deep neural networks to analyze hydroponic factors for controlling operations. The 5000 real-time records that made up the training dataset included crucial variables like pH, temperature, humidity, light intensity, and water level. 10,000 training epochs were used to train the input, which produced appropriate control actions labeled from 0 to 7. On a Raspberry Pi3, the trained model was installed, and the output was sent to an Arduino. A cloud-based database (Firebase) was used to store both the gathered data and the anticipated results in order to ensure data storage and retrieval.

A series of six trials on tomato plants were carried out, with each trial yielding a different result, to assess the efficacy of the hydroponic system. The plants' height and the number of days since the trials' commencement were the main comparison criteria. The plants in the initial trial grew to an 85 cm height by day 60.

Up to the final trial, where it reached roughly 110 cm on the 60th day, the plants' height ranged during the succeeding trials between 80 and 100 cm. These findings showed that the neural network improved over time as a result of incremental learning from each session. Its effectiveness in predicting and managing hydroponic parameters was further validated by the model's accuracy, which was found to be 88.5% (Fig. 4).

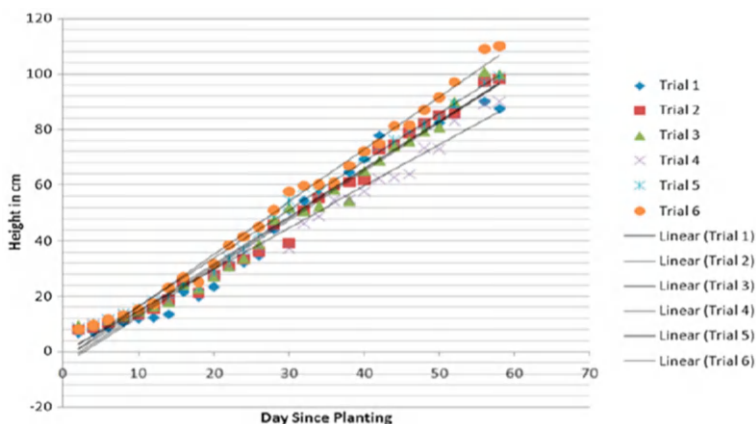


Fig. 4 Line graph of plants growth

A side-by-side comparison of tomato plants grown hydroponically and in conventional soil was done. The study's surprising results highlighted clear benefits of the hydroponic method.

Similar growth patterns were visible in both sets of plants throughout the first 12 days. But as the tests went on, the hydroponically produced plants showed more growth than their soil-grown counterparts. Notably, the hydroponic plants grew more than 50% faster than the soil-grown plants between the 30th and 32nd day. The hydroponic plants had grown to a height of roughly 95 cm by the 60th day of the trials, but the soil-grown plants had only grown to a height of 65 cm.

These findings show that hydroponics has a big advantage over conventional soil-based systems. Deep learning algorithms are incorporated, allowing for precise control of hydroponic settings and improved plant development. This study demonstrates how hydroponics may be used to reduce the negative environmental effects of conventional agriculture methods. IoT and machine learning algorithms can be used to improve hydroponic systems, enabling efficient and sustainable food production (Fig. 5).

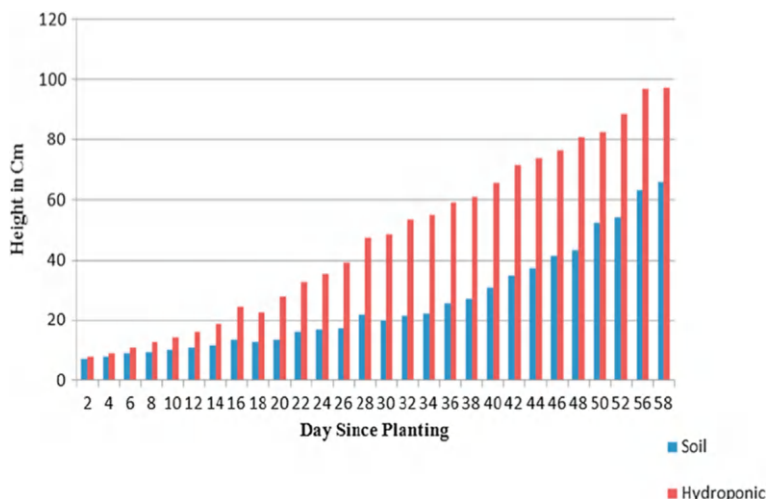


Fig. 5 Bar chart comparing plant growth in soil versus hydroponic systems

4 Conclusion

The economy of the nation is significantly influenced by agriculture. A KNN machine learning-based automated irrigation system has been developed, with intelligence deployed at the edge to predict soil condition based on moisture and irrigate the field. The Internet of Things (IoT) has made it possible for machines to communicate with one another. Hydroponics and aeroponics, which enable plants to live in a range of situations, such as hanging or hung, without the usage of soil, are currently the focus of agricultural breakthroughs. Such a technique has the drawback of requiring manual plant monitoring in order to spray the right nutrients for efficient growth.

As a result, several IoT-based hydroponic monitoring systems with mobile applications for control have been developed with the advent of IoT. Some research has been done utilizing artificial neural networks and Bayesian networks to apply intelligence in machine learning to the data set that was obtained for managing the parameters for proper hydroponic plant development. In light of all of this, in order to grow a proper plant in hydroponics, we need to autonomously regulate the hydroponic system's parameters without the need for human interaction.

Therefore, we developed an intelligent IoT-based hydroponic system utilizing the tomato plant as a case study, where pH, temperature, humidity, level, and illumination are five factors that control the hydroponic environment. To offer the right control action, which is labeled, the deep neural network is employed to train these parameters. To reach an accuracy of 88% in predicting the proper output action, these attributes are acquired in real-time over a period of several weeks and trained 10,000 times.

The planned control action for real-time data is stored in the cloud. The deep neural network model is used to forecast output and connect with the Arduino on the Raspberry Pi3 which acts as the edge. As a proof of concept, this was made. In the future, the system might be expanded by introducing a deep neural network-based intelligent IoT-based hydroponic system for additional hydroponically grown plants, allowing for greater precision. More hydroponic plants could be grown in additional tanks, and the system could be enlarged by doing so. The parameters could then be adjusted to offer the appropriate control action utilizing intelligence. This may be carried out on a large scale, enabling the precision of the system to be determined for a variety of hydroponic plants.

References

1. Banerjee T, Roy S (2019) Deep neural network fitting model for hydroponics system. In: 2019 10th international conference on computing, communication and networking technologies (ICCCNT). IEEE, pp 1–6
2. Deepak B, Singh R (2021) IoT-based smart agriculture: a review on smart hydroponic systems. In: 2021 5th international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB). IEEE, pp 1–6
3. Goyal S, Sahu S (2020) IoT-based smart hydroponics system for optimal plant growth. In: 2020 international conference on emerging smart computing and informatics (ESCI). IEEE, pp 47–51
4. Zhang Y, Yang J, Zhang M, Zhang C (2019) Application of internet of things and deep learning in hydroponics. In: 2019 IEEE international conference on artificial intelligence and computer applications (ICAICA). IEEE, pp 172–176
5. Remya V, Kumar AS (2020) IoT and machine learning-based smart hydroponic system for efficient plant growth. In: 2020 IEEE international conference on communication and electronics systems (ICCES). IEEE, pp 1585–1590
6. Rodríguez-Díaz JA, Andrade-Aréchiga M, Muñoz-Romero S, Rodríguez-Jiménez GC (2020) Development of a smart agriculture system based on IoT and machine learning. IEEE Access 8:136219–136228
7. Sun S, Gao Y, Zhang Y, Yu J (2021) Design of IoT-based hydroponic system and research on environmental perception. In: 2021 7th international conference on control, automation and robotics (ICCAR). IEEE, pp 381–386
8. Nair SV, Pillai VK (2020) IoT based precision agriculture monitoring system for sustainable crop production. In: 2020 11th international conference on computing, communication and networking technologies (ICCCNT). IEEE, pp 1–5
9. Deng Y, Xu Y, Peng H (2020) Research on the integration of internet of things and hydroponics. In: 2020 IEEE 4th information technology, networking, electronic and automation control conference (ITNEC). IEEE, pp 1935–1938
10. Sharma S, Sharma D, Kumar S, Singh K (2019) Smart hydroponics farming system using IoT and machine learning. In: 2019 5th international conference on computing, communication, control and automation (ICCUBEA). IEEE, pp 1–6
11. Sridharan S, Vishnu R, Anand MM, Narayanasamy P (2021) Design and development of IoT based smart agriculture system using machine learning techniques. In: 2021 international conference on inventive systems and control (ICISC). IEEE, pp 1306–1311
12. Das S, Choudhury T, Biswas S (2020) IoT-based automated nutrient control system for hydroponics. In: 2020 international conference on inventive research in computing applications (ICIRCA). IEEE, pp 1–6

13. Zhang M, Zhang Y, Zhang C (2019) Study on the application of internet of things and machine learning in hydroponics. In: 2019 international conference on advanced computer science and information systems (ICACSIS). IEEE, pp 409–414
14. Thomas N, Prakash A (2020) Implementation of IoT and machine learning techniques in hydroponics. In: 2020 11th international conference on computing, communication and networking technologies (ICCCNT). IEEE, pp 1–6
15. Dong H, Peng C (2020) Design and implementation of intelligent hydroponic system based on IoT. In: 2020 3rd international conference on robotics and automation engineering (ICRAE). IEEE, pp 156–160
16. Manasa MV (2020) IoT-based hydroponics monitoring and control system. In: 2020 5th international conference on automation, control and robotics engineering (CACRE). IEEE, pp 91–96
17. Huang Y, Liu Z (2020) Design of hydroponic system based on internet of things and machine learning. In: 2020 3rd international conference on robotics, control and automation (ICRCA). IEEE, pp 91–96
18. Bera S, Mandal S, Chatterjee R (2020) IoT-based smart farming system using deep learning and hydroponics. In: 2020 10th international conference on cloud computing, data science & engineering (Confluence). IEEE, pp 255–260
19. Sharma R, Latha R (2019) An IoT based smart hydroponic farming system using deep learning. In: 2019 3rd international conference on trends in electronics and informatics (ICOEI). IEEE, pp 692–697
20. Anirudh P, Nandhin R, Divyesh R, Rajarajan V, Sampathkumar A (2020) IoT based smart hydroponics for sustainable agriculture. In: 2020 international conference on emerging trends in communication, control and computing (ICONC3). IEEE, pp 1–5

Index

A

Acute kidney injury, 397
Adaptive Savitzky–Golay filters, 193, 195, 200, 215
Adaptive weighted frost filter, 491, 505
Advertisement, 187, 188, 191
Agriculture, 27–30, 37, 111, 113, 139, 140, 235, 236, 247, 429–431, 433, 434, 465, 468, 561–563, 568, 569
Air quality index, 139, 141, 151, 198, 200
Air quality prediction, 146, 193, 195–197, 204, 215, 216
Alzheimer’s Disease (AD), 251–262, 264–267, 320
Amazon, 219, 220, 232, 340, 341, 344–348, 384
Artificial intelligence, 18, 108, 112, 124, 188, 235, 281, 287, 289, 325, 331, 397, 399, 400, 431, 480, 485, 492, 538, 562
Artificial neural network, 20, 39–42, 46–48, 84, 178, 247, 300, 383, 401, 457, 482, 487, 569
Audio data analysis, 298
Audio modality, 299, 306–308
Autism spectrum disorder, 443–445

B

Black rot, 465, 466, 468, 477
Black spots, 393
Bone fracture, 479–483, 485–488
Brain tumor, 491–494, 496, 503–505, 525, 526, 528, 529, 531, 532, 534

Breast cancer, 39–42, 46, 50, 51, 67–70, 72–77

C

Campus safety, 381, 382, 386, 393, 394
Cancer, 39, 40, 45, 67–69, 455–464, 526, 534
Chronic diseases, 400, 537, 539, 541
Chronic kidney disease, 397–403, 405, 407–410, 541
CNN (ConvNet), 41, 148, 154, 170, 219, 221, 225–227, 230–232, 236–238, 240, 247, 269–275, 277, 278, 297, 301, 304–309, 327, 331, 332, 351–353, 358, 361, 365, 368–374, 376, 384, 400, 444, 445, 447, 455, 456, 459, 469, 471, 481–483, 486, 488, 492, 494, 503, 509–521, 523, 526–534
CNN-LSTM hybrid model, 221, 270, 274, 277, 278
Cognitive psychology, 281, 289
Colab, 15–17, 45, 372
Computer vision, 112, 127, 128, 130, 271, 351, 352, 360, 362, 367–369, 385, 480, 485, 516, 518, 538
Consumer sentiment analysis, 219
Convolutional Neural Network (CNN), 41, 67, 69, 70, 72, 74–77, 113, 114, 116, 129–132, 135, 169, 170, 196, 204, 206, 219, 226, 227, 269, 270, 272, 274, 297, 308, 313, 342, 352, 353, 368, 372, 400, 401, 434, 443, 445, 447, 450, 468–470, 483, 504, 509,

- 510, 512–517, 519, 526, 529, 532, 534
- CT scans, 455, 458, 459, 463, 464, 481, 483, 526, 528
- D**
- Deception detection, 299–301, 305–309
- Decision Tree (DT), 1–6, 11, 20–22, 41, 44–46, 48, 50, 69, 70, 81, 84, 87, 88, 90, 92–94, 97, 99, 103–107, 176–178, 180, 184, 284, 292, 304, 307, 314, 339, 341, 343–345, 348, 403, 405–407, 432, 434–437, 439, 456, 479, 485, 486, 488, 516, 537, 538, 541, 543, 545, 547, 551
- Deep learning, 3, 13, 18, 40, 41, 55, 56, 59, 61, 63, 64, 68–70, 72, 83, 84, 97–100, 102, 113, 127–129, 131, 135, 139, 148, 162, 171, 193–195, 197, 204, 215, 220, 222, 232, 237, 238, 244, 269–271, 276, 278, 286, 290, 300, 301, 308, 309, 314, 326–328, 341, 352, 353, 358, 360, 361, 367–370, 378, 383, 384, 389, 397, 400, 402–405, 427, 434, 444, 445, 456, 465, 467, 469, 480–483, 486–488, 492–494, 502, 505, 510–514, 516, 518–521, 529, 531, 568
- Deep neural network, 161, 162, 164, 166, 168, 171, 202, 314, 385, 389, 399–403, 468, 539, 564, 567, 569, 570
- Depth-wise separable convolutional neural network, 139, 144, 148
- Diabetes, 81–89, 92–94, 319–322, 324, 325, 330, 397, 404, 414
- Diabetes mellitus, 82, 398
- Diabetic retinopathy, 319–330, 332, 333
- Dog breed, 509–517, 521–523
- Dragonfly, 537–541, 543, 545, 547, 548
- Dropout, 73, 116, 166, 168, 242, 243, 283, 305, 413–427, 499
- Duo attention, 139, 144, 147, 148
- E**
- E-commerce, 339–342
- EEG data, 16, 17, 287
- Efficientnet, 241, 365, 444, 445, 447
- Ensemble, 6, 20, 42, 44, 53, 59, 60–63, 67, 69, 73–77, 84, 89, 100, 103, 115, 116, 118, 119, 124, 145, 156, 157, 177, 180, 269, 276, 278, 279, 283–286, 290–292, 326, 402, 403, 405–407, 427, 432, 434, 435, 456, 459, 460, 539, 552
- Ensemble learning, 31, 44, 53, 71, 77, 89, 103, 170, 171, 290, 403, 405, 406, 474, 516
- Ensemble models, 62, 63, 67, 69, 73–76, 100, 276, 277, 281, 402, 403, 405, 481, 550
- Entity detection, 549
- ESCA, 465, 466, 468, 477
- European night crawler optimization, 491, 500
- Evaluation metrics, 61, 74, 107, 124, 167, 170, 173, 186, 244, 521, 526, 547
- F**
- Faster RCNN, 369, 370, 378, 379, 488
- Feature engineering, 27, 59–61, 84, 99, 102, 161, 173, 175, 177, 194, 306–308, 387
- Feature vector, 56, 130, 161, 165, 166, 204, 241, 276, 518, 520, 521
- Fire detection, 269–278
- Football fake news detection, 97
- Fracture detection, 479, 481, 482, 485–488
- Functional Magnetic Resonance Imaging (fMRI), 251–253, 258, 261, 265, 267
- H**
- Haar Cascade, 443–446, 450
- Healthcare, 51, 67, 68, 81, 83, 94, 108, 176, 191, 293, 383, 398, 399, 402, 403, 407, 409, 410, 443, 444, 450, 455, 480, 482, 483, 487, 492, 526, 537, 539
- HindiBERT, 53, 55, 56, 59, 60, 62–64, 99
- Humidex value, 139, 140, 150, 152, 155–157
- Hydroponic plant growth, 561
- Hydroponics, 561–565, 567–570
- Hyperparameter tuning, 81, 88, 89, 93, 168, 196, 271, 272, 406, 417, 418, 445, 520
- I**
- Image captioning, 127–129, 131, 134, 135
- Image description, 127, 128, 137
- Image processing, 235, 236, 238, 360, 367, 372, 479, 480, 482, 486

Inception, 113, 130, 214, 238, 269, 279, 322, 353, 369, 445, 512–514, 517–520, 526, 528–532
 InceptionV3, 113, 238, 271, 326, 360, 486, 509, 511, 513, 514, 516, 525, 527, 528, 530
 Innovative agricultural strategy, 561
 Interaction degree, 413, 417–425, 427
 Internet of Things (IoT), 11, 22, 146, 383, 561–563, 565, 567–570

K

KNN classification, 28

L

Latent Dirichlet allocation, 549–551, 554, 558
 Leaf blight, 112, 113, 465, 466, 468, 477
 Limited receptive field mechanism, 491, 505
 Logistic regression, 20, 21, 39–41, 43, 46–48, 50, 68–70, 81, 83, 84, 87, 88, 91–94, 99, 342, 343, 348, 349, 384, 400, 413, 414, 417, 419, 425–427, 456, 513, 514, 539
 Long Short-Term Memory (LSTM), 69, 70, 98, 99, 154, 177, 185, 197, 213, 221, 230, 269–272, 274, 275, 277, 278, 299, 300, 331, 340, 341, 344, 345, 349, 384, 455, 456, 459, 463, 464, 551
 Long short-term memory network, 69, 219, 221, 222, 227, 231, 232
 Loss function, 20, 29, 60, 72, 104, 116, 117, 130, 132, 166, 167, 180, 237, 242, 271, 357, 368, 369, 389, 414, 416, 427, 459, 468, 470, 499, 521

M

Machine learning, 1, 3, 5, 11, 13, 14, 16–18, 21, 27, 29, 30, 32, 37, 39–41, 43–45, 48, 50, 53–55, 59, 60, 64, 68–70, 81, 83, 84, 86–88, 94, 97, 113, 168, 174–180, 187–189, 191, 194, 242, 244, 283–286, 289, 290, 293, 298, 299, 306, 309, 312, 313, 315, 339, 340, 342–344, 346–349, 360, 381, 383–388, 393, 397, 399, 400, 402–404, 406, 413, 429, 431, 432, 435, 439, 444, 445, 456, 457, 469, 479, 485–488, 515, 516, 520,

525–529, 534, 537–539, 541, 543, 545, 551, 552, 559, 562–564, 569
 Machine learning algorithms, 1, 13, 20, 27–29, 31, 41, 42, 48, 70, 84, 87, 94, 173–175, 180, 271, 290, 293, 342, 384, 388, 431, 432, 434, 482, 483, 486, 487, 516, 526, 537–539, 547, 551, 559, 561, 564, 565, 567, 568
 Major Depressive Disorder (MDD), 281–289, 293
 Malicious software, 161–166, 168–170
 MobileNetV2, 269, 270, 276, 278, 279, 527, 528
 Mobile network, 193, 195, 202, 203, 213, 215, 351, 353–355, 365, 468, 526, 528, 529, 532
 Momentum, 115, 116, 333, 413, 414, 416–425
 M-squared normalization, 139, 147
 Multidimensional, 73, 359, 367–369, 373, 378, 516
 Multiple linear regression, 187, 189–191, 432
 Multi-Scale Residual network, 491, 505

N

Naive Bayes, 69, 97, 98, 104–107, 177, 313, 340, 342, 343, 345, 346, 537, 538, 543, 545
 Naive Bayesian, 339, 341
 Named Entity Recognition (NER), 315, 331, 549–552, 555, 556, 559
 Natural Language Processing (NLP), 29, 53–55, 57, 59, 86, 128, 130, 162, 220, 224, 290, 298, 311, 312, 314–317, 339–342, 344, 369, 551, 552, 555
 Natural Language Toolkit (NLTK), 311, 312, 315
 Nipah, 1–3, 6, 8, 11
 Nutrient solutions, 561, 563

O

Object detection, 113, 272, 299, 351–354, 356, 360–362, 365, 367–371, 373–378, 470, 471, 473, 475, 481, 486, 549
 Operation code adjacency graph, 161, 163–165, 169
 Optimizer, 104, 115, 168, 242, 243, 271, 389, 413, 414, 416–422, 424, 425, 427, 470, 521, 530

P

Pima Indian Diabetes Dataset, 81–84, 94
 Plant disease diagnosis, 113
 Population classification, 163, 166, 168–170
 Potato plant, 247
 Poultry farm egg production, 144
 Prediction, 1–6, 8, 10, 13, 15, 17–21, 27–29, 31, 32, 37, 40–42, 44, 47, 50, 57, 60, 63, 67, 68, 70, 73–76, 103, 139, 141, 144–148, 150, 153–157, 167, 177, 179, 180, 183, 188, 193, 195–197, 215, 227, 232, 244, 270, 279, 283–286, 290, 292, 293, 298, 305, 315, 329, 353–359, 368, 384, 385, 389, 390, 397, 400–403, 406, 408–410, 415, 426, 427, 429–431, 433–436, 439, 443, 445, 455–457, 459–464, 470, 472, 499, 510, 516, 521, 538, 539, 543, 552, 563
 Prediction analysis, 156, 157
 Predictive policing, 382
 Preprocessing, 57, 59, 70, 71, 83, 100, 130, 132, 193, 195, 200, 215, 219–221, 223, 225, 232, 271, 315, 326, 343, 349, 354, 358, 387, 388, 446, 455, 459, 469, 483, 487, 488, 491, 493, 494, 496, 505, 516, 517, 529, 556
 Python-Keras, 352

R

Random Forest (RF), 1–6, 11, 20–22, 29, 31, 37, 39, 41, 50, 56, 68, 70, 81, 83, 84, 87, 89, 91–93, 97, 99, 100, 103, 105–107, 177, 178, 185, 271, 284, 292, 293, 299, 304, 307, 308, 313, 343, 344, 348, 383, 400, 402, 403, 405–407, 410, 432, 435, 436, 439, 456, 479, 486, 516, 537, 538, 541, 543, 545, 547
 Random forest classifier, 39–41, 44–48, 50, 55, 56, 103, 106, 107, 177, 406, 439
 Rat optimisation algorithm, 219
 Real-time surveillance, 385
 Recruitment, 550–552, 554, 558
 Red time zones, 393
 Region-Based Convolutional Neural Network (RCNN), 367, 369, 370, 373–375, 378, 379, 488
 Regression models, 21, 147, 191, 373, 555
 ResNet50, 269, 272, 279, 327, 511–514, 525, 527, 528, 530, 531
 ResNetEnsemble, 111, 116–119, 123, 124

Restricted Boltzmann Machine (RBM), 3
 Resume evaluation, 549–552, 557–559
 Rice leaf diseases, 468
 Ridge classification, 27, 30, 36
 RoBERTa, 311, 312, 316, 317

S

Sales and social media, 187
 Sarcasm, 53–57, 60, 61, 63, 64
 Security systems, 270, 382
 Seizures prediction, 14
 Semantic gap, 127, 128
 Sentiment analysis, 54–57, 219, 220, 222, 223, 225, 300, 317, 339–344, 346–349
 Single Stage Detector (SSD), 351–358, 360, 365
 Social media, 54, 57, 61, 63, 97–99, 108, 187–189, 191, 219, 220, 312, 313, 315, 394
 Software library, 252
 Soil-less cultivation, 561
 SpaCy, 549, 555
 Speech analysis, 298
 Stanza, 311, 312, 316, 317
 Structural Magnetic Resonance Imaging (sMRI), 258, 261
 Support Vector Machine (SVM), 18, 19, 21, 22, 41, 42, 46, 50, 68–70, 81, 84, 87–89, 91–94, 97, 98, 103–107, 146, 176, 178, 185, 198, 222, 271, 273, 300, 304, 307, 313, 331, 339–346, 348, 349, 370, 373, 375, 383, 400–402, 439, 456, 457, 479, 481, 485–487, 493, 509, 512, 516–521, 523, 529, 537–539, 541, 543–545, 547, 551
 Support Vector Model (SVM), 342

T

TextRNN, 97, 99, 103, 105–108
 Transfer learning, 55, 235, 237, 271, 276, 427, 447, 481, 493, 494, 510, 511, 513, 514, 518, 528, 529, 531
 TuberNet, 235, 236, 240, 245, 247

V

Verbal cues, 298
 VGG, 130–132, 214, 237, 313, 445, 527–529, 531, 532, 534
 VGG16 and DenseNet201, 525

Virus colony search optimisation, [193](#), [209](#),
[214](#), [216](#)
Vision, [127](#), [128](#), [130](#), [146](#), [235](#), [272](#), [319](#),
[320](#), [330](#), [333](#), [365](#), [376](#), [502](#)

W

Water quality index, [139](#), [141](#), [151](#)

X

Xception, [269–271](#), [276](#), [278](#), [279](#), [468](#),
[493](#), [509](#), [511–514](#), [516–520](#),
[527–529](#)

XGBoost, [60](#), [69](#), [71](#), [72](#), [173](#), [175–180](#),
[185](#), [186](#), [284](#), [286](#), [299](#), [304](#), [307](#),
[400](#), [402](#), [439](#)

XG-Boost Classifier, [44](#)

X-ray, [479–483](#), [485–488](#), [496](#)

Y

YOLO-NAS, [468–471](#), [474](#), [475](#), [477](#)

YOLOv5, [465](#), [468–470](#), [474](#), [476](#), [477](#)

YOLOv8, [465](#), [469–471](#), [474](#), [476](#), [477](#)

You Only Look Once (YOLO), [351–353](#),
[365](#), [367–370](#), [372](#), [376–378](#), [465](#),
[468–471](#), [473–475](#), [477](#)