

DATA COMMUNICATION SERIES

BIG DATA SECURITY GOVERNANCE AND PREVENTION

Traffic Anti-Fraud in Practice

KAI ZHANG, ZE YANG,
LIYANG HAO, and QI XIONG



CRC Press
Taylor & Francis Group



人民邮电出版社
POSTS & TELECOM PRESS

Big Data Security Governance and Prevention

This book provides a practical reference for traffic anti-fraud, establishing a new standard for accessible, real-world traffic security governance that empowers readers to design scalable defenses while maintaining an optimal user experience.

The internet's rapid growth has enabled a surge in digital fraud. Cybercriminals exploit every stage of online traffic, from fake promotion scams and bot-driven account fraud to "coupon hacking" during e-commerce sales and sophisticated phishing campaigns. These threats cost billions globally and demand urgent solutions to protect users and platforms. This practical guide demystifies traffic anti-fraud with a 12-chapter framework. It begins with foundational concepts and then dissects real-world fraud tactics, then focuses on data preparation and governance. Core chapters introduce cutting-edge tools, such as device fingerprinting, AI-powered anomaly detection, graph-based network analysis, and cross-modal threat fusion. The final chapter provides step-by-step strategies for building adaptive anti-fraud systems.

This exceptional resource is ideal for cybersecurity professionals, developers, researchers, and students interested in cybercrime prevention, risk governance, and big data security.

Kai Zhang is a principal engineer at Tencent with over a decade of experience in combating cyber-crimes. He has led security projects in game security protection, financial risk control systems, and anti-fraud architectures. His core expertise lies in big data security threat modeling.

Ze Yang is a researcher at Tencent dedicated to financial risk governance. He has developed AI-powered mechanisms to combat underground economy threats in payment ecosystems.

Liyang Hao is a researcher at Tencent focusing on behavioral security systems. He has designed real-time gambling/fraud intervention engines for social payment scenarios.

Qi Xiong is a principal engineer at Tencent with 15 years of experience in security architecture. He has spearheaded compliance-driven security solutions for fintech applications and mobile ecosystems.

Data Communication Series

Cloud Data Center Network Architectures and Technologies

Lei Zhang and Le Chen

Campus Network Architectures and Technologies

Ningguo Shen, Bin Yu, Mingxiang Huang and Hailin Xu

Enterprise Wireless Local Area Network Architectures and Technologies

Rihai Wu, Xun Yang, Xia Zhou and Yibo Wang

Software-Defined Wide Area Network Architectures and Technologies

Cheng Sheng, Jie Bai and Qi Sun

SRv6 Network Programming

Ushering in a New Era of IP Networks

Zhenbin Li, Zhibo Hu and Cheng Li

IPv6 Network Slicing

Offering New Experience for Industries

Zhenbin Li, Jie Dong, Yawei Zhang, and Xinzong Zeng

Guide to SRv6 Network Deployment

Edited by Minwei Jin, Zhenbin Li, Lanjun Luo, Weidong Li

IPv6 On-Path Telemetry

Driving New Transformation in Network O&M

Zhenbin Li, Pingan Yang, Tao Han, and Tianran Zhou

Big Data Security Governance and Prevention

Traffic Anti-Fraud in Practice

Kai Zhang, Ze Yang, Liyang Hao, and Qi Xiong

For more information on this series, please visit: <https://www.routledge.com/Data-Communication-Series/book-series/DCSHW>

Big Data Security Governance and Prevention

Traffic Anti-Fraud in Practice

Kai Zhang, Ze Yang, Liyang Hao, and Qi Xiong



CRC Press

Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business



人民邮电出版社

POSTS & TELECOM PRESS

First edition published 2027
by CRC Press
2385 NW Executive Center Drive, Suite 320, Boca Raton FL 33431

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

© 2027 Kai Zhang, Ze Yang, Liyang Hao, and Qi Xiong
Translated by DeepL

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access www.copyright.com or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. For works that are not available on CCC please contact mpkbookspermissions@tandf.co.uk

For Product Safety Concerns and Information please contact our EU representative GPSR@taylorandfrancis.com. Taylor & Francis Verlag GmbH, Kaufingerstraße 24, 80331 München, Germany.

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

English Version by permission of Posts and Telecom Press Co., Ltd.

ISBN: 978-1-041-25535-2 (hbk)
ISBN: 978-1-041-36323-1 (pbk)
ISBN: 978-1-042-00212-2 (ebk)

DOI: 10.1201/9781042002122

Typeset in Minion
by codeMantra

Contents

Preface to the Chinese Edition, x

CHAPTER 1 ■ Introduction	1
1.1 THE EVOLUTION OF INTERNET TRAFFIC	1
1.1.1 PC Era	2
1.1.2 Mobile Internet Era	3
1.1.3 The Era of Cloud Computing and Big Data	5
1.2 TRAFFIC FRAUD IN THE BIG DATA ERA	5
1.3 CHALLENGES IN TRAFFIC FRAUD IN THE BIG DATA ERA	6
1.3.1 Regulatory Level	6
1.3.2 Industry Level	7
1.3.3 Business Layer	7
1.4 TRAFFIC ANTI-FRAUD SYSTEM ARCHITECTURE	8
1.4.1 Evolution of Traffic Anti-Fraud Solutions	8
1.4.2 Architecture of Traffic Anti-Fraud Systems	10
1.5 SUMMARY	12
BIBLIOGRAPHY	13
CHAPTER 2 ■ Traffic Fraud Tactics and Their Impact	14
2.1 PRE-TRAFFIC PHASE TRAFFIC FRAUD	15
2.1.1 Promotion Settlement Fraud	15
2.1.2 Registration Fraud	19
2.1.3 Login Fraud	20
2.2 MID-TRAFFIC PHASE FRAUD	22
2.2.1 “Wool-Pulling” Fraud	22
2.2.2 Brushing Volume Fraud	24
2.2.3 Traffic Diversion Fraud	24

2.3	POST-TRAFFIC PHASE TRAFFIC FRAUD	28
2.3.1	Telecom Fraud	29
2.3.2	Resource Monetization Fraud	30
2.4	ANALYSIS OF FRAUDULENT GAINS	32
2.5	SUMMARY	33
CHAPTER 3 ■ Traffic Data Governance and Feature Engineering		34
3.1	BASIC DATA FORMATS	34
3.1.1	Pre-Traffic Phase Data	34
3.1.2	Mid-Traffic Phase Data	36
3.1.3	Post-Traffic Phase Data	36
3.1.4	Traffic Data Characteristics Comparison	37
3.2	DATA GOVERNANCE	39
3.2.1	Data Collection	39
3.2.2	Data Cleansing	40
3.2.3	Data Storage	41
3.2.4	Data Processing	42
3.3	FEATURE ENGINEERING	43
3.3.1	Feature Construction	43
3.3.2	Feature Evaluation and Feature Selection	44
3.3.3	Feature Monitoring	46
3.4	SUMMARY	47
CHAPTER 4 ■ Device Fingerprinting Technology		48
4.1	THE VALUE OF DEVICE FINGERPRINTING	48
4.2	TECHNICAL PRINCIPLES	50
4.2.1	Basic Concepts	50
4.2.2	Development History	51
4.2.3	Generation Methods	52
4.3	TECHNICAL IMPLEMENTATION APPROACH	54
4.3.1	Evaluation Metrics	54
4.3.2	Feature Construction	55
4.3.3	Generation Algorithm	56
4.4	SUMMARY	60

CHAPTER 5 ■ CAPTCHA Verification	61
5.1 CAPTCHA VERIFICATION FUNDAMENTALS	61
5.1.1 The Birth of CAPTCHAs	61
5.1.2 Application Scenarios of CAPTCHAs	62
5.1.3 Framework for Building CAPTCHAs	63
5.2 FOUNDATIONAL SECURITY MEASURES	64
5.3 DESIGN-LEVEL OFFENSE AND DEFENSE	65
5.3.1 Text-Based CAPTCHA	66
5.3.2 Behavioral CAPTCHA	68
5.3.3 Next-Generation CAPTCHA	70
5.4 SUMMARY	71
CHAPTER 6 ■ Rules Engine	72
6.1 RISK LIST	72
6.1.1 Risk List Fundamentals	72
6.1.2 Evolution of Risk Lists in Defense and Offense	74
6.1.3 Risk List Deployment and Operations	75
6.2 GENERIC RULES	76
6.2.1 Evolution of IP	77
6.2.2 Evolution of Device Strategy: Offense and Defense	81
6.2.3 The Evolution of Account Strategy Offense and Defense	84
6.3 BUSINESS-CUSTOMIZED RULES	89
6.3.1 Rule Intelligent Preprocessing Module	90
6.3.2 Rule-Based Intelligent Construction Module	90
6.3.3 Rule Intelligent Screening Module	91
6.3.4 Other Modules	92
6.4 SUMMARY	92
CHAPTER 7 ■ Machine Learning Countermeasures	93
7.1 ZERO-SHOT SCENARIO	95
7.1.1 Traditional Statistical Testing Approach	96
7.1.2 Unsupervised Learning Approach	97
7.2 FEW-SHOT SCENARIO	100
7.2.1 Traditional Semi-Supervised Learning Approaches	100
7.2.2 Behavioral Sequence Learning Approach	103

7.3	MANY-SHOT SCENARIO	108
7.3.1	Binary Classification Models vs. Regression Models	109
7.3.2	Explainability Scenarios	109
7.3.3	Ensemble Models	113
7.4	SUMMARY	118
CHAPTER 8	■ Complex Network Adversarial Approaches	119
8.1	PRE-TRAFFIC PHASE SOLUTIONS	120
8.1.1	Detecting Syndicates based on Single-Dimensional Resource Aggregation	122
8.1.2	Detection of Groups Aggregating Multi-Dimensional Resources	125
8.2	MID-TRAFFIC PHASE STRATEGY	130
8.2.1	Structural Similarity Gang Detection	130
8.2.2	Fraud Resource Family Detection	134
8.3	POST-TRAFFIC PHASE SOLUTIONS	137
8.3.1	Graph Construction	138
8.3.2	Bidirectional PageRank Cross-Identification of Black Market Brokers	139
8.3.3	Fraudulent User Account Coloring Identification	141
8.4	SUMMARY	142
CHAPTER 9	■ Multimodal Integrated Countermeasure Scheme	143
9.1	SOURCES OF MULTIMODAL DATA	144
9.1.1	Relationship Graph Information	144
9.1.2	Textual Information	145
9.1.3	Image Information	146
9.1.4	Other Modal Information	146
9.2	MULTIMODAL FUSION APPROACH	147
9.2.1	Data-Layer Fusion	147
9.2.2	Feature-Layer Fusion	149
9.2.3	Decision-Layer Fusion	150
9.2.4	Hybrid Fusion Approach	153
9.3	SUMMARY	153

CHAPTER 10 ■ Novel Adversarial Approaches	154
10.1 FEDERATED LEARNING	154
10.1.1 Federated Learning Framework	155
10.1.2 Detection Effectiveness for Anomalous Traffic	157
10.2 KNOWLEDGE DISTILLATION	157
10.2.1 Knowledge Distillation Framework	158
10.2.2 Steps for Anomalous Traffic Detection	160
10.2.3 Effectiveness of Anomalous Traffic Detection	161
10.3 SUMMARY	161
CHAPTER 11 ■ Operational Framework	162
11.1 STABILITY OPERATIONS	164
11.1.1 Service Stability	164
11.1.2 Data Stability	168
11.2 FALSE-POSITIVE PREVENTION	169
11.2.1 Suspected Whitelist	171
11.2.2 Automated Operations	171
11.3 USER FEEDBACK HANDLING	172
11.3.1 User Appeal Handling	172
11.3.2 User Report Handling	173
11.4 ALERT HANDLING SYSTEM	174
11.5 SUMMARY	175
CHAPTER 12 ■ Knowledge Intelligence Mining and Application	176
12.1 BLACK/GRAY INDUSTRY GANG INTELLIGENCE MINING	176
12.1.1 “Freeloading” Intelligence	177
12.1.2 Comment Farming Intelligence	180
12.2 INTELLIGENCE MINING OF BLACK/GRAY INDUSTRY BEHAVIOR PATTERNS	183
12.2.1 Short Video Platform Account Farming	183
12.2.2 Ad Click Fraud	184
12.3 BLACK/GRAY INDUSTRY PRICE INTELLIGENCE MINING	185
12.4 SUMMARY	187

Preface to the Chinese Edition

As one of the first security teams involved in anti-fraud social governance, we consolidated our team's decade-long anti-fraud technology framework and practical experience in 2022. This culminated in the January 2023 publication of *Big Data Security Governance and Prevention: Building an Anti-Fraud System*. The book received widespread acclaim upon release. However, as an introductory textbook for big data security anti-fraud systems, it focuses on foundational concepts and general methodologies, leaving gaps in specific domains like traffic security and URL security.

As a practical guide in the field of traffic fraud, this book delves into the adversarial techniques and details encountered in real-world traffic fraud operations. It equips readers with the theoretical foundations of traffic security while enhancing their technical application and hands-on experience.

This book is divided into 12 chapters. Chapter 1 covers the evolution of internet traffic, traffic fraud challenges in the big data era, countermeasures against traffic fraud, and the architecture of traffic fraud prevention systems. Chapter 2 details traffic fraud tactics and their impacts. Chapter 3 addresses traffic data governance and feature engineering. Chapters 4–10 explore foundational techniques and countermeasures for practical traffic fraud prevention. Chapters 11–12 focus on operational frameworks and the mining and application of knowledge intelligence.

Traffic fraud prevention is a critical area within big data security. The successful synthesis and systematization of relevant technologies and frameworks owe much to the power of teamwork. Beyond the primary authors, the following three contributors also played significant roles in this book's development.

- Yang Ze authored Chapter 3, "Traffic Data Governance and Feature Engineering"; Chapter 7, "Machine Learning Countermeasures"; Chapter 9, "Multimodal Integrated Countermeasures"; Chapter 10, "Novel Countermeasures"; and Chapter 11, "Operational Framework."
- Hao Liyang authored Chapter 2, "Traffic Fraud Methods and Their Impact"; Chapter 4, "Device Fingerprinting Techniques"; and Chapter 12, "Knowledge Intelligence Mining and Application."
- Xiong Qi provided constructive guidance on the book's thematic focus, direction, and content.

As this manuscript nears completion, we extend special gratitude to numerous friends. Li Ning provided robust support from a project management perspective, facilitating the writing process, securing resources, and handling post-production matters. Cai Chaowei offered numerous constructive revisions based on years of practical experience in the anti-fraud industry and technology implementation. We also thank editor Shan Ruiting of People's Posts and Telecommunications Press for her comprehensive support throughout the publication process.

While we have made every effort to ensure the completeness and accuracy of the content during the writing process, due to limitations in our writing skills, oversights and shortcomings may still exist in the book. We sincerely welcome readers' criticism and suggestions.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Introduction

This chapter introduces fundamental concepts related to traffic anti-fraud, covering four key areas: the evolution of internet traffic, the problem of traffic fraud in the big data era, challenges in traffic anti-fraud, and system architecture. This lays the groundwork for subsequent chapters detailing comprehensive countermeasures against traffic fraud.

1.1 THE EVOLUTION OF INTERNET TRAFFIC

The internet originated in the 1960s, evolving from the Advanced Research Projects Agency Network—a small network established by the U.S. Department of Defense. Initially, the network comprised few nodes, primarily including the University of California, Los Angeles; the Stanford Research Institute; and the University of Utah. Data exchange and sharing among these nodes facilitated communication between military and research institutions.

As the early internet was a small network with limited users, traffic volumes were minimal. However, with continuous technological innovation, the internet gradually evolved into a massive, global shared network connecting “everything.” Connected devices expanded beyond PCs, tablets, and smartphones to include smart wearables like smartwatches and smart glasses. The massive influx of global users interacting and sharing information across the internet led to explosive growth in internet traffic. This traffic explosion did not occur overnight but unfolded over an extended developmental journey. This section specifically focuses on the evolution of China’s internet traffic. Based on the continuous expansion of China’s internet traffic scale, its development can be divided into three distinct phases: the PC internet era, the mobile internet era, and the cloud computing and big data era. The developmental timeline of China’s internet traffic is illustrated in Figure 1.1.

Each of these three phases exhibits distinct characteristics.

- **PC Internet Era:** Traffic carriers were websites and PC client software, with interaction models centered on user searches and browsing. Overall data volumes were relatively modest. Traffic fraud issues in this era mainly focused on traditional foundational security threats, such as malware, vulnerability exploitation, and phishing Trojans.

2 ■ Big Data Security Governance and Prevention

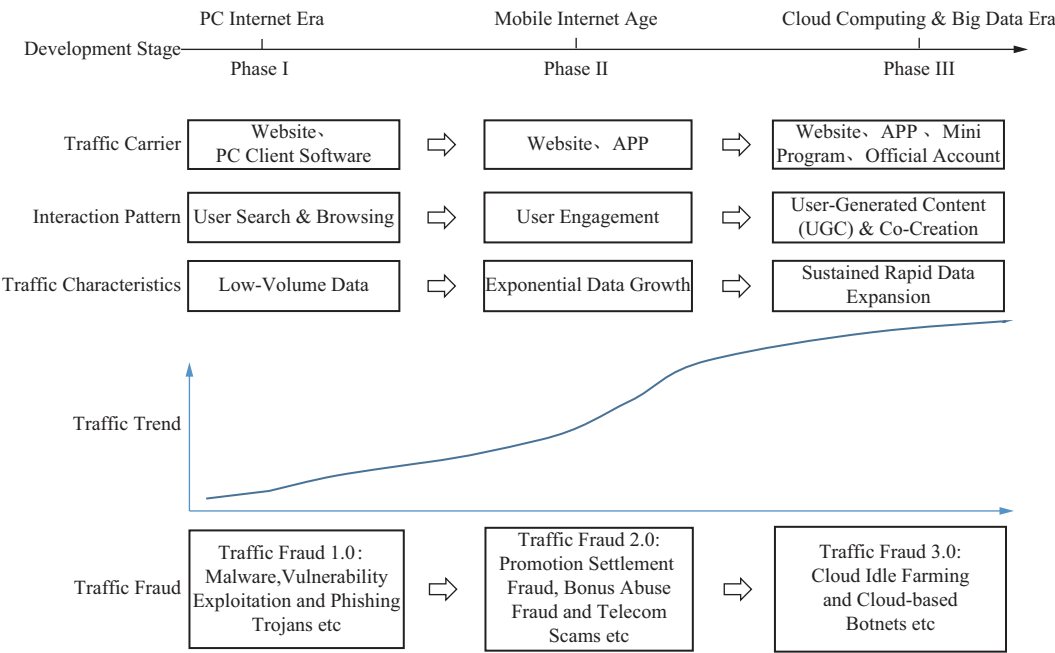


FIGURE 1.1 Evolution of China's internet traffic.

- Mobile Internet Era: Traffic carriers shifted to websites and apps, with interaction models increasingly centered on user participation, leading to explosive data growth. The diverse business models and rich application scenarios of this era gave rise to persistent traffic fraud issues, including promotion settlement fraud, “wool-pulling” scams, and telecom fraud.
- Cloud Computing and Big Data Era: New traffic carriers like public accounts and mini-programs emerged, alongside the rapid development of applications such as short videos. These provided users with more entry points for interaction and content creation at lower barriers, leading to sustained rapid data growth. Concurrently, traffic fraud saw the emergence of more convenient cloud-based black market tools, such as cloud idle farming and cloud-based botnets.

The following sections will detail the traffic characteristics and security challenges across three pivotal phases of China's internet traffic evolution.

1.1.1 PC Era

The first significant phase in China's internet traffic development was the PC internet era. During this period, major traffic carriers included portal websites like Tencent, Baidu, Sohu, and NetEase. Users primarily accessed information through search and browsing, engaging in one-way interactions. The interaction model of the PC internet era, as shown in Figure 1.2, lowered barriers to information access and enhanced dissemination efficiency. However, user engagement remained limited during this period. Users rarely participated

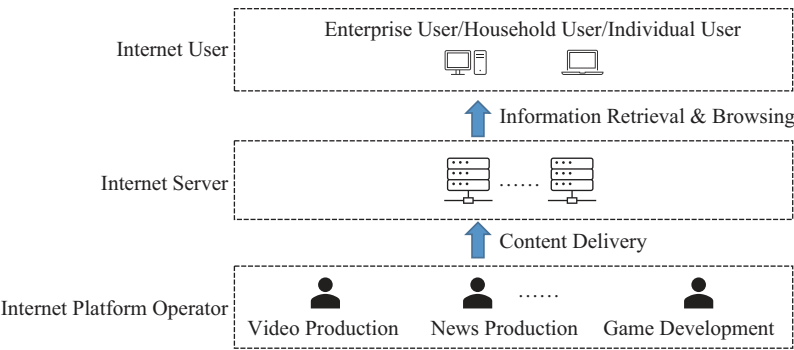


FIGURE 1.2 Interaction model in the PC internet era.



FIGURE 1.3 Three key nodes in the evolution of internet products during the PC internet era.

deeply in internet content creation, and the generated data primarily consisted of text with minimal image content. Additionally, internet penetration in China was low, resulting in a relatively small overall traffic scale. Traffic fraud issues during this phase primarily involved foundational security concerns like malware and vulnerability exploitation.

From the perspective of high user penetration rates, the evolution of internet products during the PC era underwent three key milestones, as illustrated in Figure 1.3. Search engines served as the foundational infrastructure of the PC internet era, connecting people with information. Consequently, the first products to achieve high user penetration were search engine portals like Baidu, establishing themselves as the primary entry points for internet traffic. Subsequently, as infrastructure matured, the focus shifted to connecting people. Social entertainment products like Weibo and QQ joined the ranks of high-penetration products. Finally, the critical node of connecting people with goods emerged, with e-commerce portals like Taobao and JD.com joining the ranks of high-penetration products.

1.1.2 Mobile Internet Era

With the widespread adoption of third-generation (3G) mobile communication technology networks and smartphones, high-speed data transmission gained stronger support, ushering in the second pivotal phase of internet traffic development—the mobile internet era. Key products emerging during this phase included migrations from PC platforms to mobile, such as QQ, Taobao, JD.com, and Baidu. As the era progressed, representative new products emerged, including WeChat, Meituan, Pinduoduo, and Douyin. Compared

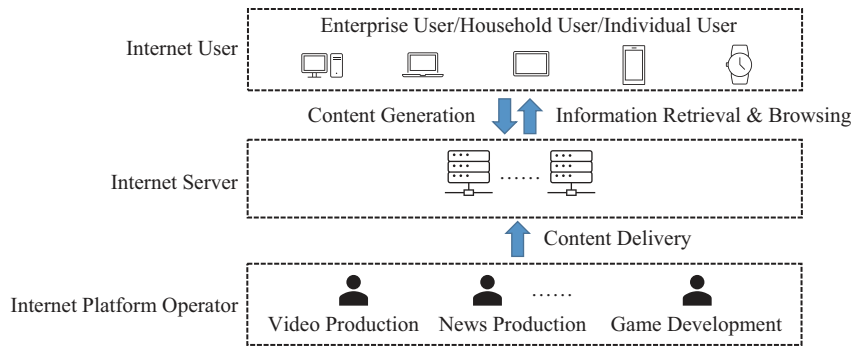


FIGURE 1.4 Interaction models in the mobile internet era.

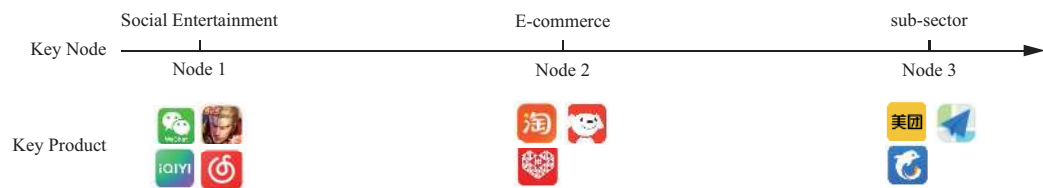


FIGURE 1.5 Three key milestones in the evolution of internet products during the mobile internet era.

to the first phase, mobile apps became a new traffic carrier. Users no longer engaged in one-way interactions with platforms; instead, they could share their lives and work anytime, anywhere based on personal preferences, forming two-way interactions with platforms. The interaction model of the mobile internet era is illustrated in Figure 1.4. Users generated vast amounts of user-generated content and professionally generated content during these interactions, leading to explosive growth in internet traffic. The data formats generated by users during this phase are primarily text and image-based, with a smaller amount of audio and video data. The main traffic fraud issues in this phase include business fraud such as promotion settlement fraud, “wool-pulling” fraud, telecom fraud, and payment fraud.

From the perspective of high user penetration rates, the evolution of internet products in the mobile internet era underwent three key milestones, as illustrated in Figure 1.5. Unlike the PC internet era, the products that first achieved high user penetration in this phase were social entertainment offerings such as social media, long-form video, music, and gaming. Among these, social products replaced search engines as the primary gateway to mobile internet traffic. Subsequently, e-commerce migrated from PC to mobile platforms, achieving further growth and entering the ranks of high-penetration products, with Taobao, JD.com, and Pinduoduo serving as representative examples. Finally, products in other niche sectors beyond social entertainment and e-commerce—such as food delivery, navigation, and travel—also attained high user penetration.



FIGURE 1.6 Three key milestones in the evolution of internet products in the cloud computing and big data era.

1.1.3 The Era of Cloud Computing and Big Data

With advancements in cloud computing and related technologies, the storage, computation, and application of massive internet data became feasible, ushering in the third pivotal phase of internet traffic growth: the cloud computing and big data era. The evolution of internet products during this era primarily unfolded through three key milestones, as illustrated in Figure 1.6. Products emerging at these milestones included live streaming, short videos, online conferencing, and online education. Compared to the previous phases, new traffic carriers such as mini-programs and public accounts emerged. Users in this phase engage not merely in simple interactions but actively participate in internet activities through self-generated content based on personal interests or profit motives. This generates massive data volumes that continue to grow exponentially. Beyond text and image data, this era produces unprecedented volumes of audio and video data, truly ushering in the big data era of the internet. Traffic fraud issues in this phase encompass not only existing business fraud problems from the mobile internet era—such as promotion settlement fraud, “wool-pulling” fraud, and telecom fraud—but also new types of fraud like cloud idle farming and cloud-based botnets, emerging alongside the development of cloud services. The following sections detail traffic fraud challenges in the big data era.

1.2 TRAFFIC FRAUD IN THE BIG DATA ERA

Internet traffic in the big data era is not only massive but also highly complex, accompanied by various types of black/gray industry fraud. Traffic fraud issues permeate the entire lifecycle of an app, as illustrated in Figure 1.7.

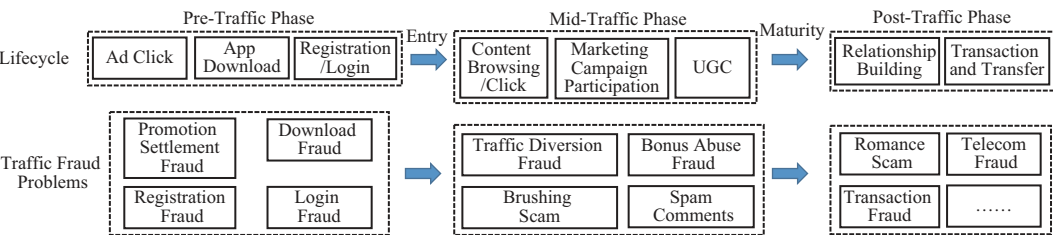


FIGURE 1.7 Traffic fraud in the big data era.

The following sections will examine traffic fraud from the perspectives of the pre-traffic phase, mid-traffic phase, and post-traffic phase. For specific methods and impacts of traffic fraud, please refer to Chapter 2.

1. Traffic Fraud in the Pre-Traffic Phase

The pre-traffic phase primarily involves four stages: ad clicks, app downloads, account registration, and account login. Traffic fraud issues arising in this phase mainly include promotion settlement fraud, download fraud, registration fraud, and login fraud.

2. Mid-Traffic Phase Traffic Fraud

The mid-traffic phase primarily involves user behavior after entering the app, such as content browsing, clicks, comments, participation in marketing activities, and sharing referral URLs. Traffic fraud issues arising during this phase mainly include traffic diversion fraud, “freeloading” fraud, fake likes/chart manipulation fraud, and spam comments.

3. Post-Traffic Phase Traffic Fraud Issues

The post-traffic phase primarily involves establishing social relationships between users and transaction transfers. The main traffic fraud issues arising in the post-traffic phase include “romance scams,” telecom fraud, and transaction fraud.

1.3 CHALLENGES IN TRAFFIC FRAUD IN THE BIG DATA ERA

Anti-fraud efforts face multifaceted challenges. From a high-level perspective, traffic fraud in the big data era involves complex challenges across regulatory, industry, and operational dimensions, as illustrated in Figure 1.8.

1.3.1 Regulatory Level

The regulatory level primarily concerns regulatory and data compliance requirements. In recent years, with increasingly stringent demands for user privacy data protection and personal information rights safeguarding, China has successively enacted laws and regulations such as the Data Security Law of the People’s Republic of China and the Personal Information Protection Law of the People’s Republic of China. As app data collection and

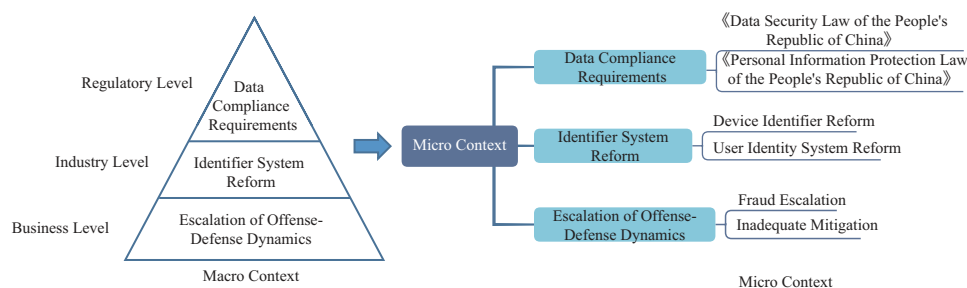


FIGURE 1.8 Multi-dimensional challenges in traffic fraud in the big data era.

compliance become more stringent, data compliance has become the foremost priority in big data governance. Anti-fraud identification in big data must be based on models that fully protect user privacy and utilize legally authorized data, imposing higher standards on the construction of traffic fraud systems.

1.3.2 Industry Level

The primary challenge at the industry level stems from the transformation of anti-fraud identification systems, specifically encompassing changes in both device identification systems and user identity identification systems.

1. Device Identification System Transformation

As a core element of traffic fraud prevention, the traditional device identification system dominated by mobile terminals and operating systems is no longer viable. It has been replaced by the decentralized open anonymous device identifier system developed by major domestic manufacturers. However, the new open anonymous device identifier system cannot verify or calibrate device fingerprints, nor can it authenticate their validity.

2. User Identity Identification System Transformation

The industry-standard user identity identifier is the International Mobile Subscriber Identity. However, access to the International Mobile Subscriber Identity has been restricted due to operating system updates, rendering identity verification and traffic fraud detection impossible in certain scenarios.

1.3.3 Business Layer

The primary challenge at the service level is the intensifying battle between attackers and defenders, which can be broken down into two aspects: fraud escalation and inadequate service controls.

1. Fraud Evolution

- Fraud techniques are becoming increasingly diverse. For instance, in the realm of high-risk device usage, the black/gray industry began with fake devices and fake user behavior, using emulators for traffic fraud. To bypass business risk controls, they evolved to genuine devices with fake user behavior, leveraging group control systems for fraud. Finally, they upgraded to genuine devices with genuine users but fake motivations, distributing part-time tasks to ordinary users via crowd-sourcing platforms to commit traffic fraud, significantly increasing the difficulty of business risk control.
- Fraud techniques have become increasingly specialized. The black/gray industry has evolved from initial “lone wolf” operations to organized, division-of-labor “gang operations,” forming specialized black/gray industry chains, as shown in Figure 1.9. The black/gray industry has also adopted cutting-edge AI technologies,

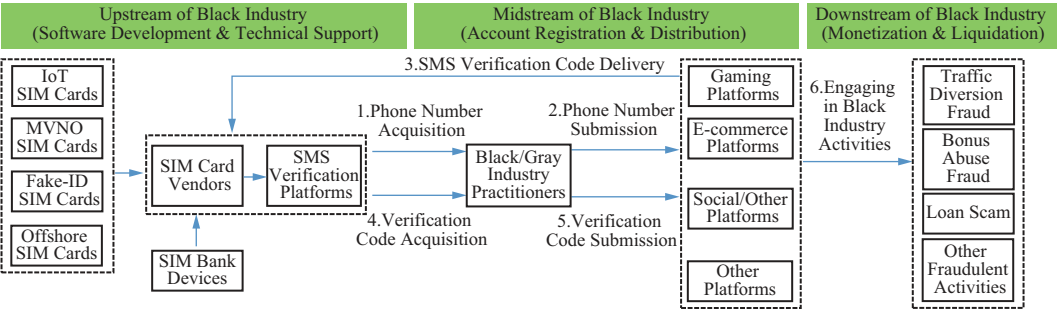


FIGURE 1.9 Specialized black/gray industry chain.

making fraud more efficient and harder to detect. Whenever unknown vulnerabilities exist in a business system, criminal groups quickly exploit them to commit fraud, causing significant losses in a very short time. By the time the business discovers the breach, the criminal team has already vanished.

- Fraud is shifting overseas. As China intensifies crackdowns on black/gray industry fraud through measures like “card-cutting” and “number-cutting,” these operations are gradually relocating abroad. Transnational fraud is growing rapidly, including cross-border gambling, money laundering, and romance scams.

2. Inadequate Business Prevention Measures

- Limited prevention methods. Businesses rely solely on simple risk lists or manual rules for isolated countermeasures, lacking a comprehensive anti-fraud system architecture that covers pre-event, in-event, and post-event processes.
- Lagging prevention. Due to the stealthy and rapidly evolving nature of black/gray industry fraud techniques, obtaining samples of unknown fraud methods often lags behind. Consequently, supervised models can only identify known fraud types, leaving blind spots in detecting unknown fraud patterns.
- Operating in isolation. Black/gray industry fraud syndicates exploit every profitable opportunity, leveraging limited resources to perpetrate continuous fraud across multiple platforms. However, business controls fail to establish effective cross-industry coordination, resulting in poor prevention outcomes.

1.4 TRAFFIC ANTI-FRAUD SYSTEM ARCHITECTURE

Addressing the challenges of traffic fraud in the big data era, this section will focus on two key aspects: the evolution of traffic fraud solutions and the architecture of anti-fraud systems in the current big data era.

1.4.1 Evolution of Traffic Anti-Fraud Solutions

The historical evolution of traffic anti-fraud solutions is illustrated in Figure 1.10. Key stages include: expert rules-based countermeasures during the PC internet era; machine learning and deep learning countermeasures in the mobile internet era; and complex network

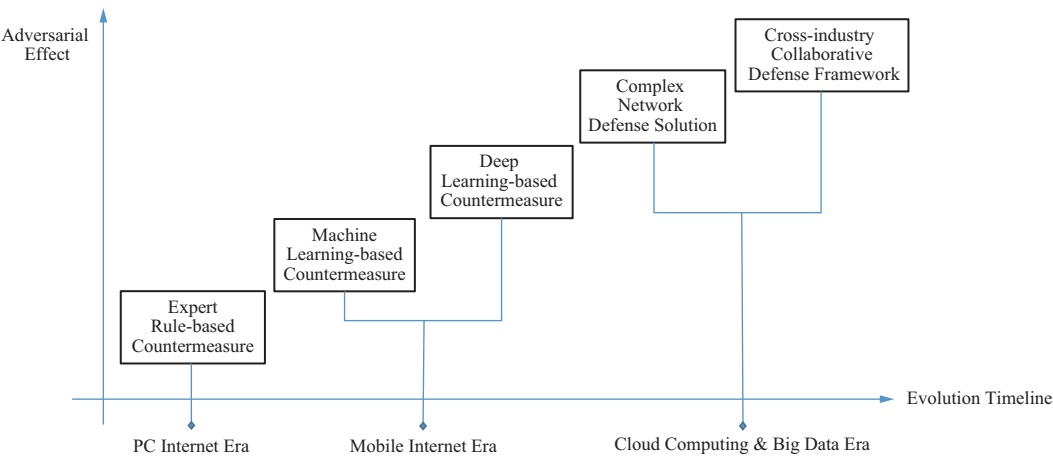


FIGURE 1.10 Historical evolution of traffic anti-fraud solutions.

countermeasures alongside cross-industry collaborative defense solutions in the cloud computing and big data era. As technology advances, the effectiveness of these countermeasures continues to improve.

1. PC Internet Era

During the PC internet era, fraud tactics employed by members of the black/gray industry were relatively simple. Expert rules-based countermeasures could achieve satisfactory results. This approach primarily relied on expert experience, analyzing fraudulent cases through data to manually extract basic rules, which were then combined with risk lists for implementation. Its advantages included simplicity, ease of use, and strong interpretability. However, it could only identify more obvious fraud methods, and the manual extraction process was costly.

2. Mobile Internet Era

In the early stages of the mobile internet era, explosive growth in internet traffic led to increasingly rich business feature information. Concurrently, machine learning algorithms matured and began to be widely applied in traffic fraud detection. The machine learning countermeasure construction process primarily involves integrating business feature information to learn black and gray industry fraud patterns from black and white sample training sets, then generalizing these learnings for online fraud detection. The advantage of this countermeasure is its ability to identify complex and highly concealed traffic fraud. The disadvantage is its weak explainability and lagging results in detecting unknown fraud types.

During the mid-stage of the mobile internet era, the massive generation of multimodal data—including images, speech, and video—rendered traditional feature extraction methods inefficient and ineffective. Deep learning demonstrated unique advantages in extracting features from such multimodal data, offering superior learning capabilities, higher extraction efficiency, and better results. Consequently, this approach gained widespread adoption in traffic fraud prevention.

3. Cloud Computing and Big Data Era

In the era of cloud computing and big data, enhanced computational and storage capabilities have enabled robust support for complex large-scale models, paving the way for the development of sophisticated network-based countermeasures. These approaches primarily leverage vast relational datasets, utilizing structural information between nodes and node attribute feature data to identify black and gray industry fraud activities from a holistic perspective. The advantage of this countermeasure is improved coverage capabilities and the ability to proactively detect unknown fraud types. However, it has disadvantages in terms of high computational demands and significant resource costs.

Concurrently, as business operations across industries increasingly migrate to the cloud and novel countermeasure approaches like federated learning emerge, cross-industry collaborative defense and control solutions can be implemented at low cost. These new countermeasures, grounded in industry-wide governance, enable effective dimensional escalation against the black and gray industry, substantially raising the cost of illicit operations.

Throughout the evolution of internet traffic, anti-fraud solutions have continuously advanced alongside AI technology. Each stage has seen the emergence of new traffic fraud countermeasures tailored to its characteristics, though this does not render previous approaches obsolete. Confronted with the ever-evolving, multifaceted tactics of the black and gray industry, anti-fraud efforts must integrate multiple countermeasures into a systematic framework to effectively address the complex challenges of traffic fraud in the big data era.

1.4.2 Architecture of Traffic Anti-Fraud Systems

Based on the characteristics of traffic fraud in the big data era and incorporating the latest advancements in artificial intelligence, the architecture of a big data-era traffic fraud system has been established, as shown in Figure 1.11.

The architecture of the traffic anti-fraud system in the big data era is primarily divided into four layers from bottom to top: the big data platform layer, the data governance layer, the traffic fraud detection model layer, and the online service layer. Meanwhile, the intelligence system and operational framework serve the entire traffic anti-fraud system architecture. The specific functions of each component within the traffic anti-fraud system architecture are detailed below.

1. Big Data Platform Layer

The big data platform layer serves as the foundational platform and framework, supporting the storage and computation of massive internet traffic data in the big data era. It primarily includes big data storage and computing infrastructure platforms (such as Hadoop and Spark), distributed data warehouses (such as Hive and Presto), and stream data processing frameworks (such as Flink and Storm).

2. Data Governance Layer

The core principles of the data governance layer are: first, ensuring that all utilized data has been properly authorized by users; second, encrypting and isolating

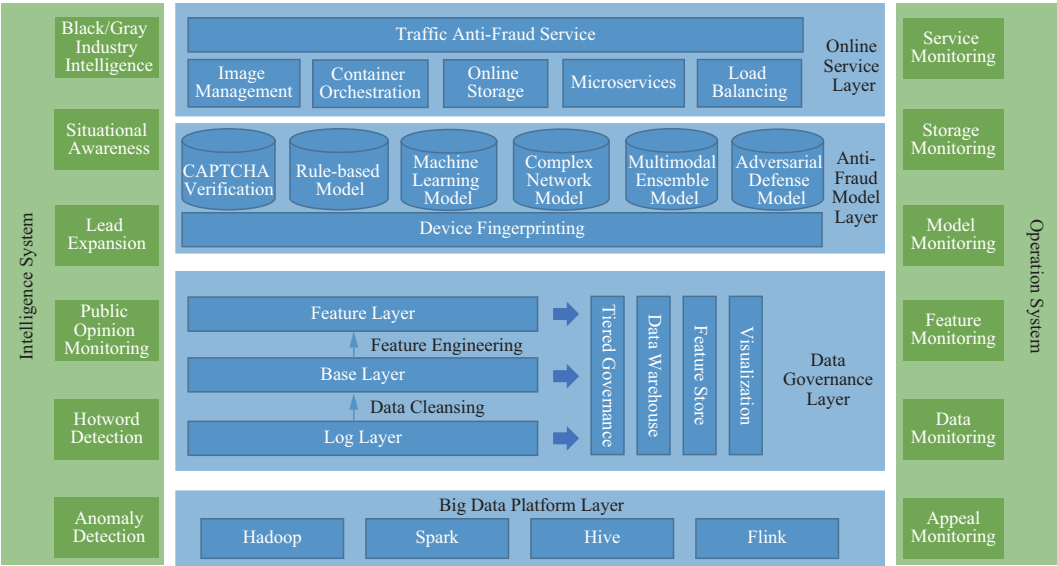


FIGURE 1.11 Architecture of traffic anti-fraud systems in the big data era.

privacy-sensitive and log data to guarantee data security; and finally, centrally performing data cleansing, processing, and management on raw traffic log data to enhance data quality. Since raw log data originates from diverse business scenarios, it often suffers from inconsistent field formats, non-standard naming conventions, and poor data quality. The data governance layer processes raw log data into standardized foundational layer data through methods like data cleansing. It then constructs high-quality profiling features required for traffic fraud modeling via feature engineering.

3. Anti-Fraud Model Layer

The anti-fraud model layer primarily utilizes profile feature data obtained through data governance. It employs artificial intelligence technologies to train various traffic fraud detection models, thereby identifying business traffic risks. As shown in Figure 1.12, the traffic fraud detection model spans the entire lifecycle of business platform traffic. Specifically, during the pre-traffic phase, CAPTCHA verification serves as the first security line to identify potential risks. Rule-based models then act as the second security line to further detect more obvious black/gray industry fraud activities. In the mid-traffic phase and post-traffic phase, machine learning models, complex network models, and multimodal ensemble models identify more covert and aggressively adversarial black/gray industry fraud activities. Finally, novel adversarial models address special cases within traffic fraud scenarios.

Additionally, device fingerprinting technology serves as the unique identifier for user identities in internet services and forms the foundational infrastructure for traffic fraud detection, spanning the entire lifecycle. Since all traffic fraud detection models are built

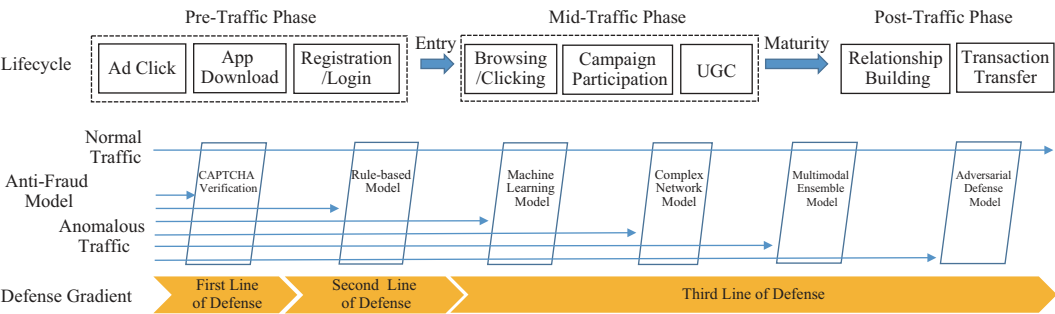


FIGURE 1.12 Traffic fraud detection model.

upon device fingerprinting technology, mastering this capability is an indispensable core competency.

1. Online Service Layer

The online service layer serves as the output layer for traffic fraud, directly supporting various business traffic scenarios via API interfaces. It primarily includes modules such as image management, container orchestration, online storage, microservices, and load balancing. This layer enables elastic scaling based on business traffic request volumes, supporting hundreds of billions of concurrent accesses while maintaining service stability and reliability.

2. Operations System

The operations system comprises core modules including service monitoring, storage monitoring, feature monitoring, data monitoring, and model monitoring. These modules oversee metric monitoring and operational management to ensure the stable and healthy operation of the anti-fraud system. Additionally, the system includes an appeal processing module, establishing user feedback and resolution channels for potential risk misjudgments within the anti-fraud system, thereby safeguarding user rights.

3. Intelligence System

The intelligence system serves as the “eyes” of the traffic anti-fraud system. On one hand, through big data analysis, it detects evolving tactics of the black/gray industry to evaluate the effectiveness of security countermeasures. On the other hand, it proactively captures the latest dynamics of the black/gray industry across the entire network. This provides risk control personnel with up-to-date intelligence on fraudulent methods, tools, and dark web transactions, enhancing their awareness of emerging trends in these illicit activities.

1.5 SUMMARY

This chapter primarily covers the background, challenges, and solutions related to traffic fraud. First, it outlines the three developmental stages of internet traffic and the distinct fraud issues associated with each stage. Next, it focuses on traffic fraud issues in the current

big data era, detailing fraud problems throughout the entire app lifecycle. Then, based on these challenges, it highlights the multi-layered obstacles faced by anti-fraud efforts at regulatory, industry, and operational levels. Finally, it presents the overall architecture of modern traffic anti-fraud systems to address these big data-era problems and challenges.

BIBLIOGRAPHY

- Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3), 235–255.
- Borodo, S. M., Shamsuddin, S. M., & Hasan, S. (2016). Big data platforms and techniques. *Indonesian Journal of Electrical Engineering and Computer Science*, 1(1), 191–200.
- Cai, X., Nie, F., Cai, W., & Huang, H. (2013). Heterogeneous image features integration via multi-modal semi-supervised learning model. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1737–1744). Darling Harbour in Sydney, Australia.
- Ladley, J. (2019). *Data governance: How to design, deploy, and sustain an effective data governance program*. Academic Press.
- Li, L., Fan, Y., Tse, M., & Lin, K. Y. (2020). A review of applications in federated learning. *Computers & Industrial Engineering*, 149, 106854.
- Mjolsnes, S. F., & Olimid, R. F. (2019). Private identification of subscribers in mobile networks: status and challenges. *IEEE Communications Magazine*, 57(9), 138–144.
- Molodetska, K., Timonin, Y., & Melnychuk, I. (2020). The conceptual model of information confrontation of virtual communities in social networking services. *International Journal of Electrical & Computer Engineering*, 10(1), 1043–1052.
- Qian, L., Luo, Z., Du, Y., & Guo, L. (2009, December). Cloud computing: An overview. In *IEEE International Conference on Cloud Computing* (pp. 626–631). Berlin, Heidelberg: Springer.
- Radhakrishnan, S. V., Uluagac, A. S., & Beyah, R. (2014). GTID: A technique for physical device and device type fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 12(5), 519–532.
- Sagiroglu, S., & Sinanc, D. (2013, May). Big data: A review. In *2013 International Conference on Collaboration Technologies and Systems (CTS)* (pp. 42–47). San Diego, CA, USA.
- Sullivan, E. (2022). Understanding from machine learning models. *The British Journal for the Philosophy of Science*, 73(1), 109–133.
- Tang, D., Chen, G., & Liu, Q. (2022, December). The current situation of internet underground industry chain and its governance model. In *2022 5th International Conference on Humanities Education and Social Sciences (ICHESS 2022)* (pp. 1008–1017). Chongqing, China. Atlantis Press.

Traffic Fraud Tactics and Their Impact

With the rapid development of mobile internet, common user needs can be fulfilled through various internet platforms—such as shopping on e-commerce sites, watching videos on content platforms, chatting on social media, and gaming on entertainment apps. As integral components of platform traffic, users engage in activities like registering accounts, logging in, participating in events, and making payments during their platform usage. To attract massive user traffic, platforms often aggressively promote across channels and design lucrative incentive programs to draw users—simultaneously creating opportunities for the black/gray industry to profit.

User traffic behavior paths during platform interactions can be categorized into the pre-traffic phase, mid-traffic phase, and post-traffic phase. In the pre-traffic phase, the core scenario involves reaching users through ad placements. After downloading the app, users register an account and log in. During the mid-traffic phase, users begin generating behaviors such as browsing, clicking, searching, sharing, and participating in platform marketing activities. Concurrently, they produce various forms of user-generated content (UGC), including text, images, audio, and video. In the post-traffic phase, after deep engagement and social interactions, users generate related social behaviors and transactional activities. A typical behavioral path for app user traffic is illustrated in Figure 2.1.

At each stage of the traffic cycle, leveraging different platform interaction methods, the black/gray industry employs distinct fraudulent tactics using corresponding tools. Typical fraud methods across traffic cycle phases are illustrated in Figure 2.2.

This chapter introduces several typical traffic fraud tactics across different phases of the traffic lifecycle, including promotion-phase fraud, registration fraud, login fraud, “wool-pulling” fraud, brushing volume fraud, traffic diversion fraud, telecom fraud, and resource monetization fraud. It also analyzes the difficulty and profitability of fraud across these phases.

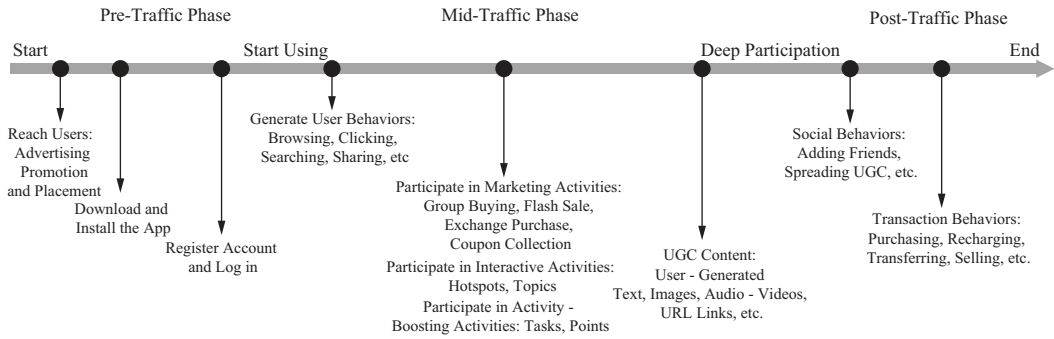


FIGURE 2.1 Typical user behavior path in app traffic.

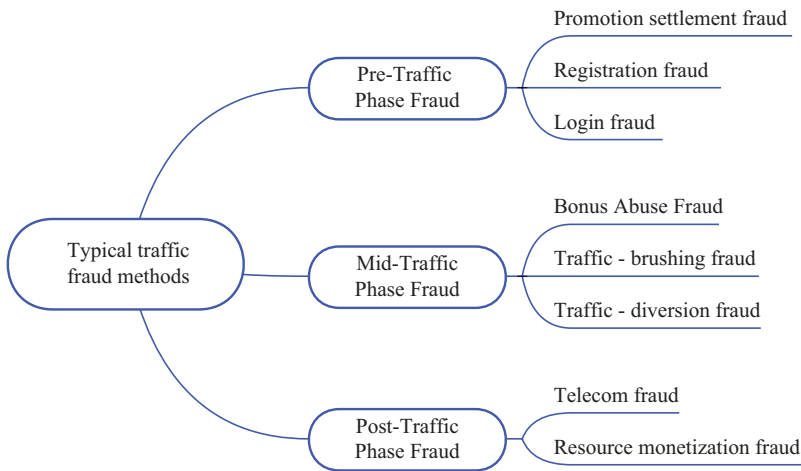


FIGURE 2.2 Typical traffic fraud tactics across different traffic cycle stages.

2.1 PRE-TRAFFIC PHASE TRAFFIC FRAUD

This section details typical fraud behaviors occurring during the pre-traffic phase of traffic campaigns, such as promotion settlement fraud, registration fraud, and login fraud, which emerge when black/gray industry operators reach users through ad placements and during user account registration and login processes.

2.1.1 Promotion Settlement Fraud

During the pre-traffic phase, traffic fraud primarily occurs between advertisers and advertising service traffic platforms. Promotion settlement fraud causes multiple harms, including increased customer acquisition costs for advertisers, disruption of traffic platform ecosystems, and black/gray industry actors exploiting fake traffic to defraud users. Below, we first outline the basic process of advertising promotion.

The fundamental advertising deployment process, illustrated in Figure 2.3, involves advertisers generating ad placement demands. Through traffic providers' channels, ads

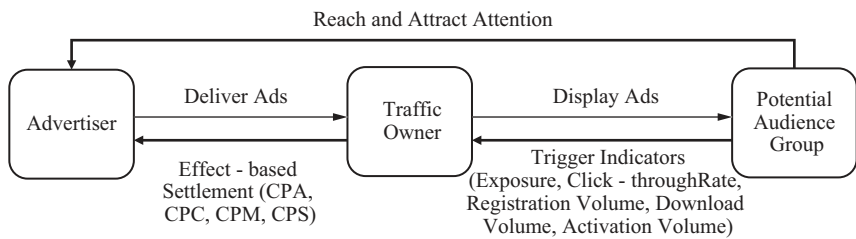


FIGURE 2.3 Basic process of advertising promotion.

are displayed to reach potential audiences. Traffic providers monetize their traffic, while advertisers benefit from engaging these potential audiences.

First, advertisers select ad formats such as feed ads, search engine marketing, demand-side platforms, or ad networks. Next, they choose the advertising payment method according to different cost settlement models—including cost per action (CPA), cost per click (CPC), cost per thousand impressions (CPM), or cost per sale (CPS). Finally, they track advertising metrics (such as impression rate, click-through rate, registrations, downloads, and activations) to evaluate campaign effectiveness.

The following sections outline the primary aspects of promotion settlement fraud from three distinct dimensions.

2.1.1.1 Dimension: Automated Fraud Methods

The characteristics of machine fraud versus manual fraud are compared in Table 2.1. Machine fraud primarily relies on automated scripts for bulk execution, exhibiting pronounced short-term clustering patterns that are easily detectable, with low fraud costs. Manual fraud, however, centers on crowdsourcing real individuals, resulting in weaker clustering patterns. Such fraud is relatively covert, easily evades detection, and incurs higher costs.

2.1.1.2 Traffic Authenticity Dimensions

Traffic fraud primarily falls into three categories: fake traffic, attribution fraud, and illegal traffic. Fake traffic, also known as invalid traffic (IVT), can be further categorized into general IVT and sophisticated IVT. Fake traffic directly manipulates metrics to commit traffic fraud. Typical manifestations of fake traffic across different channels are illustrated in Figure 2.4.

TABLE 2.1 Comparison of Characteristics between Automated and Manual Fraud

Characteristics	Machine Fraud	Manual Fraud
Automation level	High	Low
Cost investment	Low	High
Aggregation characteristics	Highly clustered, easily detectable	Low clustering, relatively concealed
Difficulty of countermeasures	Low	High
Organizational structure	Controlling botnet devices	Professional “brushers”
Typical methods	Data hijacking & tampering, crawler simulation	Task cashback, online earning incentives, human crowdsourcing



FIGURE 2.4 Typical forms of fake traffic across different channels.

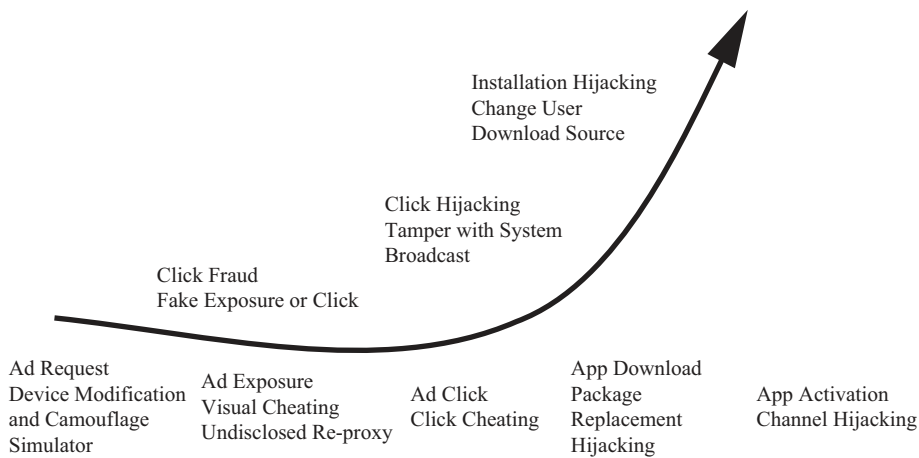


FIGURE 2.5 Attribution fraud in advertising campaigns.

General IVT refers to traffic identifiable through lists and predefined parameter features using conventional filtering methods. Primary forms include IDC traffic, simple bot traffic, and non-human behavior traffic. Sophisticated IVT requires deep analysis for detection, such as disguised user traffic, hijacked traffic, fake website traffic, spoofed location traffic, and invalid proxy traffic.

Attribution fraud constitutes a more sophisticated form of deception, primarily manifesting as click fraud, click hijacking, ad stacking, and installation hijacking. Attribution fraud within advertising promotions is illustrated in Figure 2.5. This fraud primarily exploits the extended lifecycle of ad campaigns and the last-click attribution principle. Fraudsters merge traffic from intermediate channels into their own channels to fabricate data, manipulate campaign metrics, and thereby defraud advertisers of their campaign budgets.

Illegal traffic refers to traffic fraud where traffic providers privately employ gray-area tactics to obtain genuine traffic. Primary forms include incentivized tasks, online money-making scams, clickbait downloads, and Trojan backdoor operations.

2.1.1.3 Advertising Settlement Metrics Dimensions

During ad campaigns, members of the black/gray industry combine multiple payment models to commit traffic fraud. Common methods of promotion settlement fraud, categorized by settlement metrics and profit models, are illustrated in Figure 2.6.

2.1.1.3.1 CPC/CPM-Based Settlement Fraud Primary fraud methods exploiting CPM and CPC as settlement metrics include fake impressions and fake clicks.

- Fake traffic: By fabricating traffic requests, fraudsters deceive ad performance monitoring codes to falsify genuine ad campaign results. Examples include deploying automated crawlers on servers to generate fake requests or embedding cheating code in web client JavaScript to simulate traffic through hidden pop-up ads, ad stacking, or multiple request submissions.
- Fake Clicks: Primarily achieved by combining automated scripts with group control devices to simulate bulk clicks by real users, luring clicks through sensational content, or incentivizing crowdsourcing of human participants to generate massive volumes of invalid clicks. This prevents advertisers from achieving their expected campaign outcomes.

2.1.1.3.2 CPA-Based Fraud The primary fraud methods targeting CPA as a settlement metric include fake downloads, fake installations, and fake activations. Common CPA-based cheating tactics are summarized in the following three points.

- Modifying device information and forging user identities through emulators and device farms, then simulating user behavior to perform software downloads, installations, and activations.

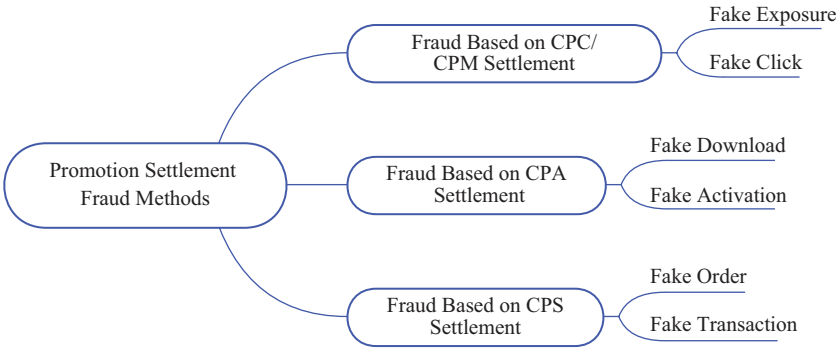


FIGURE 2.6 Common promotion settlement fraud methods.

- By bypassing third-party detection SDKs, fraudsters can perform fake downloads, installations, and activations without actual installations, thereby depleting advertiser budgets.
- Employing silent downloads, bundled downloads, and package hijacking to automatically download, install, and activate certain software without user awareness.

2.1.1.3.3 CPS-Based Settlement Fraud The primary fraud methods targeting CPS as a settlement metric include cookie planting and HTTP hijacking of user browsing behavior. CPS settlement is stringent and relatively difficult to manipulate. The typical fraud method for CPS-based settlements relies on cookie planting—installing specific programs on websites to implant cookies from those sites onto users’ computers. Since ad networks use cookies to track ad performance, these cookies are stored in users’ web browsers. When users visit specific sites and place orders, the implanters can then claim affiliate commissions.

The black/gray industry continuously probes traffic platforms to uncover business security vulnerabilities, exploiting these flaws to execute various malicious fraud schemes for profit. When users encounter an internet product through promotional channels and begin using it, the black/gray industry and its activities inevitably follow. Using a platform’s product involves stages such as registration, login, participation in marketing interactions, and payment settlement. Below, we primarily describe fraudulent behaviors in two typical traffic fraud scenarios.

2.1.2 Registration Fraud

Before committing malicious fraud, members of the black/gray industry must first acquire platform accounts—the “entry ticket” for accessing the platform and reaching users. Registration fraud involves using false or illegal identity information to bypass platform business risk controls through a combination of manual and automated tools, mass-registering accounts without intent for legitimate use. This includes various methods such as secondary number registration, registration machine registration, automated script batch registration, and “manual field promotion” registration. Accounts registered through these non-standard channels enable black/gray industry groups to conduct fraudulent activities such as scams, gambling, pornography, “wool-pulling” (exploiting promotional offers), and fake traffic marketing. The links in the registration fraud industrial chain are illustrated in Figure 2.7.

The registration fraud industrial chain can be divided into the following three parts.

- **Upstream of the Industry Chain:** The upstream primarily provides information, materials, automated tools, and technical support for malicious registration in black/gray industries. This segment includes SIM card dealers, SMS verification platforms, and CAPTCHA solving platforms. Specific links involve device farm support, proxy IP support, device modification tool development, and personal information trafficking.

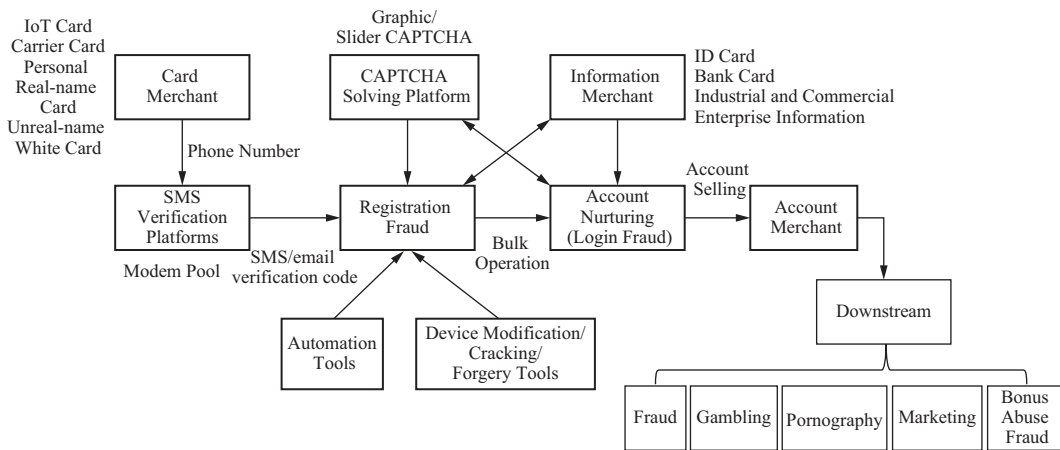


FIGURE 2.7 Components of the registration fraud industry chain.

- **Midstream of the Industry Chain:** Black/gray industry operators leverage bulk user personal information and mobile phone numbers supplied by upstream sources to complete platform account registration and nurturing. This involves using SMS verification platforms to obtain CAPTCHA codes, employing CAPTCHA solving platforms to identify image CAPTCHA codes, and utilizing email accounts for authentication and activation.
- **Downstream of the Industry Chain:** The downstream primarily consists of multi-tiered account merchants holding large quantities of platform accounts generated through malicious registration. They are responsible for supplying these malicious accounts to other members of the black/gray industry for fraudulent activities.

Although various traffic platforms rely on active accounts to drive traffic, the primary purpose of maliciously registered accounts is to support activities in the black/gray industry. These accounts do not generate positive revenue for the platform but instead disrupt platform order and interfere with normal operations. Accounts used for registration fraud are primarily newly registered accounts. Limited account-level information and the rapid iteration of methods within the registration fraud supply chain are key challenges encountered when combating registration fraud.

2.1.3 Login Fraud

Beyond controlling accounts through registration fraud, illegally stealing legitimate user accounts via login fraud represents another avenue for the black/gray industry to acquire account resources. Login fraud primarily encompasses account theft fraud and account farming fraud. After successfully stealing accounts, the black/gray industry may choose to commit large-scale malicious activities within a short timeframe, maximizing the account's value until the account owner reclaims it or the platform detects and takes action. Alternatively, they may farm and hoard accounts, storing them as resources for subsequent

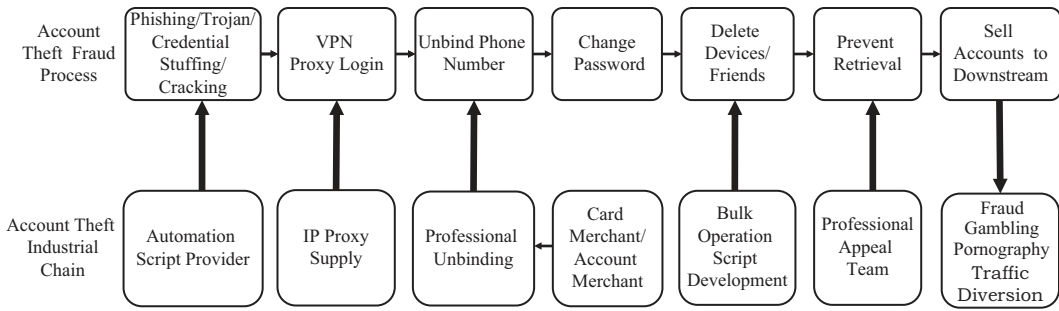


FIGURE 2.8 Relationship between account theft fraud process and the account theft fraud industry chain.

fraudulent activities. The account farming fraud supply chain represents the next stage following registration fraud. The relationship between account theft fraud processes and the account theft fraud supply chain is illustrated in Figure 2.8.

Account theft fraud can be categorized into four types:

- Black/gray industry account thieves directly obtain usernames and passwords through social engineering databases, trojans, viruses, and phishing websites.
- The black/gray industry repeatedly attempts to crack passwords using methods such as credential stuffing and automated cracking tools.
- The black/gray industry impersonates legitimate users to file account recovery requests, exploiting platform authorization to modify account passwords.
- The black/gray industry conducts database dumping to leak sensitive information such as platform users' accounts, passwords, and personal privacy. They then steal user information for data laundering operations while simultaneously monetizing fraud. Finally, they perform credential stuffing attacks on other platforms to commit account takeover fraud against the same users.

Additionally, to prevent accounts obtained through registration fraud or account theft from being deactivated by platforms due to prolonged inactivity, or to conceal traces of abnormal logins, members of the black/gray industry also engage in account laundering and nurturing. The core method of account nurturing involves mimicking the behaviors of legitimate users to bypass security checks. This transforms low-quality secondary accounts into high-quality platform accounts with clean historical activity and strong natural user attributes, which are then utilized by the black/gray industry for subsequent fraud schemes. The basic process of account nurturing fraud on short-video platforms is illustrated in Figure 2.9.

Platforms often organize various activities to boost traffic, attract user attention, enhance user engagement, and stimulate consumption. For instance, e-commerce platforms frequently host promotions, flash sales, and spend-and-save campaigns; video platforms offer

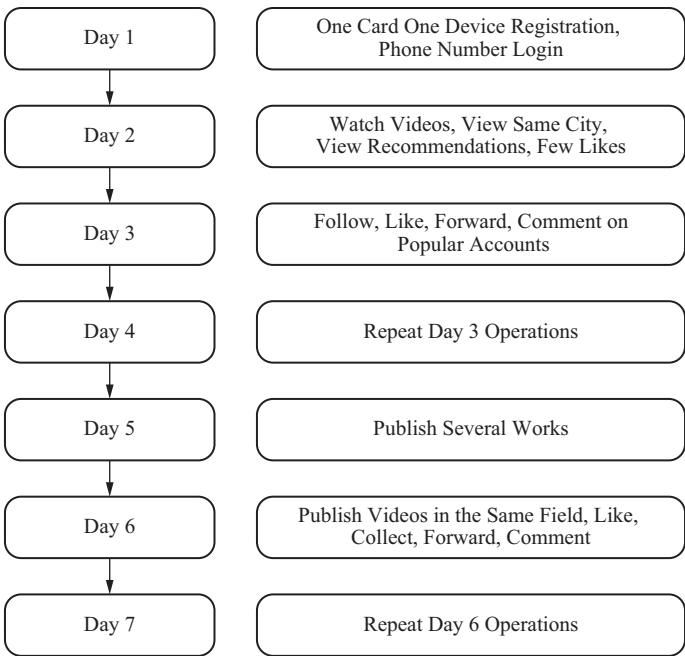


FIGURE 2.9 Basic process of account farming fraud on short-video platforms.

discounted membership packages; and gaming platforms provide value-added services like discounted points or in-game items. Merchants actively participate in these marketing activities to drive traffic, secure platform recommendations, and attract users. The black/gray industry exploits loopholes in traffic platform rules to conduct traffic fraud on these platforms.

2.2 MID-TRAFFIC PHASE FRAUD

This section details typical fraud tactics employed by the black/gray industry during the mid-traffic phase of traffic activities, including “scraping” fraud and brushing volume fraud. Additionally, this section covers UGC created during platform usage, such as false, fraudulent, or abusive content posted by malicious users. It focuses on fraudulent UGC, which predominantly serves to funnel traffic for post-traffic phase fraud. Some false UGC also directly constitutes fraudulent activity.

2.2.1 “Wool-Pulling” Fraud

“Wool-pulling” refers to users exploiting marketing campaigns on traffic platforms to obtain discounts through various means, enabling them to purchase goods or value-added services at extremely low prices. While platform promotions aim to attract new users, reactivate old ones, and reward active users, members of the black/gray industry use large numbers of platform accounts and automated software to specifically target logical or system vulnerabilities in platform operations. They circumvent platform restrictions to bulk-purchase goods or coupons and then resell them for profit. The emergence of

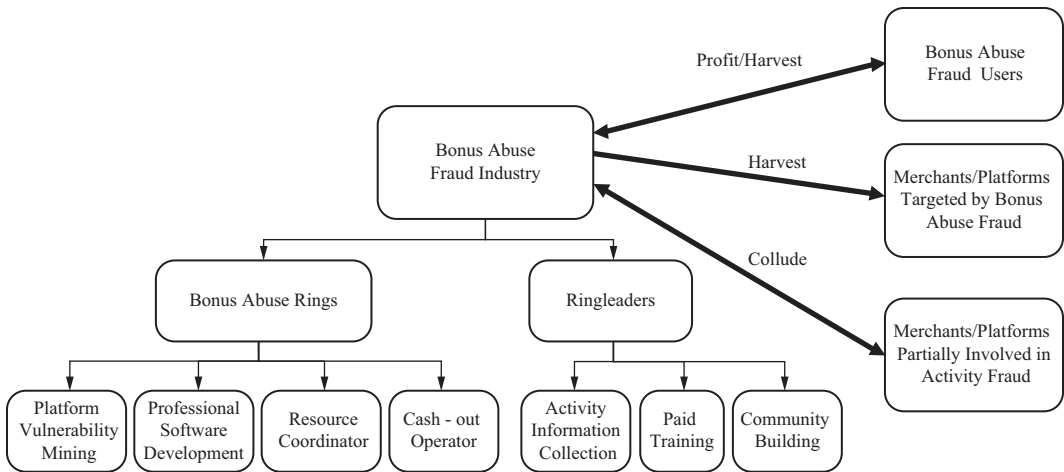


FIGURE 2.10 “Wool-shearing” fraud process.

large-scale malicious traffic from “wool-shearing parties” disrupts the ecosystem of traffic platforms. This impacts both normal traffic and the reputation of the platforms while also causing economic losses to the platforms or their merchants.

The “wool-shearing” fraud process is illustrated in Figure 2.10. Currently, “wool-shearing parties” exhibit characteristics of specialization, team organization, professionalization, and transnational operations. “Wool-shearing” fraud primarily falls into two categories.

The first type involves manual “wool-shearing” for promotional fraud. Led by organizers (“sheep heads”), these groups use tools to detect available platform discounts, notify “wool-shearing parties” within communities, and execute promotional fraud. Some intelligence groups even require user fees for participation, allowing organizers to profit twice. On the other hand, the “sheep-head” uses intelligence to manipulate “wool-shearing gangs” as controllable traffic sources. They collude with traffic providers to commit promotion settlement fraud against merchants or conspire with merchants to inflate data metrics, thereby earning commissions from both the platform and the merchants.

The second type involves professional “freeloaders” using specialized “freeloading” software and tools from the black/gray industry to exploit platform vulnerabilities on a large scale, manipulating market prices or reselling for profit. This type of activity is more covert and carries greater risk but also yields higher returns. Wherever profitable marketing campaigns exist, “wool-pulling parties” will appear—snatching ride-hailing coupons, one-yuan flash sales, limited-time liquor deals, exclusive sneakers, premium smartphones, gold jewelry, and more. These parties buy at rock-bottom prices and resell at inflated rates to pocket the difference. Professional “wool-pulling” gangs feature distinct roles: first is the intelligence gathering role, responsible for identifying platform activity vulnerabilities and assessing risk control strategies. Next is the fraud resource preparation role, liaising with upstream suppliers to source accounts, mobile phone numbers, bank cards, and SMS verification platforms. Then comes the development role, creating customized automated scripts. Finally, the fraud operation role liquidates acquired resources for profit.

2.2.2 Brushing Volume Fraud

Brushing volume fraud typically involves fabricating positive data and metrics to mislead platforms or users for fraudulent gain. Specific methods include, but are not limited to inflating sales, clicks, or views for specific products/services through fake transactions; fabricating positive or negative reviews; generating fake orders, sales figures, and reviews on e-commerce platforms; inflating fake clicks, views, or play counts on content platforms; or inflating active users on gaming platforms. Taking content platforms as an example, the brushing volume fraud chain is illustrated in Figure 2.11.

Brushing volume fraud primarily involves two types: machine-generated traffic and human crowdsourcing. Common methods include protocol proxies, automated scripts, group control systems, crowdsourcing platforms, and online money-making task groups.

Order padding is another form of traffic manipulation, primarily involving sellers paying buyers (or “padders”) to artificially boost product visibility, order volume, and positive reviews through non-transactional means. Common order padding fraud chains and organizational structures are illustrated in Figure 2.12. Order padding not only introduces traffic fraud risks but may also spawn additional fraud risks like brushing volume fraud scams.

2.2.3 Traffic Diversion Fraud

Traffic diversion fraud serves as a critical link between the initial and later stages of traffic fraud operations within the black/gray industry. It primarily leverages UGC entry points on traffic platforms to increase exposure for illicit activities. As user-created content, UGC constitutes a vital component of traffic ecosystems, typically appearing in formats such as text, voice, images, videos, and third-party links. Common UGC forms include messages/comments, posting/sharing, following/liking, messaging, and link sharing.

The diversity of UGC formats not only caters to users’ varied needs but also provides the black/gray industry with convenient exposure channels and methods. Common malicious UGC includes content that violates national laws and regulations, pornography, abusive

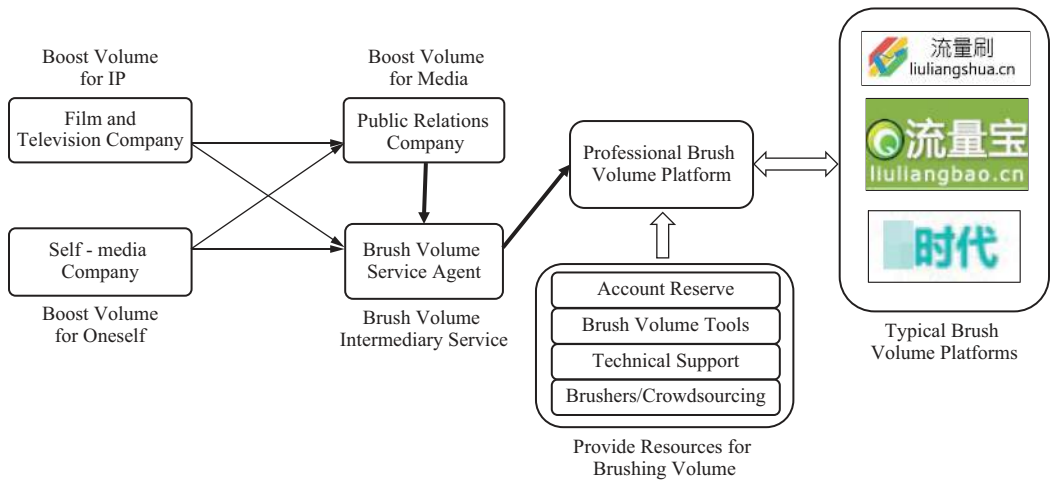


FIGURE 2.11 Brushing volume fraud chain on content platforms.

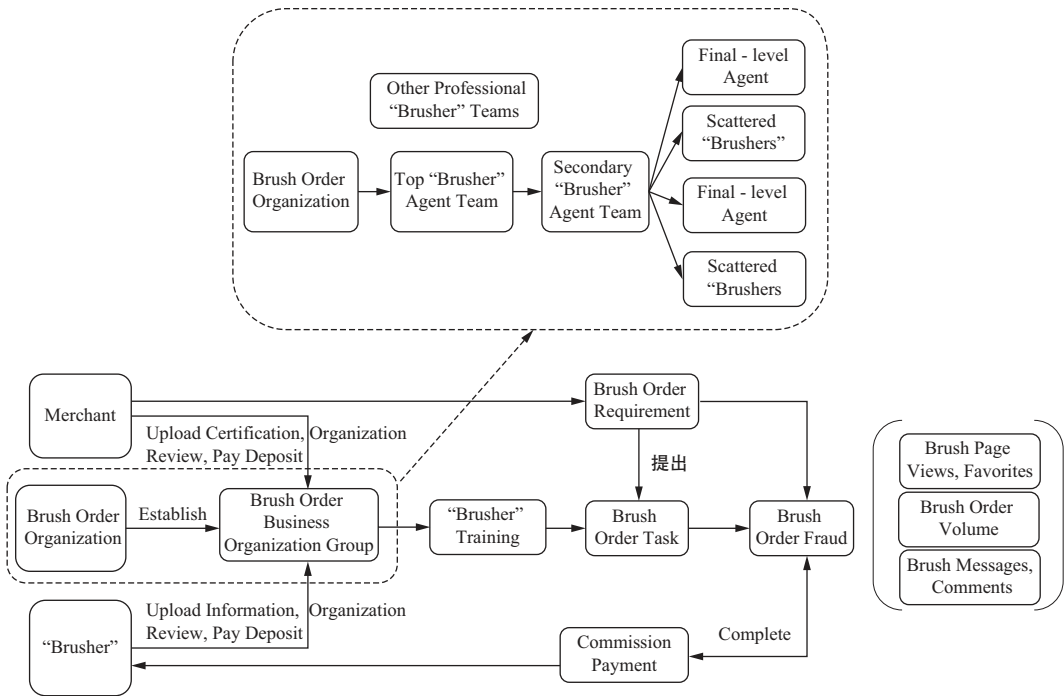


FIGURE 2.12 Common fake order fraud chains and organizational structures.

language, and other materials that adversely affect users; low-quality advertising and marketing content; content that influences public opinion; and URLs or APK content that redirect users to third-party platforms for fraud. Typical UGC data, platforms, and formats are illustrated in Figure 2.13.

From a traffic perspective, UGC serves as a core driver for traffic diversion. Legitimate UGC-driven traffic diversion primarily supports advertising sales, personal networking, group communication, and business collaborations. Malicious UGC, however, includes fraudulent content produced and disseminated by the black/gray industry through various channels to achieve malicious traffic diversion—such as directing users to fraudulent schemes, gambling platforms, and adult livestreaming services. Common methods of UGC-based traffic diversion fraud, as illustrated in Figure 2.14, include replacing account profile pictures with QR codes to redirect users to external adult streaming platforms; using message text to divert traffic to fraudulent part-time job platforms; sending private messages to direct users to adult streaming platforms; and embedding URLs within video content to funnel traffic to gambling platforms.

As illustrated by the above cases, scenarios prone to traffic diversion fraud on traffic platforms include embedding diversion content within published posts, customizing account avatars and nicknames on user profiles, modifying location settings, and leveraging private messages and comment sections. The emergence of diversion content primarily relies on user-accessible features. Table 2.2 demonstrates the correspondence between user-accessible features and diversion scenarios in traffic diversion fraud.

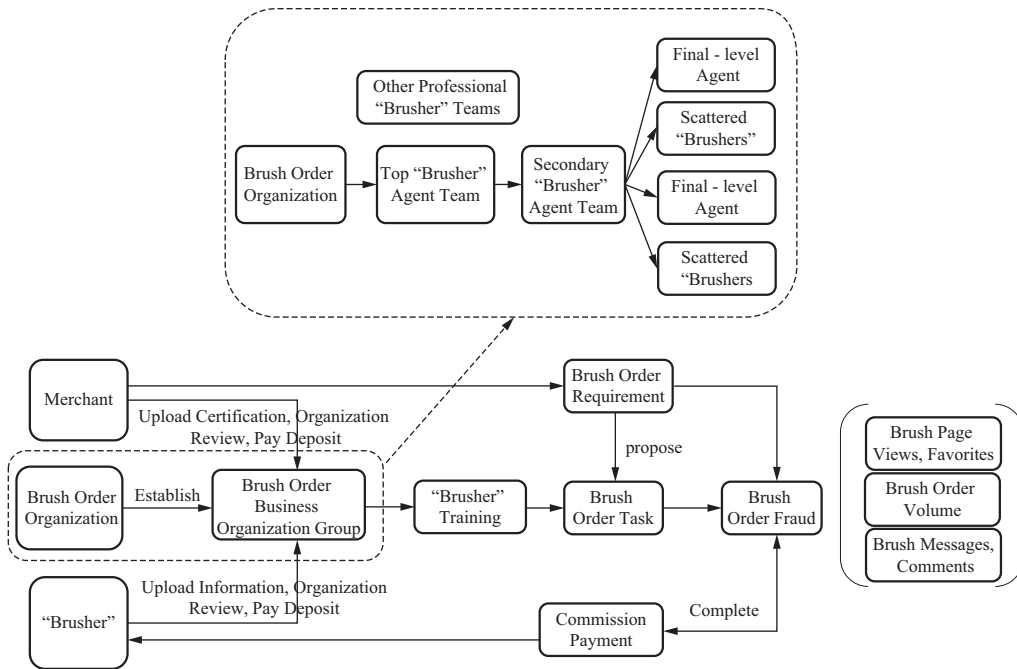


FIGURE 2.13 Typical UGC data, platforms, and formats.



FIGURE 2.14 Common methods of traffic diversion fraud using UGC.

TABLE 2.2 Correspondence between User-Accessible Features and Traffic Redirection Scenarios in Traffic Diversion Fraud

User-Accessible Features on Traffic Platforms	Traffic Platform Scenarios for Traffic Redirection
User-posted content	Embedding traffic-redirecting content (text, images, videos, audio, etc.) within published content
Modifying user personal information	Modifying personal information (nickname, profile picture, location, etc.) to include traffic-driving content
Users sending private messages	Private message content (text, images, etc.) containing promotional material
Users posting messages	Comment content (text, images, etc.) containing promotional material
Users send video bullet comments	Video bullet comments containing traffic-driving content
.....	...

The black/gray industry typically tailors its traffic diversion fraud pathways to exploit specific functionalities of traffic platforms. They often use tools to mass-generate traffic-redirecting content. To circumvent platform anti-fraud measures, the black/gray industry employs industry jargon, content distortion tactics, and covert embedding techniques to reach legitimate users and execute traffic diversion fraud. Such fraud leveraging UGC commonly occurs in user content publishing scenarios and user customization settings. The primary manifestations of traffic diversion fraud are illustrated in Figure 2.15.

The core process involves attracting users with attention-grabbing content to convey the traffic diversion fraud message. For instance, on a short-video platform, videos featuring attractive influencers retain user engagement. When users click into the diverting account’s profile page, the black/gray industry leaves contact information there. After users reach out, the black/gray industry executes subsequent order-padding fraud via social messaging apps. The entire traffic diversion fraud pathway can be summarized as follows: first, the platform recommends video content; then, users are lured to view the custom bio on the personal profile page; finally, fraud is carried out via social media platforms.

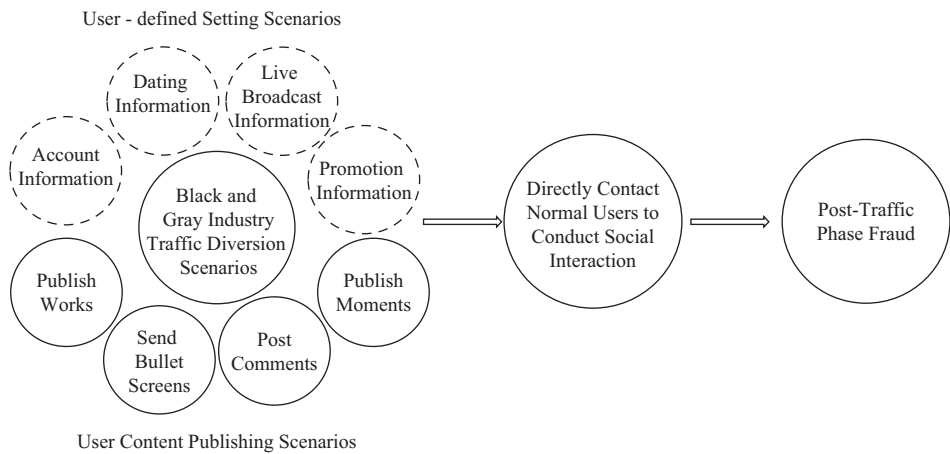


FIGURE 2.15 Primary manifestations of traffic diversion fraud.

TABLE 2.3 Common Black/Gray Industry Traffic Diversion Fraud Pathways

Traffic Diversion Fraud Scenario	Path Origin	Traffic Redirection Path
Esports gambling Website traffic redirection	Keyword search	Search for esports content, users watch videos, embedded URL text within esports videos, users visit gambling websites
Part-time job scams Traffic redirection	Content recommendations	Recommend part-time job articles for earning money. Users browse the content and add contact information in the article comment section.
Adult streaming App traffic redirection	Live streaming feature	Live stream bullet comments with URL links; users visit the URL, download the adult live streaming app, and install it
Offline adult events Traffic generation	People nearby	Attractive female profile pictures; users visit personal profiles and add contact information
Real-person crowdsourcing Traffic boosting	Send private messages	Send private messages inquiring about part-time work; users read messages, add contact information, and join traffic boosting organizations
.....		

For example, on an e-commerce shopping platform, a clothing product’s main image features a provocative model, prompting users to click into the product details page. Fraudulent traffic diversion fraud messages then appear in the product comment section. Users who click the poster’s profile obtain contact information, add the poster on social media, and complete the traffic diversion fraud. The traffic diversion fraud path can be summarized as follows: first, users are lured through keyword searches. Then, they are enticed to click, view comment sections, and visit personal profiles. Finally, fraud is carried out via social media platforms.

Beyond the two case studies above, actual traffic anti-fraud processes encounter numerous traffic diversion fraud pathways tailored to specific products. Common black/gray industry traffic diversion fraud pathways are shown in Table 2.3.

As UGC formats and the functional characteristics of various traffic platforms continue to evolve, black/gray industries are constantly discovering new traffic diversion fraud pathways. After exploiting UGC to defraud legitimate users on traffic platforms, these operations enter the post-traffic phase. At this point, black/gray industries begin directly contacting legitimate users through social media applications.

2.3 POST-TRAFFIC PHASE TRAFFIC FRAUD

This section outlines typical traffic fraud tactics employed during the post-traffic phase of traffic-based campaigns. After fraudulent UGC in the mid-traffic phase attracts legitimate users, black/gray industry cybercriminals initiate interactions through social behaviors, ultimately committing crimes via telecom fraud and other deceptive tactics. Additionally, following “scraping” fraud in the mid-traffic phase, black/gray industry cybercriminals in the post-traffic phase exploit social channels to sell acquired resources through fraudulent means, generating illicit profits while causing losses to platforms.

2.3.1 Telecom Fraud

To maximize profits in the post-traffic phase, the black/gray industry often resorts to more direct methods to defraud users of their money. They engage in direct social interactions with legitimate users through various channels—such as impersonating identities likely to intersect with users on the platform or setting traps to lure users into making payments. The black/gray industry uses social apps, phone calls, text messages, emails, and instant messaging software to induce payments, a practice known as telecom fraud. Telecom fraud primarily falls into categories such as e-commerce shopping scams, impersonation scams, financial investment scams, and dating scams.

Typical forms include “pig-butcher schemes,” “fish-butcher schemes,” and “bird-butcher schemes” in telecom fraud. “Pig-butcher schemes” involve the black/gray industry developing emotional bonds through chats to gain trust, then luring victims into gambling, lottery, or investment platforms for deposits to steal funds. “Fish-butcher schemes” involve the black/gray industry enticing users with promises of credit card limit increases, forced loan account activations, or cashback offers. They then trick victims into clicking fraudulent links to siphon funds. “Bird-butcher schemes” use part-time job or money-making scams to attract participants, manipulating victims with persuasive tactics to continuously invest until they lose everything. The primary steps of telecom fraud by the black/gray industry can be summarized as follows: establishing social connections with users, building trust through various manipulative tactics, luring users into participating and investing funds, and finally transferring the fraudulently obtained money to complete the scam. The main steps of telecom fraud are illustrated in Figure 2.16.

In telecom fraud, social behavior serves as a precursor to payment fraud, primarily aimed at gaining victims’ trust. Payment fraud, however, represents the ultimate objective of the black/gray industry, resulting in financial losses for victims. The relationship between social behavior and payment fraud in telecom fraud is illustrated in Figure 2.17.

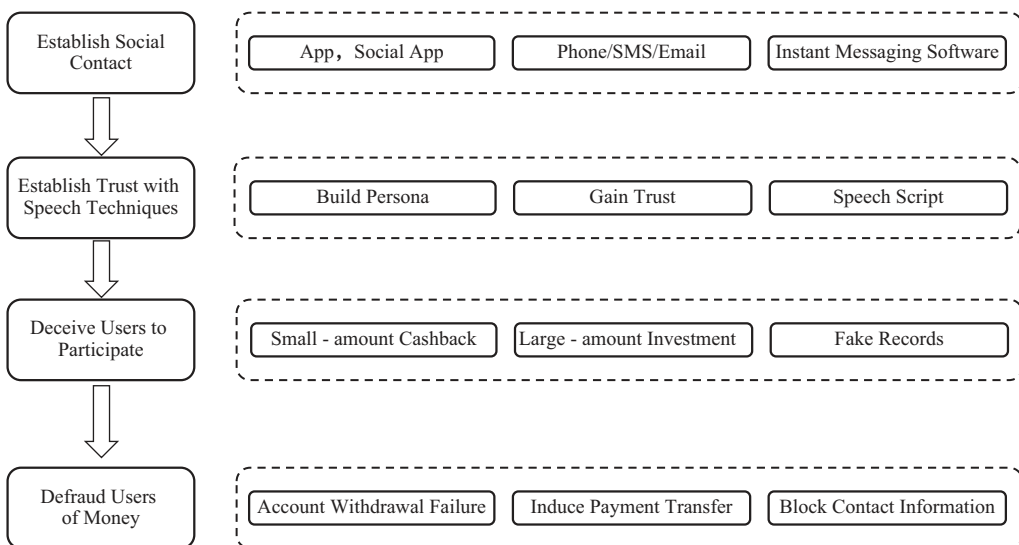


FIGURE 2.16 Key steps in telecom fraud.

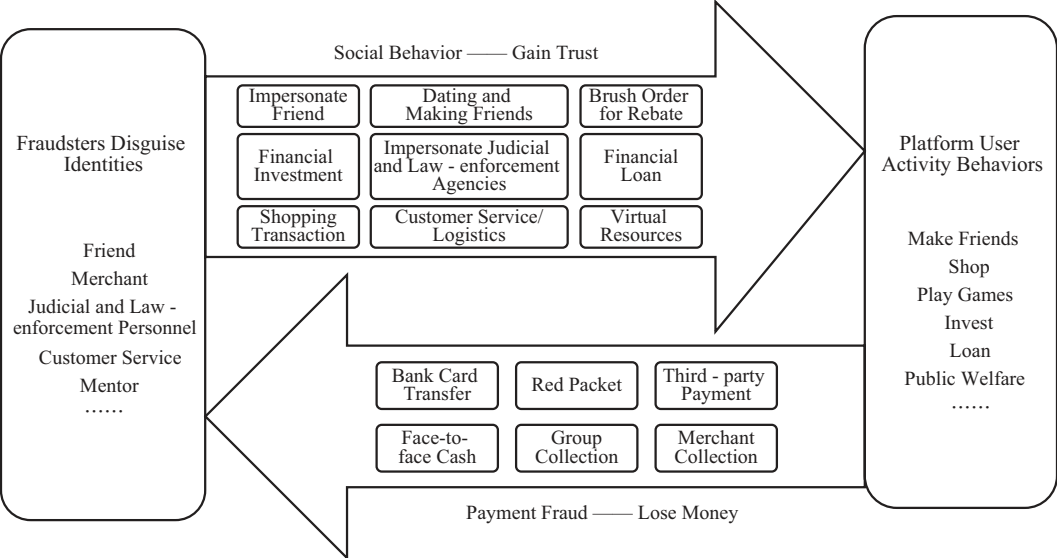


FIGURE 2.17 Relationship between social engineering and payment fraud in telecom fraud.

2.3.2 Resource Monetization Fraud

In the post-traffic phase, besides directly targeting users with telecom fraud, another typical profit-making method for the black/gray industry is to obtain platform resources through “coupon arbitrage” fraud and brushing volume fraud, then further commit fraud or monetize these resources.

Resource monetization fraud falls into two categories: one causes significant harm to platforms—such as when members of the black/gray industry use fraudulent tools to “scrape” platform coupons and resell them to users without generating new active users for the platform; the other type primarily harms users, such as when the black/gray industry obtains large quantities of concert tickets through improper means and resells them at inflated prices. Another example involves selling counterfeit cigarettes and alcohol to users via social channels or collecting payments without delivering goods, directly defrauding users of their assets. Common types of resource monetization fraud are illustrated in Figure 2.18.

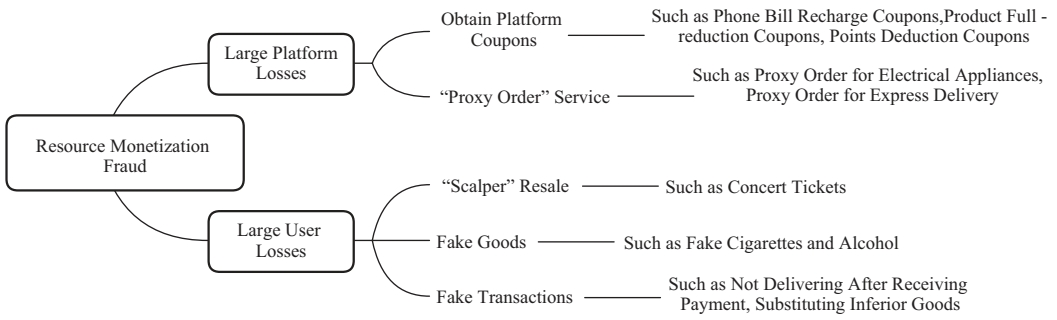


FIGURE 2.18 Common resource monetization fraud schemes.

The following two detailed cases illustrate the specific processes of resource monetization fraud.

The first typical case involves members of the black/gray industry posing as “unofficial agents” of platforms to sell resources for profit, such as “order placement services” on certain e-commerce platforms. “Order placement services” refer to scenarios where users do not generate orders directly from platform merchants but instead place orders through third parties. These third parties interface with the platform, typically possessing merchant accounts that allow them to source goods at wholesale prices. Some third parties also have access to inventory from other platforms. The black/gray industry exploits platform price protection rules to profit from price differentials, causing economic losses to the platform. The “order placement service” process is illustrated in Figure 2.19.

The second typical case involves scalper reselling. For instance, a certain brand of baijiu has long been described by users as “the more expensive it is, the harder it is to buy; the harder it is to buy, the more expensive it becomes.” Consequently, scalper reselling occurs. First, the “ringleader” (organizer) uses crowdsourcing to distribute baijiu purchase tasks on task platforms. Then, “cattle” (human crowdsourced participants) engage in scattered purchases. Upon successful purchase, the “cattle” provide the order number to the “bull” to receive a commission, while also obtaining bank rebates and points through credit card payments. Next, after acquiring baijiu resources through the “cattle,” the “bull” colludes with logistics personnel to secretly hoard the baijiu. Finally, the “bullhead” sells the liquor at inflated prices to end users, forcing ordinary consumers to purchase at exorbitant costs. The scalper resale process is illustrated in Figure 2.20.

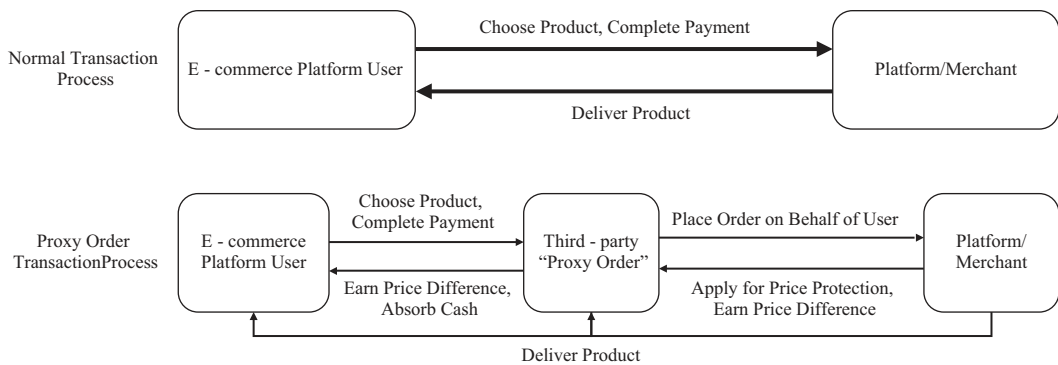


FIGURE 2.19 “Order placement service” process.

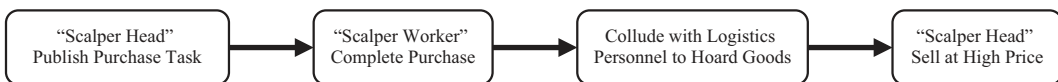


FIGURE 2.20 Process of “scalper” resale.

2.4 ANALYSIS OF FRAUDULENT GAINS

Across the entire product lifecycle—from user acquisition campaigns, registration and login, app usage (including social interactions), to revenue settlement—the black/gray industry infiltrates every stage, exploiting various methods to generate illicit profits. From a fraud profit perspective, the investment costs and returns for members of the black/gray industry vary across different stages. From a security defense perspective, the ideal strategy is to deter these operators by raising the barriers to their illicit activities and reducing their profits.

Across different phases of platform traffic growth, the disparity between the costs incurred by the black/gray industry and its fraudulent profits is significant. Figure 2.21 illustrates the relationship between single-instance fraud profits and investment costs for typical fraudulent methods. In the pre-traffic phase, promotion settlement fraud yields low single-instance profits but requires relatively low investment. Conversely, in the post-traffic phase, resource monetization fraud and telecom fraud generate higher single-instance profits but also demand higher investment costs. Consequently, combating black/gray industries becomes more challenging in later phases, necessitating increasingly complex data dimensions and algorithms.

During the pre-traffic phase, mainstream governance and prevention methods include device fingerprinting technology, CAPTCHA verification, and simple rules. In the mid-traffic phase, when users enter the platform and become active, governance and prevention methods must be upgraded from the initial solutions. This requires building business rule expert engines, machine learning countermeasure solutions, and complex network models. In the post-traffic phase, multimodal solutions and novel countermeasure approaches must be introduced. Wherever profit opportunities exist, black/gray industries will persistently attempt to circumvent defenses. At this stage, governance and defense

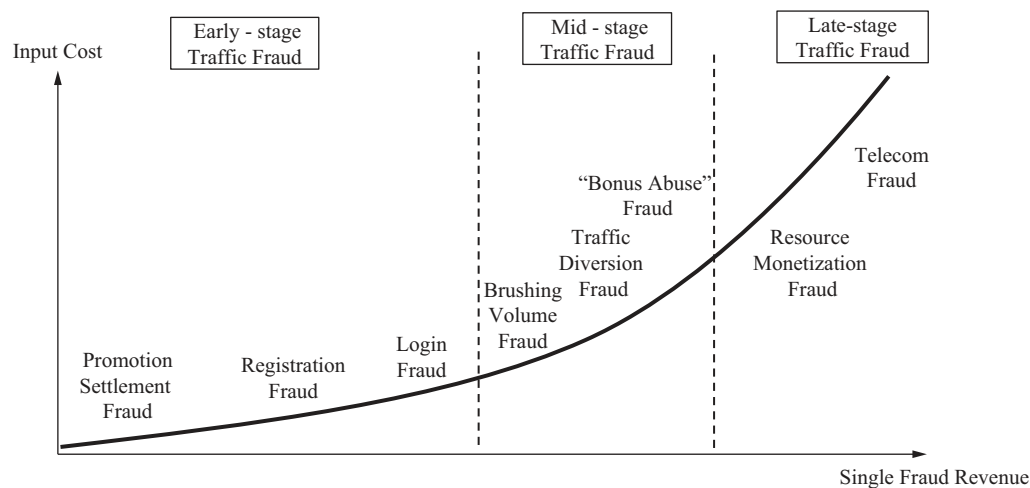


FIGURE 2.21 Relationship between single-instance fraud profit and investment cost for typical fraud methods.

measures must be upgraded based on specific scenario challenges. Subsequent chapters of this book will address key issues encountered throughout the platform traffic lifecycle, introducing common mainstream technologies and countermeasures for data security governance and prevention.

2.5 SUMMARY

This chapter primarily outlines typical traffic fraud tactics employed throughout the lifecycle of traffic campaigns, including pre-traffic phase promotion settlement fraud, registration fraud, and login fraud; mid-traffic phase “freeloading” fraud, brushing volume fraud, and traffic diversion fraud; as well as post-traffic phase telecom fraud and resource monetization fraud.

Regardless of the fraud scenario, the black/gray industry exploits loopholes in business rules to monetize platform traffic for illicit gains. As business rules continuously improve, black/gray industry fraud tactics also evolve. From automated tools to human crowdsourcing for fraud, the integration of black/gray industry tools and industrial chains deepens. These actors disguise themselves as legitimate users to bypass platform restrictions. Therefore, effectively identifying traffic fraud has become a critical issue in platform development.

Traffic Data Governance and Feature Engineering

For solutions identifying anomalous traffic within big data streams, richer and higher-quality input data significantly enhances the precision and recall rates of fraudulent traffic detection. However, the raw log data collected is often of poor quality and cannot be directly used for model training. It requires data governance and feature engineering to transform it into feature data that models can utilize directly for fraudulent traffic detection.

This chapter primarily addresses traffic data generated during the evolution of user behavior. It first outlines the fundamental data formats, then introduces the data governance process, and finally generates model-ready feature data through feature engineering.

3.1 BASIC DATA FORMATS

From the perspective of user behavior evolution, data generated during the pre-traffic phase, mid-traffic phase, and post-traffic phase is illustrated in Figure 3.1. In the pre-traffic phase, users have not yet entered the platform, and core data primarily consists of identity attributes and network environment parameters, such as IP addresses, accounts, and device-related metrics. During the mid-traffic phase, users become active across various platform scenarios, generating rich behavioral and user-generated content (UGC) data—including personal information like nicknames and avatars, as well as text, audio, and video content they publish. In the post-traffic phase, user activities center on social interactions and transactions, yielding more comprehensive data such as social graph information. The following sections detail the characteristics of data across these three distinct phases.

3.1.1 Pre-Traffic Phase Data

The pre-traffic phase encompasses stages such as users clicking on app promotion ads, registering accounts, and downloading/installing the app. During this phase, users first generate network environment data and device environment data, followed by providing basic account information during the registration process. The promotion and account registration process for a certain app is illustrated in Figure 3.2.

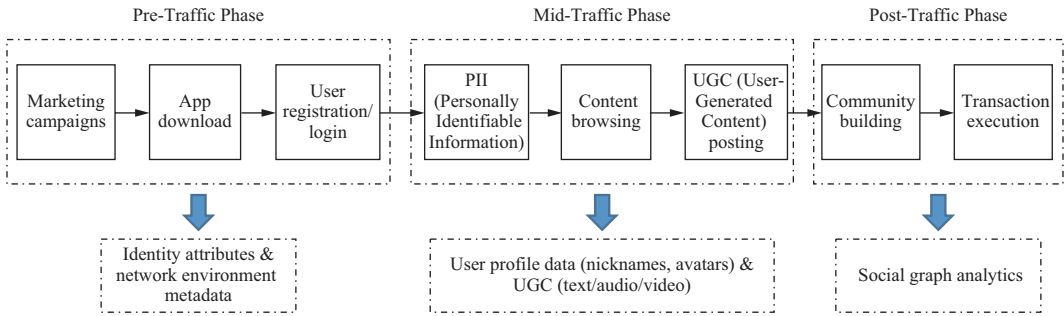


FIGURE 3.1 Data generated during the pre-traffic phase, mid-traffic phase, and post-traffic phase.



FIGURE 3.2 App promotion and account registration process.

For identifying traffic fraud, key data available during the pre-traffic phase includes network environment data, device environment data, and basic user account information, as detailed below:

- **Network Environment Data:** This refers to information about the user's network environment, with the IP address being the most commonly used metric. If a registered user's IP address originates from a proxy server or exhibits abnormal distribution patterns (e.g., predominantly from remote regions or overseas), it may indicate an IP address used by the black/gray industry.
- **Machine Environment Data:** Machine environment data refers to the environment of the user's device, including hardware, software, and status information such as whether scripts are running on the device, browser cookies, GPS, temperature, gyroscope, and the user-agent submitting the registration request. If script execution is detected on the device or multiple cookies exist, this may indicate automated behavior generated by the black/gray industry.
- **Basic User Account Information:** Basic user account information encompasses two aspects. First, it includes the personal details provided during registration, typically

a mobile phone number or email address. Second, it covers information such as the username entered during registration. If the mobile phone number used for registration is a blacklisted number, secondary number, or virtual number, and the registered usernames exhibit significant similarity patterns, this may indicate bulk registration activities originating from the black/gray industry.

3.1.2 Mid-Traffic Phase Data

The mid-traffic phase refers to the period from when a user enters the app and begins activities until they form community relationships. During this phase, user activities are generally diverse, including setting personal nicknames, avatars, and profiles; posting text, audio, or video updates; commenting on other users' content; and participating in activities by liking and sending messages, generating massive amounts of UGC. Traffic fraud during this phase primarily involves the publication of malicious referral content. Members of the black/gray industry induce other users to click their links or download other apps to generate profits.

Mid-traffic phase data primarily includes user-generated text, audio, image, video, and behavioral data, as detailed below.

- **Text Data:** User nicknames, personal profiles, comment text, trending topics, video bullet comments, and similar data fall under text data.
- **Audio Data:** Voice updates and audio components within videos constitute audio data. When text-based data faces stringent crackdowns, the anti-fraud battle shifts to the audio domain, where black/gray industry operators convert traffic-driving text into audio formats for dissemination.
- **Image Data:** User profile pictures, posted image updates, and video thumbnails are all image data susceptible to exploitation by the black/gray industry.
- **Video Data:** Mainstream apps now support video content across comment sections, posts, messages, and video uploads. Since videos are fundamentally composed of sequential images, sampling video data yields multiple image frames for subsequent processing.
- **Behavioral Data:** User actions within apps constitute behavioral data, including browsing, searching, clicking, liking, commenting, bookmarking, and check-ins. Behavioral sequences reveal behavioral patterns. Since members of the black/gray industry often employ bulk operations, their behavioral paths exhibit high similarity.

3.1.3 Post-Traffic Phase Data

The post-traffic phase refers to the stage where users establish community relationships with other users or groups within the app and engage in transactional activities. Fraudulent traffic in this phase primarily involves members of the black/gray industry spreading malicious information to defraud users or conducting fake transactions to gain illicit profits.

Post-traffic phase data primarily includes social relationship data and transaction relationship data between users, as detailed below:

- **Community Relationship Data:** Community relationships encompass social connections between users and other users or groups. The community relationship graph is illustrated in Figure 3.3, where light colors represent individual users and dark colors denote groups composed of multiple users. These groups are formed based on in-app users and take various forms, such as group-buying teams, interest-based communities, or fan groups. Connections between individual users represent social interactions such as following, adding friends, or liking. Connections between individual users and groups represent social relationships involving actions like creating, managing, or joining groups.
- **Transaction Relationship Data:** Transaction relationships encompass user-to-merchant transactions and user-to-user transfers.

The transaction relationship graph is shown in Figure 3.4, where light colors represent ordinary users and dark colors represent merchants. Connections between ordinary users represent transfer relationships, while connections between ordinary users and merchants represent shopping relationships.

3.1.4 Traffic Data Characteristics Comparison

The preceding sections highlighted the three distinct phases of user traffic evolution and their corresponding primary data patterns. Each phase presents unique security challenges, associated with different data sources and countermeasures. Specific characteristics of traffic data across different periods are compared in Table 3.1.

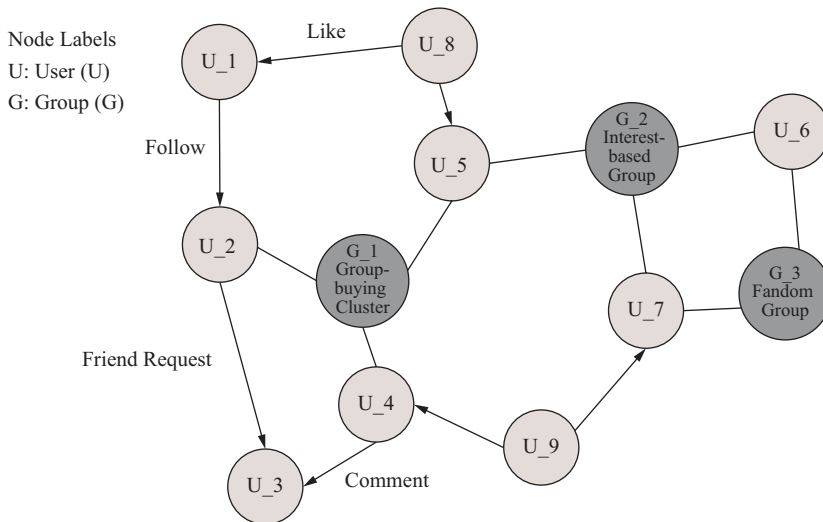


FIGURE 3.3 Community relationship graph.

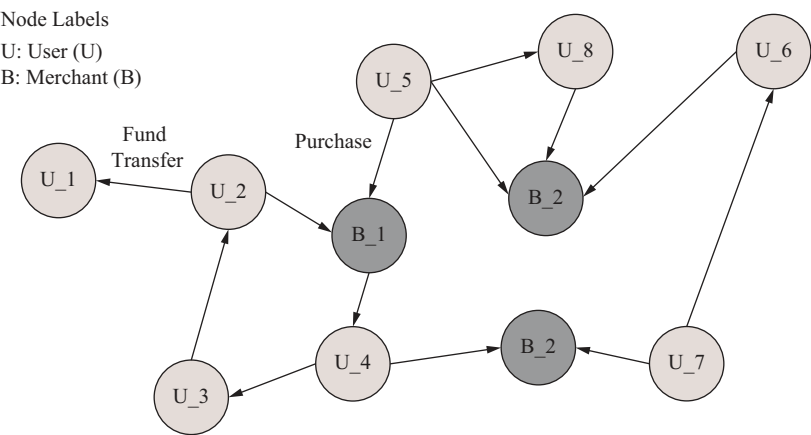


FIGURE 3.4 Transaction relationship graph.

TABLE 3.1 Comparison of Specific Characteristics of Flow Data Across Different Periods

Period of Data	Pre-Traffic Phase	Mid-Traffic Phase	Post-Traffic Phase
Data format	Account and device information	Multimodal content information	Graph relationship information
Characterization dimensions	Environmental and identity characterization	Content and behavior profiling	Community relationship characterization
Feature dimensions	Simple	Diverse	Complex
Computational complexity	Low	Medium	High
Countermeasures	Preemptive protection, primarily through monitoring and basic restriction rules	Individual targeting, precise identification of fraudulent accounts	Organized crime crackdown, comprehensive coverage of traffic fraud

Table 3.1 compares the characteristics of traffic data across different periods, focusing on data format, profiling dimensions, feature dimensions, computational complexity, and countermeasure approaches.

- **Data Formats and Characterization Dimensions:** Pre-traffic phase traffic data primarily comprised account identity attributes, network environments, and device information, characterized through environmental and identity dimensions. Mid-traffic phase traffic data centered on UGC and behavioral data, characterized through content and behavioral dimensions. Post-traffic phase traffic data focused on social relationships and transaction graph information derived from user interactions and transactions, characterized through social relationship dimensions.
- **Features Data and Computational Complexity:** The pre-traffic phase saw traffic data dominated by identity and environmental information, with smaller data volumes and lower computational complexity. During the mid-traffic phase, user-generated text, images, videos, and behavioral sequences were included, showcasing richer

content and larger data scales with moderate processing complexity. During the post-traffic phase, user friends, groups, and transaction records were included, exhibiting greater diversity in data forms and higher processing complexity.

- **Countermeasure Approach:** Pre-traffic phase traffic data primarily supports initial defense mechanisms, intercepting obviously blacklisted users and anomalous network/device traffic during registration—serving as the first line of defense against the black/gray industry. Mid-traffic phase traffic data leverages user content data to enable precise targeting of individuals, effectively purging fraudulent content from the app. Post-traffic phase traffic data focuses on detecting organized fraud rings, significantly improving the detection recall rate of fraudulent traffic.

Transforming raw log data into high-quality features data usable by models requires data governance to process data into high-quality data, followed by feature engineering to transform this high-quality data into model-ready features data. The following sections focus on detailing data governance and feature engineering.

3.2 DATA GOVERNANCE

The core objective of data governance is to process raw data securely and to a high standard while ensuring compliance. Data governance primarily encompasses data collection, data cleansing, data storage, and data computation.

3.2.1 Data Collection

Data sources can be categorized into three main types: first, business data reported directly from the app itself, such as UGC and user operation sequences; second, data collected via Software Development Kit (SDKs) integrated into the app, such as device environment and parameter information gathered through device fingerprinting; and third, third-party industry data, including public IP reputation databases, malicious proxy IP databases, speed dial IP databases, fraudulent intermediary number databases, and virtual operator blacklisted number databases. The data collection process for different sources is illustrated in Figure 3.5.

- **Product Business Data:** Business data is obtained through event tracking by embedding data collection logic within product functionality to capture user actions and events.
- **Integrated SDK Data:** Typically collected through actions occurring in fixed scenarios or triggered periodically, such as monitoring device environment and network data. If it is a business-customized SDK, user behavior data and business data can be collected by issuing rules.
- **Industry Third-Party Data:** To reduce costs, members of the black/gray industry often reuse their resources and commit fraud across multiple platforms. Therefore, third-party data from other businesses and platforms can be acquired through purchase or collaboration to build an anti-fraud system for your operations.

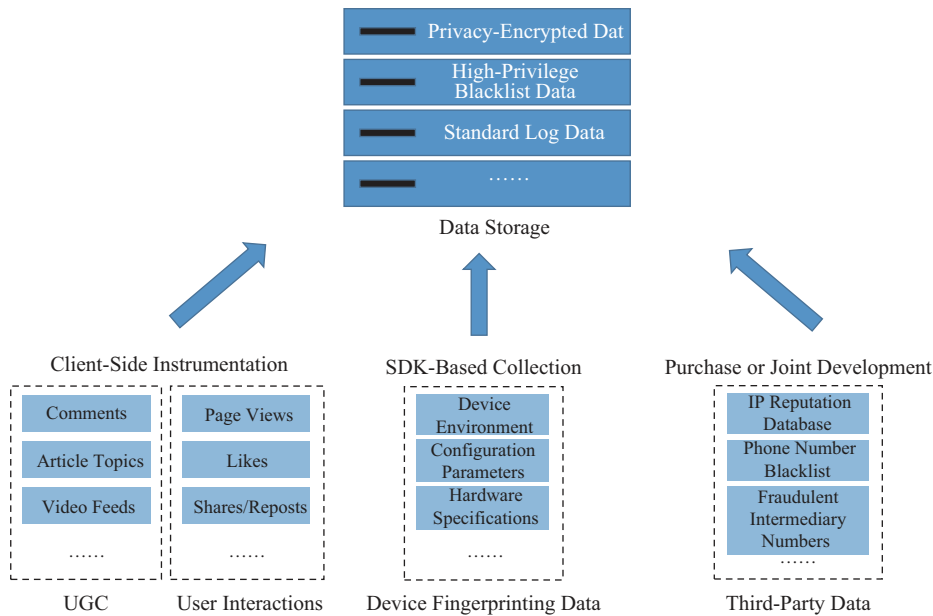


FIGURE 3.5 Data collection process for different sources.

3.2.2 Data Cleansing

Data cleansing primarily transforms raw data into high-quality data characterized by completeness, uniqueness, consistency, and accuracy. The cleansing process includes handling missing data, addressing anomalies, and performing normalization and standardization.

- **Missing Data Handling:** Missing data refers to incomplete information. Handling methods include column deletion, row deletion, and missing value imputation. Column deletion is applicable when the missing rate of a column's attribute values is excessively high, leading to the removal of that column. Row deletion is appropriate when the vast majority of attribute values for a sample are missing. In such cases, it is impossible to identify the sample through data modeling, necessitating the deletion, discarding, or special review of the sample data. When only a small number of attribute values are missing for an individual, special values are typically used for imputation, such as filling with -1 . Subsequent models will also learn to treat this type of missingness as a special case.
- **Data Anomaly Handling:** Data anomalies refer to instances where data values deviate significantly from the expected range due to errors during entry or submission. Handling methods include deletion, imputation, and special flagging. Before addressing anomalies, data must first be assessed to determine if it qualifies as an outlier. Common examples of outliers in traffic data are shown in Table 3.2.

TABLE 3.2 Examples of Common Anomalies

Data Attribute	Outlier	Anomaly Description
IPv4 address	119.611.20.56	The IP address range should be 0.0.0.0 to 255.255.255.255
Device ID IMEI	0000001234567890	The first 6 digits of the IMEI are the model allocation code. 000000 is an invalid model allocation code
Mobile phone number	136XXXX001	Domestic mobile phone numbers consist of 11 digits; this mobile phone number has 10 digits

TABLE 3.3 Differences and Examples of Data Normalization and Data Standardization

Processing Method	Use Case	Example
Data normalization	Data distribution is relatively stable	User daily visit duration (hours), with data ranging from 0 to 24, making it suitable for data normalization processing
Data standardization	Data distribution is relatively unstable with significant inter-individual variation	Daily user page views exhibit substantial inter-individual variation and instability, making data standardization an appropriate approach

- **Data Normalization and Standardization:** Data normalization compresses raw data into values between 0 and 1 while preserving its original distribution. Data standardization alters the distribution itself, mapping raw data to follow a standard normal distribution. Both normalization and standardization enhance the effectiveness of model training. Data normalization can be applied when the data distribution is relatively stable; however, when the distribution is unstable and individual variations are significant, data standardization is preferred. The differences between data normalization and standardization, along with examples, are shown in Table 3.3.

3.2.3 Data Storage

Data storage involves categorizing and storing data from the pre-traffic phase, mid-traffic phase, and post-traffic phase through processes such as anonymization, encryption, and permission separation. This facilitates targeted, compliant, and efficient utilization later. Factors considered when selecting data storage methods, as shown in Figure 3.6, can be divided into three aspects: time overhead, security, and usage cost.

- **Time Overhead:** Time overhead in data storage refers to the CPU and I/O time consumed during data write and read operations. Timely data storage ensures that reads and writes are completed within specified timeframes, preventing delays in data usage across other processes.
- **Security:** Privacy-sensitive data, high-privilege black database data, and ordinary log data should be stored separately to ensure permission isolation. Additionally, account-related privacy data such as mobile phone numbers and ID numbers should be stored in an encrypted form. Encryption methods can include the MD5 hash algorithm, Advanced Encryption Standard, or Rivest-Shamir-Adleman (RSA) cryptosystems.

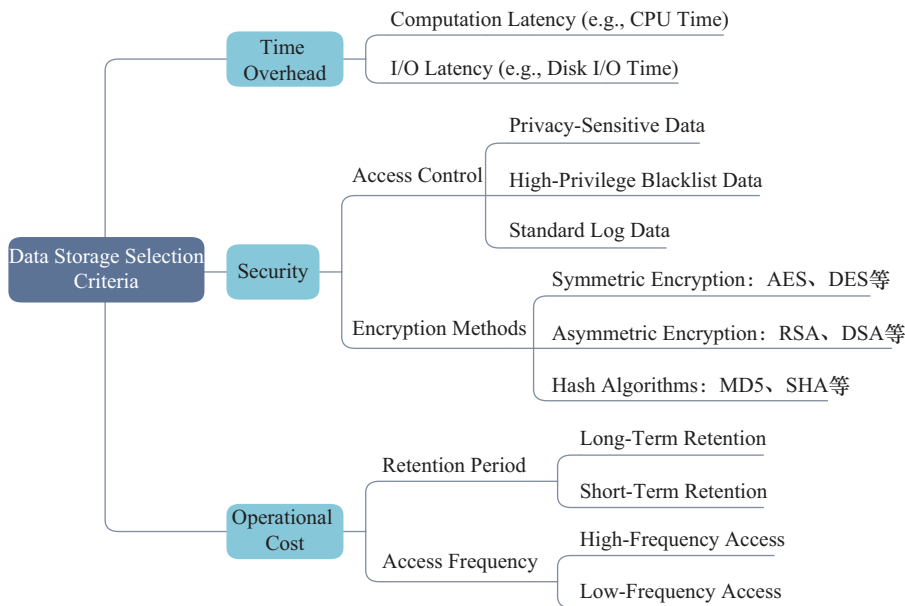


FIGURE 3.6 Factors to consider when selecting data storage methods.

- **Usage Costs:** Employing different storage methods for data across various usage scenarios can effectively reduce storage costs. For instance, log data requiring long-term retention but infrequent access—which typically involves massive volumes—can be stored using the Hadoop distributed file system. Conversely, black box data demanding low query latency and frequent access—which generally involves smaller data volumes—can be stored using high-performance key-value databases like Redis.

3.2.4 Data Processing

Data computation processes traffic data based on business requirements for data analysis. When selecting a computation method, the primary consideration is timeliness—whether real-time processing is required. For text, image, or graph data, computational cost is also a factor. The choice of computation primarily involves selecting a processing model, with big data computation models broadly categorized into four types, as shown in Figure 3.7.

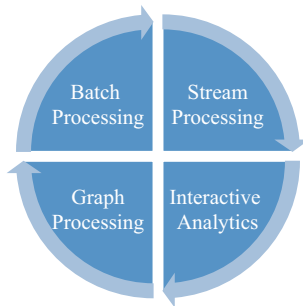


FIGURE 3.7 Four big data computing models.

- **Batch Processing:** Batch processing involves handling data in batches using tools like Hadoop or Spark. For traffic fraud detection services, offline processing of massive log data can employ batch processing.
- **Stream Processing:** Stream processing involves real-time analysis of current data streams. Tools like Storm, Spark Streaming, and Flink are commonly used for real-time computation. In traffic fraud detection services, stream processing can handle the real-time retrieval and processing of data by online models.
- **Interactive Query Computing:** Interactive query computing primarily handles real-time queries on massive datasets with high timeliness requirements. Tools like Dremel and Hive can be used. In traffic fraud detection services, interactive query computing can be employed to flag blacklisted traffic in real-time and provide evidence for such flagging.
- **Graph Computing:** Graph computing is primarily designed for processing graph-structured data and can utilize tools like PowerGraph and GraphX. In traffic fraud detection services, graph computing can be used to process community relationship data and transaction relationship data in the post-traffic phase.

3.3 FEATURE ENGINEERING

High-quality data refined through data governance can be further processed via feature engineering to create features data that is directly usable by models. Since the quality of features data directly determines a model's upper limit, the construction of feature engineering forms a critical foundation for fraud detection models.

Feature engineering involves abstracting and extracting relevant domain knowledge from raw data, mapping its distribution from the original space to the feature space. This process enhances the separability between fraudulent traffic and normal traffic in the feature space. Feature engineering primarily encompasses feature construction, feature evaluation and selection, and feature monitoring.

3.3.1 Feature Construction

Feature construction involves processing raw log data to generate features data. Depending on specific business scenarios, data processing methods vary widely but can be broadly categorized into three types: label features based on domain expertise, features derived from supervised learning, and features derived from unsupervised learning. The distinctions between these approaches and their applicable scenarios are summarized in Table 3.4.

- **Expert-based label features:** Analyze data performance across different scenarios based on expert experience to generate corresponding label features. For example, login times for accounts under group control in the black/gray industry show significant differences compared to normal user accounts. Normal users typically log in during leisure hours, such as weekends and evenings after work, whereas logins from the black/gray industry occur at irregular times—even in the middle of the night—and

TABLE 3.4 Differences and Use Cases of Feature Construction Methods

Feature Construction Method	Construction Cost	Interpretability	Advantage	Disadvantage
Label features based on expert experience	High	Strong	Features align with product perception, high versatility, generally stable, and high accuracy in fraud detection	Over-reliance on expert experience; human expertise determines the upper limit of adversarial resistance
Features based on supervised learning	Low	Moderate	Based on target samples, fully mining data, strong feature representation capability, high accuracy and recall rate for fraud detection	Completely dependent on sample quality; adversarial attacks can lag and are easily circumvented
Features based on unsupervised learning	Low	Weak	Independent of target data, actively learns data patterns, strong feature representation capability, high accuracy and recall in fraud detection	Depends on whether the sampled training dataset is representative

often in concentrated bursts. Therefore, statistical features can be generated by counting the number of logins across different time periods throughout the day.

- Features based on supervised learning: Unlike label features generated through expert experience, supervised learning features treat traffic fraud as positive samples and normal traffic as negative samples, learning features through supervised learning. For example, features can be constructed by calculating the target group index to validate data preferences across different samples.
- Features based on unsupervised learning: Unsupervised learning features primarily focus on the inherent characteristics of the data itself to mine data. For textual information, algorithms like word2vec or doc2vec can be used to learn features; for graph relationship data, algorithms like node2vec can be employed to learn features.

3.3.2 Feature Evaluation and Feature Selection

Feature evaluation and selection involve assessing each feature's representational capability for the problem after feature construction but before building the traffic fraud monitoring model, and selecting effective features for modeling. The main components of feature evaluation and selection are shown in Figure 3.8.

Feature evaluation primarily involves two dimensions: assessing the correlation between features and sample labels, and evaluating the correlation among the features themselves.

- Assessing feature-label correlation: Calculating the IV value allows for the evaluation of each feature's relevance to the sample label, enabling the removal of features with excessively low IV values. For example, if a feature's IV value is less than 0.02,

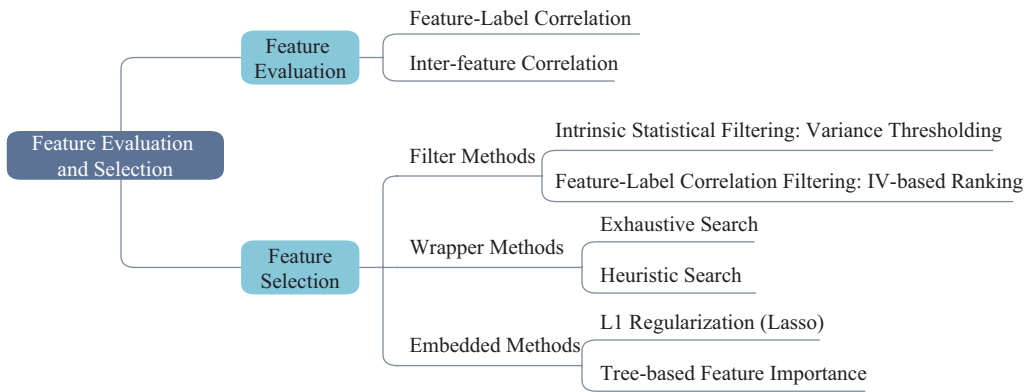


FIGURE 3.8 Key components of feature evaluation and feature selection.

it is considered to have no contribution to predicting the sample label and should be discarded. Conversely, if a feature's IV value exceeds 0.5, it indicates an extremely strong correlation with the sample label, potentially revealing the label. In such cases, manual analysis is required to confirm that the feature is not problematic before it can be included in the model.

- Assessing feature correlation: Various methods exist for evaluating feature correlation, such as covariance, Pearson's correlation coefficient, and mutual information. Pearson's correlation coefficient is commonly used to assess feature correlation. If the correlation coefficient between any two features exceeds 0.8, it indicates a high degree of correlation between them. Including both features in the model at this point does not yield additional information gain but may destabilize the model. Therefore, the feature with the higher IV value should be retained, while the feature with the lower IV value should be discarded or manually removed.

Common feature selection methods include filtering, wrapping, and embedding.

- Filter-based: Each feature is evaluated individually and then ranked based on evaluation results. The top N ranked features are selected for inclusion in the model. Feature evaluation focuses on two aspects: first, the feature's own statistical properties, such as assessing data dispersion through variance; second, the correlation between the feature and the label, where higher correlation contributes more to label prediction.
- Wrapper-based: This approach integrates feature selection with model training. Common wrappers include exhaustive search and heuristic search. Exhaustive search trains models using every possible feature combination, selecting the optimal one—a time-consuming process. Heuristic search iteratively builds models, retaining the most effective features or discarding the least effective ones after each iteration, then continues training with the remaining unselected features.

- **Embedding approach:** The embedding approach skips feature selection and directly allows the model to choose which features to incorporate. Two methods exist for feature selection using embeddings: first, construct models using all features to obtain feature importance scores. Then, rank all features based on these scores and select the top N ranked features for the final model. Second, incorporate an L1 regularization term into the loss function, which promotes learning sparse parameter matrices and automatically filters out unimportant features.

In practical applications, the choice of feature selection method depends on the sample size and feature dimension. When the feature dimension is low and the dataset is small, wrapper or embedding methods can be used for feature selection. When feature dimensions are high and the data volume is large, the computational complexity of wrapper and embedding methods becomes prohibitive. In such cases, filtering methods with low computational complexity and short runtimes are the only viable options. Feature selection methods can also be combined: first, use filtering to eliminate features that are clearly irrelevant to modeling, thereby reducing feature dimensions; then apply wrapper or embedding methods for feature selection.

3.3.3 Feature Monitoring

Feature monitoring refers to tracking the stability of feature data based on the actual features entering the model, preventing model failure caused by unstable data. The stability of feature data can be assessed by calculating the Population Stability Index (PSI) to determine whether features are stable. Generally, when the PSI value exceeds 0.25, it indicates significant changes in the feature distribution, requiring specific analysis by data operations personnel. The PSI monitoring curve for a certain feature is shown in Figure 3.9. At

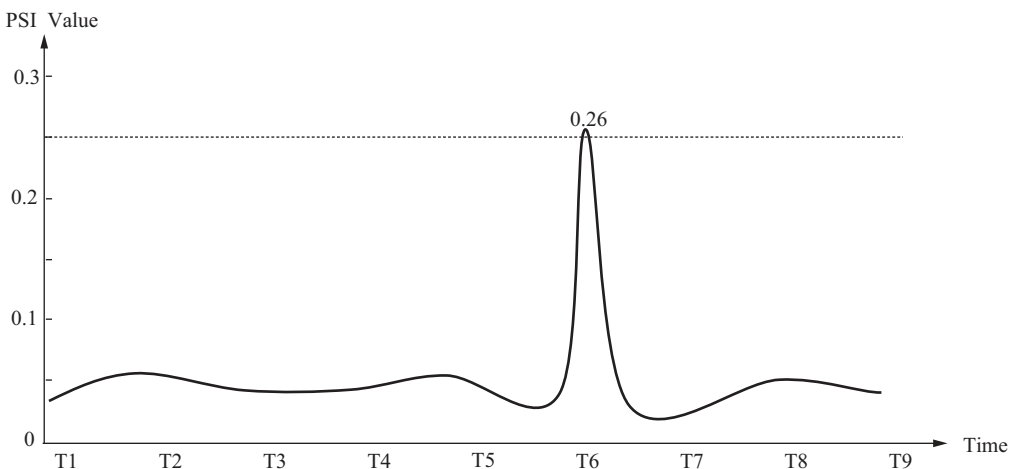


FIGURE 3.9 PSI monitoring curve for a feature.

time T6, the PSI value increased to 0.26, indicating that data operations personnel should analyze whether the model has produced anomalies. For details on monitoring and managing feature stability, readers can refer to Chapter 11 of this book.

3.4 SUMMARY

This chapter primarily introduces the fundamental data patterns of traffic across different phases and the process of transforming log data into feature data suitable for model training through traffic data governance and feature engineering. First, it details the fundamental data patterns and characteristics of traffic across different phases (pre-traffic phase, mid-traffic phase, and post-traffic phase). Next, through data governance, raw logs are compliantly encrypted and cleaned into high-quality data. Finally, the feature engineering module transforms this high-quality raw data into feature data understandable by models. When implementing traffic data governance and feature engineering in practice, it is essential to consider the data's form, characteristics, and business quantification standards. Appropriate traffic data governance and feature engineering methods should be selected based on actual adversarial scenarios.

Device Fingerprinting Technology

In the mobile internet era, users can perform nearly all traffic-related activities on their phones—shopping via e-commerce apps, watching movies on video apps, and socializing through social apps—using only a single device. For the black/gray industry seeking to profit from mobile traffic platforms, substantial device resources are often necessary. These resources may consist of simulated fake devices via emulators or real devices controlled through group control systems. To distinguish between actors from the black/gray industry and legitimate users, device fingerprinting must be established to identify genuine devices, fake devices, and fraudulent operations.

Early traditional device identification technologies primarily relied on IP addresses and cookie-based methods to tag user devices, tracking and recording multi-dimensional access information and terminal device details. However, traditional methods face limitations: IP-based identification suffers from low accuracy and poor stability; cookies stored on end devices are easily tampered with; privacy restrictions limit the number of mobile device identifiers; and the black/gray industry exploits flashing tools and emulators to maliciously alter or forge devices for fraudulent activities. Consequently, device fingerprinting technology emerged as a solution. Device fingerprinting borrows the concept of fingerprints from biology, using a device's "fingerprint" (i.e., the relative differences in certain information between devices) as its identifier. Consequently, device fingerprints possess uniqueness and stability.

This chapter primarily introduces the value of device fingerprinting in anti-fraud applications, its technical principles, and the implementation of the solution.

4.1 THE VALUE OF DEVICE FINGERPRINTING

As a unique identifier for user terminal identity, device fingerprinting can be applied in scenarios such as distinguishing genuine from fraudulent devices and tracking user traces. In traffic fraud scenarios, device fingerprinting serves as one of the foundational

technologies for detecting fraudulent activities. Its practical value in anti-fraud operations, as illustrated in Figure 4.1, is significant across various scenarios including registration fraud, login fraud, machine-simulated device fraud, brushing volume fraud, and traffic settlement fraud.

- **Marking High-Risk Devices:** Device fingerprinting effectively detects environmental information and behavioral risks during device operation. It identifies emulator behavior, root access, virtual framework invocations, debugging activities, attack injection attempts, and automated bulk actions. This technology distinguishes between genuine and simulated devices while generating diverse risk labels.
- **Distinguishing Login Fraud and Account Theft:** Device fingerprinting identifies group control devices that generate fake accounts in bulk within set time windows. It detects login attempts on spoofed devices, frequent logins across multiple accounts, and suspicious credential stuffing attempts. It also identifies unusual login devices and detects GPS location drift or spoofing.
- **Detecting Traffic Fraud in Promotion Settlements:** During app advertising campaigns, the black/gray industry generates fake traffic to defraud advertisers of settlement fees, preventing ads from reaching genuine users. Examples include creating massive numbers of fake devices to repeatedly perform fraudulent ad clicks and fake app installations. By leveraging the uniqueness of device fingerprints, fraudulent activities can be identified, effectively ensuring traffic authenticity and enabling ads to genuinely reach target audiences.
- **Identifying Fake Account Bindings in “Freeloading” Fraud:** To ensure fairness and authenticity, legitimate platform marketing activities restrict participation frequency per device and account. However, the black/gray industry can forge devices to link

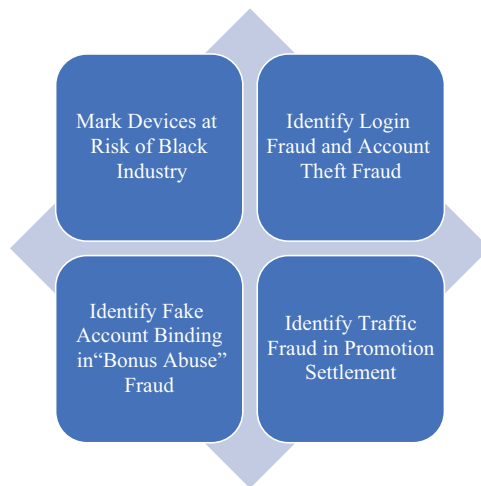


FIGURE 4.1 Value of device fingerprinting in anti-fraud operations.

a single account to multiple fake devices or associate multiple accounts with a single device, enabling repeated “freeloading” activities. Device fingerprinting technology effectively identifies device uniqueness. By analyzing device characteristics and integrating business context, appropriate countermeasures can be established.

The common scenario examples above demonstrate the application value of device fingerprinting, with even richer scenarios emerging in practice. Because device fingerprinting plays a critical role in fraud prevention, black/gray industry operators employ various technical means to circumvent it, potentially undermining the ability to uniquely identify a device. Therefore, developing a robust device fingerprinting solution is extremely important. The following section focuses on the approach to building a device fingerprinting solution from scratch.

4.2 TECHNICAL PRINCIPLES

Device fingerprinting technology can be defined as the process of utilizing a device’s multi-dimensional feature attributes to generate a unique and stable device identifier for each device through algorithmic processing. The two core characteristics of device fingerprints are uniqueness and stability. The uniqueness of device fingerprinting is demonstrated by the fact that different devices generate distinct identifiers, with each device uniquely corresponding to one identifier—meaning device fingerprints exhibit an extremely low collision rate. The stability of device fingerprinting is evident in that device fingerprints do not change with device usage or environmental variations. The following sections introduce the foundational concepts, development history, and generation methods of device fingerprinting technology.

4.2.1 Basic Concepts

The fundamental principle of device fingerprinting involves collecting characteristic attribute information from client devices and uploading it to a server. After performing encrypted and compliant data operations, an algorithm generates a unique device identifier to mark the device. The characteristic attribute information of client devices can primarily be categorized into hardware/software-related features of the system and applications, as well as business logic and device behavioral attributes.

System and application-related hardware/software attributes are primarily obtained through reserved APIs in the device’s operating system and applications to gather client hardware/software information. Concurrently, subtle differences across devices in network communication, data computation processing, and image resource loading are collected to construct feature factors.

Business logic and device behavior attributes are primarily captured by logging device and service-related behaviors during usage. These behaviors are then utilized as feature factors for construction.

Upon acquiring these device attribute features, the server generates device fingerprints through algorithms and stores device ID mapping relationships. When a device fingerprint query is requested, the client collects features data and queries the server. If the

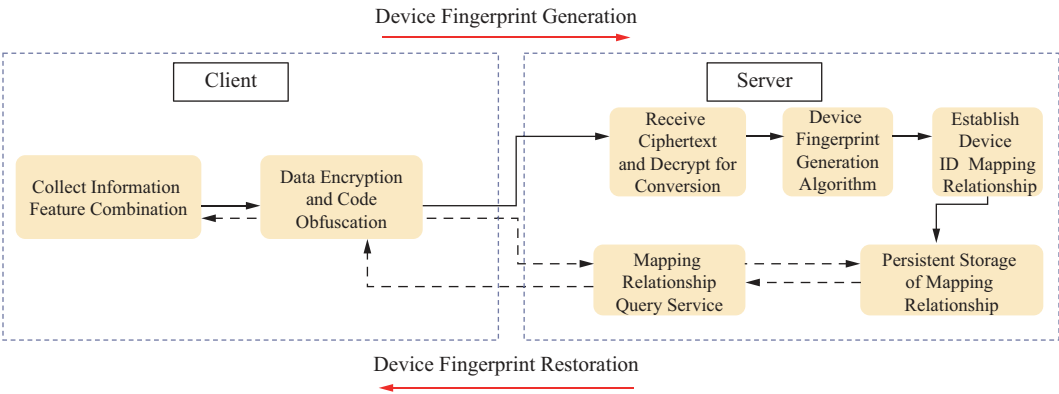


FIGURE 4.2 Basic process of device fingerprinting technology.

device fingerprints match, the devices are identified as the same; otherwise, they are considered new devices. The basic workflow of device fingerprinting technology is illustrated in Figure 4.2.

4.2.2 Development History

The development of the internet gave rise to device fingerprinting technology. In the mobile internet era, the diversification of device types and access channels has driven the evolution of user behavior and fingerprint generation schemes. Simultaneously, counterfeit device technologies employed by the black/gray industry have continuously advanced. The development history of device fingerprinting technology has primarily undergone six stages, as illustrated in Figure 4.3.

- Phase One: This phase primarily relied on IP and cookie technologies, which are precursors to device fingerprinting. At this stage, achieving unique user identification was challenging.
- Phase Two: This phase relied on foundational hardware information, system information, and manufacturer device identifiers of mobile devices, such as the International Mobile Equipment Identity (IMEI), International Mobile Subscriber Identity (IMSI), and Media Access Control (MAC) addresses commonly used in Android systems, as well as the Advertising Identifier (IDFA), App Developer Identifier, and Unique Device Identifier commonly used in iOS devices.

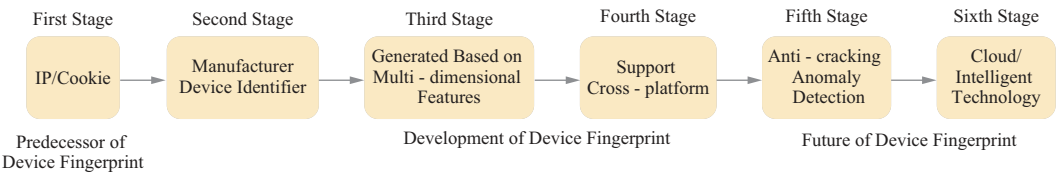


FIGURE 4.3 Evolution of device fingerprinting technology.

- Phase Three: This phase introduced multi-dimensional device information, such as network interface cards and device parameters, to construct device fingerprinting algorithms.
- Phase Four: With the rapid growth in device volume, platform diversity, and user base, along with the continuous advancement of counterfeit device technologies by the black/gray industry, phase four emerged. This phase focused on the uniqueness of device fingerprints while supporting cross-platform compatibility, leading to the development of device fingerprinting technologies that prioritize uniqueness. Examples include unified device identifiers that support different platforms such as apps, mini-programs, and H5 pages.
- Phase Five: As black/gray industry counterfeiting techniques evolve with increasing targeting, device fingerprinting technology now emphasizes anti-tampering and anti-reverse engineering protection capabilities, alongside intelligent sensing and detection in abnormal environments.
- Phase Six: Device fingerprinting technology enters a new phase of development, leveraging cloud computing and artificial intelligence technologies.

4.2.3 Generation Methods

Based on different collection and generation approaches, device fingerprinting technology can be primarily categorized into active generation, passive generation, and hybrid generation techniques.

Active generation involves collecting and generating device fingerprints at the frontend. The core approach includes embedding tracking points within applications or systems beforehand. Upon event triggering, data is gathered via SDKs or JavaScript scripts, encrypted, and transmitted to the server. Backend fingerprint algorithms then generate unique device identifiers and return them to the frontend. This technology primarily collects the following device information: MAC address, GPS location, Wi-Fi network list, device identifiers (e.g., IMEI and IDFA), unique app IDs, root access status, sensor attributes, installed app list, installation behavior, and browser properties. Active device fingerprinting enables high-response, low-latency data collection. However, the collected information is prone to change, leading to fingerprint instability. Furthermore, many dimensions of data collection face privacy restrictions and require user authorization, which also limits its applicability.

Passive device fingerprinting is a method that generates device fingerprints by analyzing network communication protocols on the backend. The core concept involves examining network packets generated during terminal-server communication connections. By leveraging OSI communication protocols and network information, multiple dimensional features are extracted. For instance, protocol parameter characteristics can differ even among identical operating systems. Features can be constructed using protocol parameters such as packet time-to-live, maximum segment size, Synchronize Sequence Numbers (SYN) packet size, and SYN window size. These are then combined with machine learning algorithms to create a unique identifier for the terminal device. Passive device fingerprinting technology

TABLE 4.1 Comparison of Active and Passive Device Fingerprint Generation Techniques

Generation Method	Advantage	Disadvantage	Implementation Method
Active device fingerprinting technology	Fast response speed, high accuracy, relatively simple	Limited by privacy protection, relatively low stability	Implemented via SDK or JavaScript scripting at the frontend
Passive device fingerprinting technology	High stability, not constrained by privacy protection	High complexity, slower response times, high development threshold	Implemented on the server backend based on the OSI protocol stack

offers simpler data collection with broader applicability compared to active methods while relatively preserving user privacy. However, its limitations include higher algorithmic complexity, longer response times, and a relatively high technical barrier. A comparison between active and passive device fingerprinting technologies is presented in Table 4.1.

Hybrid device fingerprinting technology integrates active and passive generation methods to leverage their respective strengths. Within a unified architecture, it correlates terminal-generated active features with server-generated passive device characteristics, using algorithms to produce a unique and stable device identifier.

It is particularly important to note that due to the prevalence of the black/gray industry, genuine or simulated devices are often modified to falsify device parameters at low cost. Consequently, the generated device fingerprints may also change. An integrated device modification software, as shown in Figure 4.4, supports one-click modification of hardware environment parameters, obtaining root privileges, forging app lists, cloud environment management, and device location simulation. If the generation scheme is overly



FIGURE 4.4 Integrated device modification software.

simplistic, it can be easily circumvented by the black/gray industry. Therefore, selecting the appropriate device fingerprint generation method is critical. Starting from the fifth and sixth phases mentioned above, the intelligent detection technology of device fingerprints will be additionally considered.

4.3 TECHNICAL IMPLEMENTATION APPROACH

Device fingerprinting technology can essentially be understood as a mapping process similar to a hash function—that is, the process of collecting certain data fields and mapping them into a device identifier. Section 4.2 introduced the basic concepts and technical principles of device fingerprinting technology. This section illustrates the implementation approach of device fingerprinting technology through examples.

4.3.1 Evaluation Metrics

When evaluating whether a device fingerprinting technology meets requirements, in addition to considering the uniqueness and stability of core attributes, security and usability must also be assessed in practical application scenarios. The four evaluation metrics for device fingerprints are shown in Figure 4.5.

- **Uniqueness:** Primarily determined by whether device fingerprints generated from multi-dimensional feature data processed through complex algorithms are unique across different devices. Collisions must be avoided, with high-availability scenarios typically requiring a collision-free rate exceeding 99.9%.
- **Stability:** This is primarily reflected in the device fingerprint remaining unchanged when a single device undergoes incremental updates or modifications to certain key elements. Stability is generally ensured through algorithms that generate fingerprints using multi-dimensional attribute values.
- **Security:** This is primarily reflected in the proactive generation of device fingerprints. During the frontend collection process, additional protective measures such as code encryption, code obfuscation, code hardening, and code verification are required.

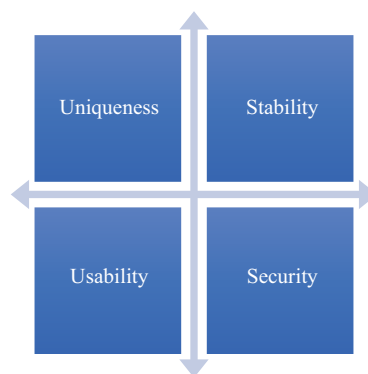


FIGURE 4.5 Four evaluation metrics for device fingerprinting.

This prevents the black/gray industry from developing targeted device fingerprint forgery tools after compromising the collection SDK or scripts, thereby significantly increasing the difficulty of reverse engineering by these actors. For passive device fingerprint generation, network communications in the backend collection process must be safeguarded against man-in-the-middle attacks and data tampering.

- **Usability:** This primarily manifests in practical deployment scenarios when applying device fingerprinting technology. Key considerations include ease of invocation, resource consumption, response time requirements, seamless integration, scalability, and support for high concurrency.

4.3.2 Feature Construction

Device fingerprinting features are generated from stable device parameters accessible within the device. Device fingerprints must balance stability and uniqueness. However, version incompatibilities and permission restrictions across different device operating systems may cause instability in collected fields, potentially compromising the fundamental requirements for fingerprint generation. Therefore, more stable device parameters are prioritized when collecting feature data fields. It is particularly important to note that due to regulatory protections for user privacy, certain parameters may be restricted or require user authorization.

Below are relatively stable device parameters commonly found in typical systems and environments, as illustrated in Figure 4.6.

- **Android System:** As Android is an open-source system, numerous custom systems based on Android development have emerged. When selecting parameters, compatibility issues across different system versions must generally be considered. Relatively stable device parameters in Android include boot random numbers, device identification codes, SIM card identification codes, physical addresses of Bluetooth and network cards, device serial numbers, device hardware information, and advertising IDs.

Android System	iOS System	Web Browse	Mini - program
Boot Random Number (ANDROID_ID)	Advertising Identifier (IDFA)	User Agent (User - Agent)	Mini - program ID (openid)
Device Identification Code (IMEI)	Manufacturer Identifier (IDFV)	2D Graphics API (Canvas)	Screen Parameters (screen)
SIM Card Identification Code (IMSI)	Network Card Physical Address (MAC)	3D Graphics API (WebGL)	Language (language)
Bluetooth and Network Card Physical Address (MAC)	System Boot Time (Boot Time)	Audio API (AudioContext)	Font Size (fontsize)
Device Serial Number (Serial Number)	Unique Device Identifier (UDID)	Browser Font List (fontlist)	System Information (system)
Device Hardware Information (Fingerprint)	Universally Unique Identifier (UUID)	Browser Plugin List (pluginlist)	Mini - program Platform (platform)
Advertising ID (Advertising_ID)	Mobile Phone Name (device_name)	IP and Communication Protocol	Hardware and IP Information
.....			

FIGURE 4.6 Relatively stable device parameters in common systems and environments.

- **iOS System:** As iOS is a closed, non-open-source system, it faces numerous restrictions with stricter permission controls, resulting in fewer freely accessible parameters. Relatively stable device parameters in iOS include advertising identifiers, manufacturer identifiers, network card physical addresses, system boot times, device unique identifiers, universal unique identifiers, and device names.
- **Web Browser:** Web device fingerprinting primarily relies on combining hardware information from the device running the browser with the browser's own configuration data. Relatively stable device parameters in web browsers include user agent, 2D and 3D graphics APIs, audio API, browser font list, browser plugin list, IP address, and communication protocols.
- **Mini-Programs:** Mini-programs are specialized applications running on platforms like WeChat mini-programs or Alipay mini-programs. Therefore, when constructing mini-program device fingerprints, in addition to collecting standard hardware or system parameters, it is also necessary to gather various parameters from the underlying platform. Relatively stable device parameters within mini-programs include mini-program ID, screen parameters, language, font size, system information, mini-program platform, as well as hardware and IP information.

4.3.3 Generation Algorithm

Due to significant variations in system environments across devices and the diverse nature of parameter information, generation algorithms differ for each device type. This section first introduces the algorithm for generating a universal device fingerprint, followed by detailed explanations of the generation algorithms for Android systems, iOS systems, mini-programs, and web browsers.

4.3.3.1 Universal Device Fingerprint

The generation process for a universal device fingerprint, based on the device fingerprint matching workflow, is illustrated in Figure 4.7. The algorithmic process can be summarized as follows: collect device information and construct a device profile, perform device similarity calculations to identify multiple similar devices, and finally execute precise matching and collision detection to generate a new device fingerprint or match an existing one. It should be noted that when generating a universal device fingerprint, if similar devices are detected, collision detection for similar device fingerprints must be performed in addition to the precise matching algorithm. The core approach for collision detection involves comparing the device credentials held by the client with the historical credentials stored on the server. If the current credentials match any historical credentials, it indicates a collision in the device fingerprint, necessitating the generation of a new fingerprint.

4.3.3.2 Android Device Fingerprint

To prevent black/gray industry actors from tampering with device parameters to bypass fingerprint detection and assign legitimate fingerprints to modified devices, additional checks are required after verifying the existence and validity. These checks use collected

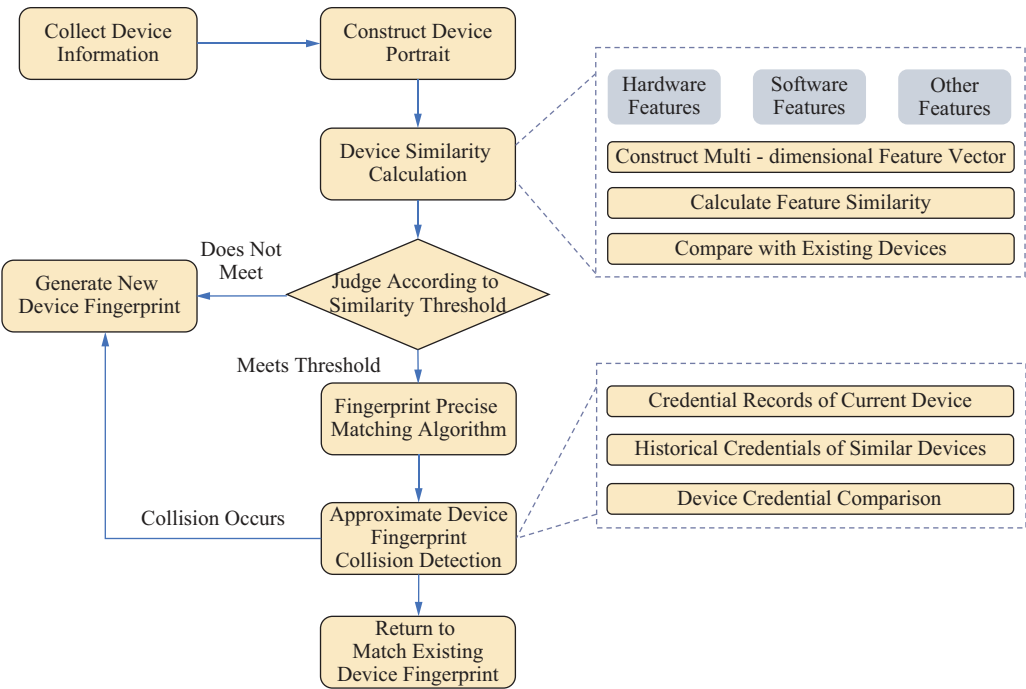


FIGURE 4.7 Universal device fingerprint generation process.

information to determine whether the device is an emulator or exhibits suspected flashing behavior. For example, to identify emulators, additional parameters like the device's CID are collected to verify the device's ROM, or features are constructed using parameters such as the Android device fingerprint.

For the collected parameters, it is necessary to distinguish their importance, generally categorizing them into primary features and auxiliary features. Primary features are typically core characteristics with high discriminative power, such as IMEI. Auxiliary features are used for strategy prioritization and combination evaluation, such as fingerprint. Feature selection should also consider the stability and uniqueness inherent to each dimension. Ultimately, multiple strategy combinations are constructed using primary and auxiliary features. This enables accurate and stable device identification when verifying whether a device matches its corresponding fingerprint. For example, if all device features match, no further comparison is needed. If most primary features match along with some auxiliary features, the device is deemed consistent; if some primary features mismatch but auxiliary features match, the device is deemed inconsistent. When multiple strategy combinations match, the highest-level strategy is selected as the judgment criterion. The Android device fingerprint generation process is illustrated in Figure 4.8.

4.3.3.3 iOS Device Fingerprint

When collecting data on iOS systems, the most readily obtainable data points are device_name, IDFA, IDFV, MAC address, and other hardware information. Depending on whether the device's MAC address can be collected, different feature combination strategy

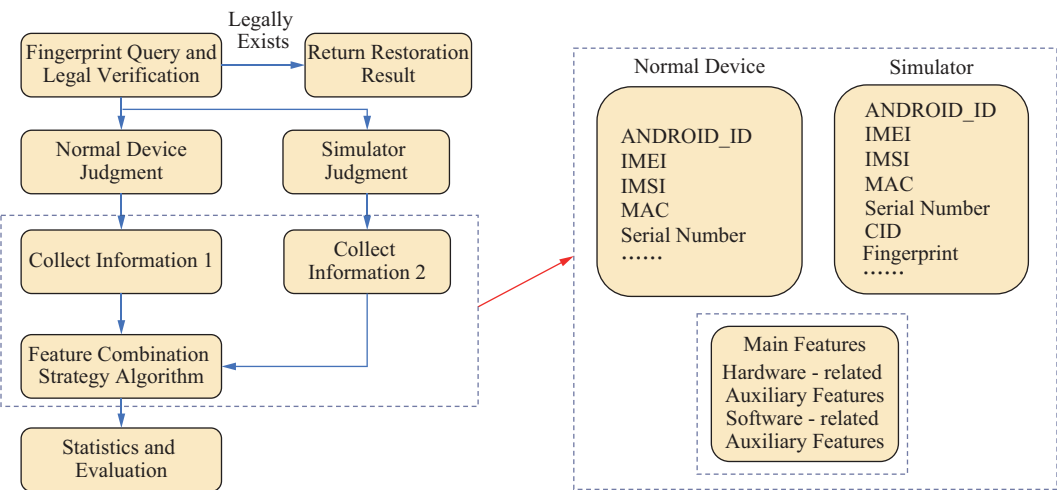


FIGURE 4.8 Android device fingerprint generation process.

algorithms can be selected. If the MAC address is available, it is combined with device_name, IDFA, or IDFV as primary features, while other hardware information serves as auxiliary features. If the MAC address cannot be collected, device_name, IDFA, and IDFV are used as primary features, along with compensatory features such as the Wi-Fi address, system version, device memory, and hard drive information. Other hardware information is then utilized as auxiliary features. Similarly, an index is generated from the features, similarity calculations are performed, approximate devices are identified, and unique devices are precisely matched based on strategy-based ranking. The iOS device fingerprint generation process is shown in Figure 4.9.

4.3.3.4 Mini-Program Device Fingerprint

In the mini-program platform scenario, beyond the standard device hardware fingerprint generation process, a mini-program user fingerprint is additionally generated. The device hardware fingerprint primarily consists of hardware information, which is less prone to change. The mini-program user fingerprint is generated by combining the mini-program's internal openid, platform version, call information, and partially de-identified user data. Consequently, factors such as software version updates, device changes, and alterations in the data collection process can impact the results.

In practical scenarios, device fingerprints can be generated using collected hardware information, while user fingerprints can be derived from mini-program user behavior. Generally, a user's unique device fingerprint is generated by combining both types of information. Individually generated device and user fingerprints can also help identify abnormal risks in fraud scenarios. For example, when user fingerprint queries match but device fingerprints differ, it may indicate a device switch during login. This can be evaluated against business contexts to assess potential login fraud risks. The mini-program device fingerprint generation process is illustrated in Figure 4.10.

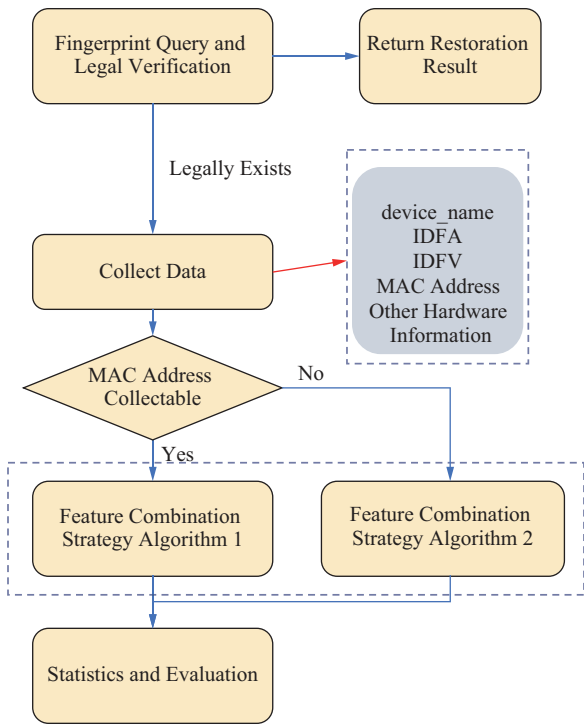


FIGURE 4.9 iOS device fingerprint generation process.

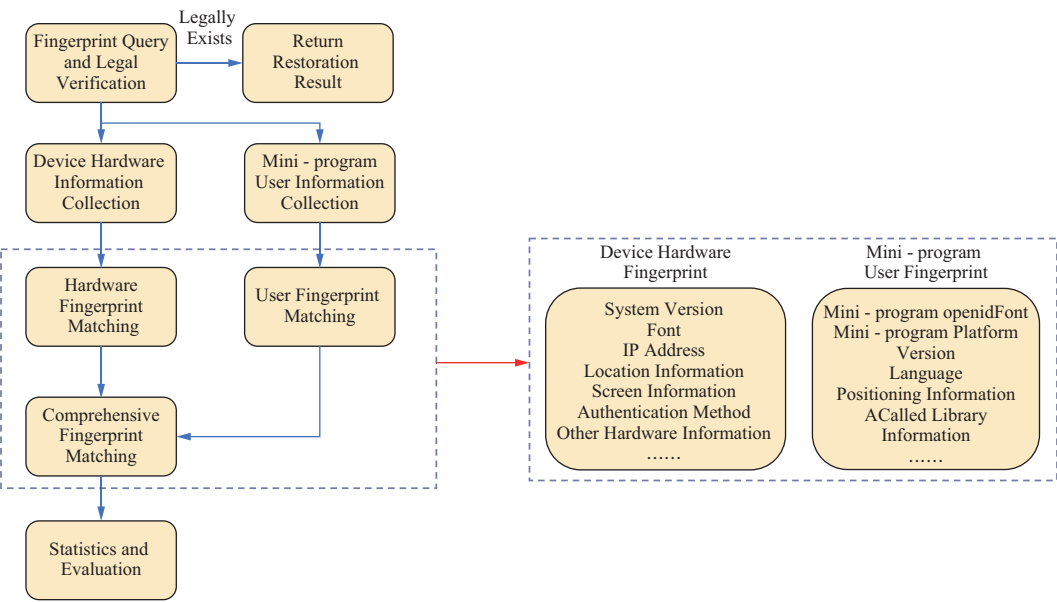


FIGURE 4.10 Mini-program device fingerprint generation process.

4.3.3.5 Web Device Fingerprint

Mini-programs essentially represent web fingerprints operating within a specific environment. Web device fingerprints are generated by combining hardware information from the device running the browser with the browser's own configuration data. They collect relatively fewer unique identifiers and exhibit poorer stability. Current web device fingerprint designs prioritize cross-platform compatibility and anomaly detection mechanisms to counteract activities carried out by the black/gray industry that exploit browsers to fabricate false device fingerprints.

Furthermore, the core process underlying the aforementioned terminal device fingerprinting techniques involves generating feature vectors, creating indices, performing similarity calculations, and matching device fingerprints. This process can fundamentally be viewed as one of classification and ranking, making machine learning methods suitable for constructing algorithmic models.

Taking device model as an example of a dimensional feature, after preprocessing the discretized data related to model parameters collected from a specific channel, feature engineering is performed to generate a multi-dimensional dense feature vector for the device model. Subsequently, dense features such as model commonality, model length, special symbol values in the model, and parameters of models frequently used in the black/gray industry are generated through analysis. Next, model training, parameter tuning, and testing are performed using clustering algorithms (such as k-means clustering or DBSCAN). This ultimately categorizes online data. For categories where devices are highly suspected of being associated with activities related to the black/gray industry, further analysis is conducted to effectively pinpoint devices with suspicious model dimensions. This allows for inferring that the fingerprints of these devices are artificially constructed.

For the identified suspicious devices, conducting multi-dimensional analysis serves dual purposes: it enhances operational personnel's understanding of how black/gray industry cybercriminals manipulate device parameters to evade detection, and it provides insights from these suspicious devices to collect additional feature dimensions. This approach offers valuable optimization strategies for refining device fingerprint generation algorithms.

4.4 SUMMARY

This chapter primarily introduced device fingerprinting technology, one of the core foundational techniques for traffic fraud prevention. Device fingerprinting effectively establishes device uniqueness, laying a solid foundation for subsequent countermeasures against the black/gray industry through business rules and machine learning solutions. Additionally, it is crucial to ensure that device fingerprint generation does not infringe upon user privacy, balancing security, reliability, performance, and user experience.

CAPTCHA Verification

Chapter 2 outlined the black/gray industry's operations across three phases: platform promotion during the pre-traffic phase, platform usage during the mid-traffic phase, and resource monetization during the post-traffic phase. Their fraudulent tactics create an illusion of traffic growth, making it difficult for business platforms to distinguish normal traffic from fake traffic. For business platforms, normal traffic holds true value, while fake traffic severely impacts platform revenue and user experience. For example, when a ticketing platform releases tickets, they are instantly snapped up by automated scripts or bots operated by members of the black/gray industry. This forces legitimate users to resort to scalpers, significantly degrading their product experience and imposing additional financial costs. Chapter 4 covered device fingerprinting technology, which is commonly used to establish unique user identifiers by collecting device information and behavioral attributes to feed data to servers.

This chapter focuses on CAPTCHA verification, which serves as the first line of defense against traffic fraud from the black/gray industry. CAPTCHA verification extends beyond superficial CAPTCHA checks, integrating comprehensive decision-making based on user trajectories, device fingerprint IDs, device data, and user environment information.

5.1 CAPTCHA VERIFICATION FUNDAMENTALS

This section introduces fundamental knowledge about CAPTCHA, covering its origins, application scenarios, and implementation frameworks. This lays the groundwork for subsequent discussions on CAPTCHA attacks and defenses at both the foundational and design levels.

5.1.1 The Birth of CAPTCHAs

At the turn of the 21st century, Yahoo! faced an unprecedented challenge as its email service was inundated with automated spam generated by scripts from the black/gray industry, severely degrading user experience. To tackle this vexing problem, Yahoo invited Dr. Luis von Ahn and others to design a spam detection solution. The core objective was to distinguish between human and machine users on the internet—a concept inversely aligned

with the Turing test. Building upon this inverse Turing test approach, Dr. von Ahn and his team ultimately developed a verification program capable of differentiating humans from machines: Yahoo’s first-generation CAPTCHA.

Yahoo’s original CAPTCHA randomly generated distorted characters for users to identify. Only by correctly inputting the recognized characters could users pass verification. Given the computational capabilities of the era, computers struggled to recognize these distorted characters, whereas humans could accomplish this effortlessly. This approach effectively blocked automated attacks from bots or scripts operated by the black/gray industry.

5.1.2 Application Scenarios of CAPTCHAs

CAPTCHAs serve four primary functions in blocking fraudulent traffic from the black/gray industry: account security, data security, operational security, and transaction security, as illustrated in Figure 5.1.

These four CAPTCHA application scenarios can be further subdivided into the following specific scenarios.

1. Account Security

- Malicious Login: Preventing brute-force attacks by the black/gray industry to ensure login security.
- Fake Registration: Preventing large-scale invalid registrations generated by registration bots, raising the barrier for registration fraud.
- Credential Stuffing Attacks: Preventing brute-force credential stuffing attacks by the black/gray industry to safeguard account security.

2. Data Security

- Malicious Scraping: Prevent illegal bulk data scraping by the black/gray industry using proxy IPs and other methods, safeguarding internet data security.

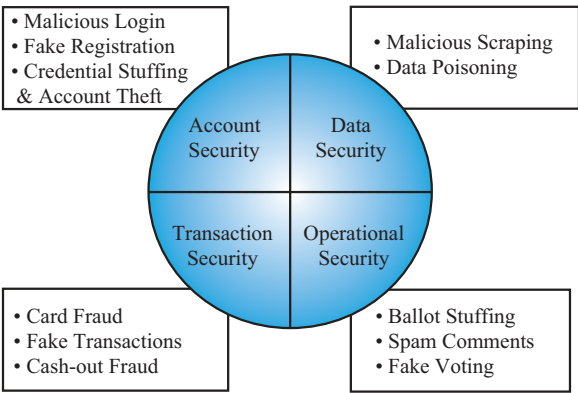


FIGURE 5.1 Application scenarios of CAPTCHAs.

- **Data Contamination:** Prevent the malicious “poisoning” of data by the black/gray industry that compromises the accuracy of risk control models on business platforms.

3. Operational Security

- **Ballot Stuffing:** Prevent ticket scalpers from bulk vote stuffing to protect the legitimate rights of regular users on ticketing platforms.
- **Spam Comments:** Prevent malicious flooding or abusive comments on forums or blogs to maintain a clean online environment.
- **Fake Voting:** Prevent invalid or fraudulent votes to ensure the fairness and validity of events or rankings.

4. Transaction Security

- **Card Fraud:** Prevent unauthorized credit card transactions to safeguard users’ financial assets.
- **Fake Transactions:** Prevent e-commerce platforms from inflating sales figures and reviews through fraudulent transactions, thereby maintaining fair competition and protecting the legitimate rights of ordinary consumers.
- **Cash-out Fraud:** Prevent the black/gray industry from conducting bulk cash-out schemes, safeguarding the interests of banks and other institutions.

5.1.3 Framework for Building CAPTCHAs

The core process of constructing CAPTCHAs involves generating and verifying challenge values. However, the entity responsible for generating and verifying these challenge values can be implemented in different ways, as shown in Table 5.1. Challenge values can be generated either on the client side or the server side, and the verification process can also occur on either side.

The implementation details of CAPTCHAs may vary slightly depending on whether the generation or verification of the challenge value occurs on the client or server side, but the overall framework remains similar. This section uses the example where both the generation and verification of the challenge value occur on the server. The process of constructing a CAPTCHA is illustrated in Figure 5.2. First, the user initiates a CAPTCHA request through the client (e.g., app, browser, or mini-program). The server responds by creating a new SessionID for this request and generating a random CAPTCHA based on a defined

TABLE 5.1 Endpoint Selection for Generating and Validating Verification Values

Core Component	End-Side Selection Method
Generating verification values	Client/server
Verify checksum	Client/server

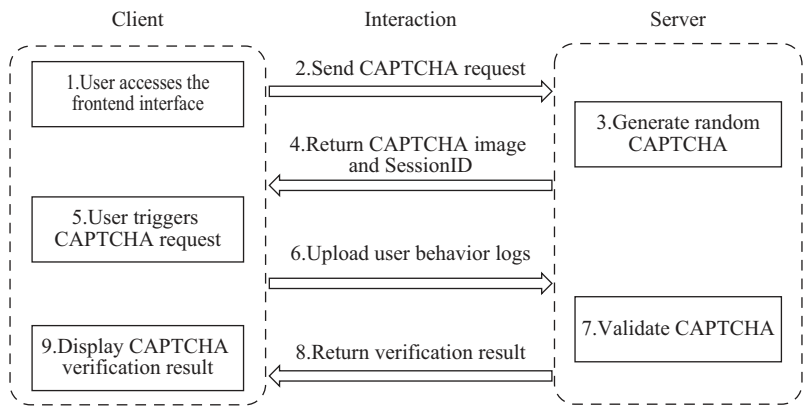


FIGURE 5.2 CAPTCHA generation process.

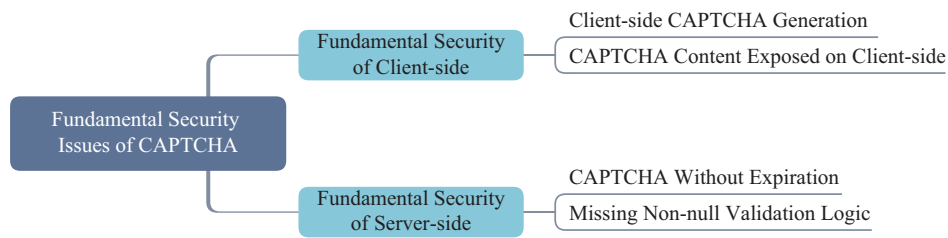


FIGURE 5.3 Foundational CAPTCHA security issues.

strategy. Next, the server returns the generated CAPTCHA along with the SessionID to the client. Upon receiving the CAPTCHA, the client displays it to the user. The user performs an action (e.g., clicking) on the displayed CAPTCHA. The client then uploads information about the user’s interaction (e.g., user behavior trace) to the server for verification. After verification, the server returns the CAPTCHA verification result to the client for display and simultaneously terminates the current session. Finally, the user sees whether the verification result is successful or failed.

5.2 FOUNDATIONAL SECURITY MEASURES

To successfully defend against bot attacks from the internet black/gray industry using CAPTCHAs, the foundational security of the CAPTCHA construction process must first be ensured. Examining each stage of the CAPTCHA construction process, the fundamental security issues requiring attention can be categorized into two main types, as shown in Figure 5.3.

1. Client-Side Foundational Security

- CAPTCHA Generation on the Client Side: CAPTCHAs generated via client-side JavaScript and validated solely through JavaScript on the client side carry the risk of being bypassed by the black/gray industry through packet sniffing or by disabling JavaScript.

- CAPTCHA Content Output on Client Side: CAPTCHA content stored in client-side cookies or response packets poses risks of reverse engineering and attacks by members of the black/gray industry.

2. Server-Side Fundamental Security

- CAPTCHA Does Not Expire: After a CAPTCHA verification completes, if the SessionID is not promptly destroyed and the session window closed, the system defaults to maintaining it for a period. This creates a risk of CAPTCHAs being reused by the black/gray industry. Ensure CAPTCHAs expire immediately after use to mitigate attack risks.
- Missing Non-Empty Validation Logic: During CAPTCHA verification, certain circumstances (e.g., deleting specific values from cookies or request parameters) may result in an empty CAPTCHA. This scenario poses a risk of exploitation by the black/gray industry.

To prevent bypassing risks, securing the CAPTCHA foundation is only the first step. In addition, it is necessary to address attacks from black and gray markets at the CAPTCHA design level by iteratively updating the CAPTCHA design schemes in a timely manner for prevention. The following sections detail CAPTCHA design-level defense strategies.

5.3 DESIGN-LEVEL OFFENSE AND DEFENSE

Since its inception, CAPTCHA design has revolved around a core principle: leveraging unsolved challenges in contemporary artificial intelligence to enhance resistance against cracking by the black/gray industry. However, as AI technology advances, so too do the techniques employed by the black/gray industry to bypass CAPTCHAs, necessitating continuous upgrades in CAPTCHA design. This has ultimately led to a protracted battle of offense and defense. The evolution of CAPTCHA design strategies is illustrated in Figure 5.4, showing that business access traffic has progressed through three stages: text-based CAPTCHAs, behavioral CAPTCHAs, and next-generation CAPTCHAs.

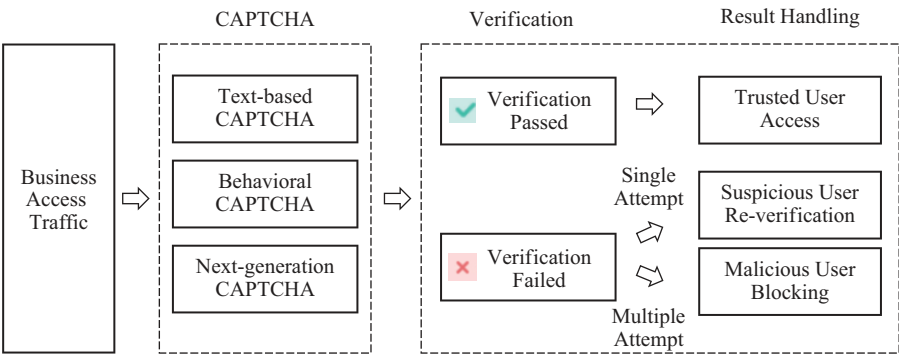


FIGURE 5.4 Evolutionary approach to CAPTCHA design offense and defense.

5.3.1 Text-Based CAPTCHA

The original Yahoo text-based CAPTCHA introduced in Section 5.1.1 represents the earliest form of text-based CAPTCHA. In these initial designs, character elements consisted solely of numbers and letters. While classifying the entire string yielded hundreds of thousands of possible combinations, offering some resistance to brute-force attacks by the black/gray industry, considering individual character combinations reveals only 36 possibilities. If strings are categorized by individual characters, the number of categories decreases significantly. This drastically reduces the difficulty of cracking text-based CAPTCHAs, enabling the black/gray industry to use machines for automated bulk recognition. Consequently, large-scale automated CAPTCHA-solving platforms emerged. The working principle of these automated platforms is illustrated in Figure 5.5.

As shown in the figure above, when black/gray industry operators (such as “freeloaders” or “scalpers”) make bulk service requests, they can use cheating software to obtain CAPTCHAs from client applications (e.g., websites, apps, or mini-programs). These CAPTCHAs are then automatically recognized and returned via the CAPTCHA-solving platform, allowing them to easily bypass text-based CAPTCHA defenses. The primary advantage of automated CAPTCHA solving is its high efficiency.

In response to these cracking methods employed by operators of the black/gray industry, text-based CAPTCHAs have implemented corresponding countermeasures, further upgrading their design schemes, as illustrated in Figure 5.6.

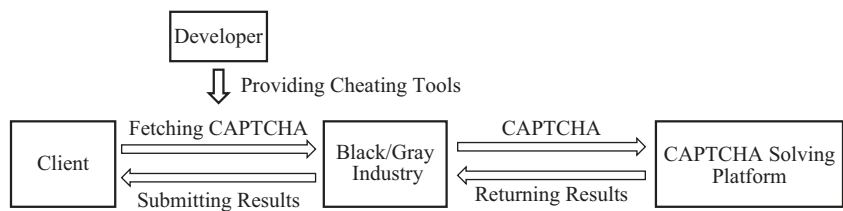


FIGURE 5.5 Working principle of automated CAPTCHA-solving platforms.

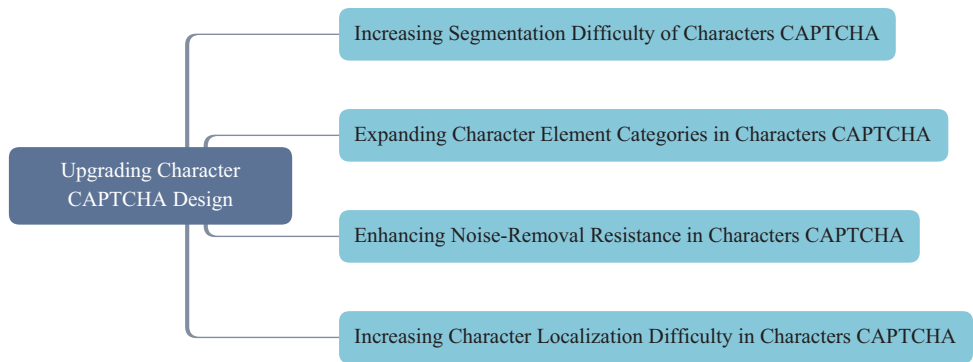


FIGURE 5.6 Enhanced text-based CAPTCHA design.

- Increasing the difficulty of splitting text-based CAPTCHAs into individual characters: This is primarily achieved by reducing character spacing, increasing character adhesion, or even partially overlapping characters.
- Increasing the number of character element categories in text-based CAPTCHAs.
- Increasing the difficulty of noise removal in text-based CAPTCHAs.
- Increasing the difficulty of character positioning in text-based CAPTCHAs: Primarily achieved through dynamic changes in character placement.

For simple text-based CAPTCHAs, AI-based machines can achieve high recognition accuracy. However, for text-based CAPTCHAs with added interference or noise, machine recognition accuracy drops significantly. Human recognition, however, remains extremely high. This has led to the emergence of crowdsourcing CAPTCHA-solving platforms. These platforms primarily organize humans to identify CAPTCHAs, thereby bypassing CAPTCHA defenses. The working principle of a crowdsourcing CAPTCHA-solving platform is shown in Figure 5.7.

As shown in the figure above, the CAPTCHA-solving platform packages CAPTCHAs received from upstream sources into tasks, which are then posted on crowdsourcing platforms. Subsequently, “traffic workers” claim CAPTCHA-solving tasks from these platforms and manually return verified results to the upstream source for validation. Crowdsourcing platforms primarily provide traffic-driving information to CAPTCHA-solving platforms. They serve as the most critical channel for CAPTCHA-solving platforms to reach users, forming a collaborative relationship with them.

Examining the entire upstream–downstream ecosystem of crowdsourced CAPTCHA-solving platforms reveals a distinct black/gray industry profit chain. Downstream part-time “traffic workers” earn monetary rewards by completing CAPTCHA tasks distributed by crowdsourcing platforms, while these platforms receive profit shares for directing traffic to CAPTCHA-solving platforms. Midstream CAPTCHA-solving platforms provide systematic CAPTCHA-solving services to upstream black/gray industry operators, earning commissions in return. Finally, upstream members of the black/gray industry profit from business clients through bulk registrations and “scraping deals.” It is precisely this profit chain that enables the existence of crowdsourcing CAPTCHA-solving platforms.

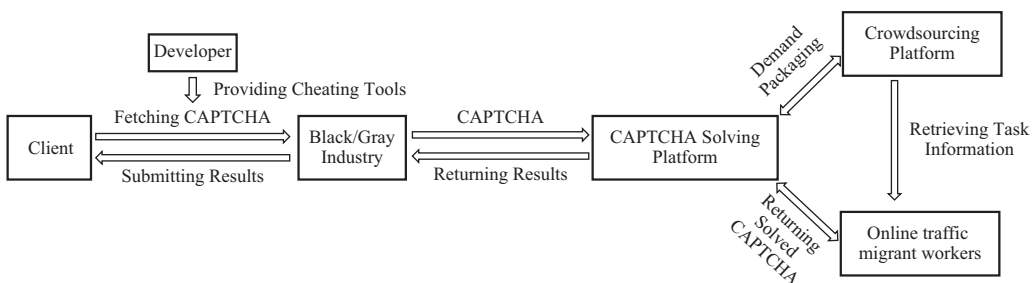


FIGURE 5.7 Working principle of crowdsourcing CAPTCHA-solving platforms.

Compared to automated CAPTCHA-solving platforms, crowdsourcing platforms offer an advantage in terms of higher accuracy but a disadvantage in terms of lower efficiency, falling short of the speed of automated systems. Considering overall cost and efficiency, members of the black/gray industry typically prioritize automated solutions in most scenarios.

The aforementioned defense strategies based on the evolution of text-based CAPTCHA designs primarily operate in a piecemeal fashion—patching vulnerabilities as they arise. This forces businesses into a reactive cycle of upgrading their defense systems. However, from a systemic defense perspective, no matter how businesses upgrade their text-based CAPTCHA designs, the black/gray industry can crack them through repeated trial-and-error attempts. Breaking text-based CAPTCHAs is merely a matter of time. Based on past practical countermeasure experiences, a certain cycle is required from the emergence of a new CAPTCHA design to its successful cracking by the black/gray industry, followed by the integration of automated tools into the black market. If businesses can update text-based CAPTCHA solutions faster than the black/gray industry can crack them, they can shift from a reactive to a proactive stance, effectively reducing the success rate of the black/gray industry in cracking text-based CAPTCHAs. Therefore, dynamic text-based CAPTCHAs are gradually becoming the new trend in this era.

5.3.2 Behavioral CAPTCHA

After prolonged competition between traditional text-based CAPTCHAs and the black/gray industry, artificial intelligence technologies like optical character recognition and deep learning have been mastered and deployed by the black/gray industry to counter text-based CAPTCHA solutions. Cracking methods based on optical character recognition and similar technologies have reached unprecedented levels of accuracy and efficiency, posing significant challenges to security defenses that rely on traditional text-based CAPTCHAs.

As the black/gray industry continuously innovates its CAPTCHA-cracking techniques, CAPTCHA design urgently requires breaking free from traditional text-based CAPTCHA approaches. This has led to the emergence of a new generation of CAPTCHAs based on behavioral verification—behavioral CAPTCHAs. Behavioral CAPTCHAs primarily utilize images as their content medium, eliminating the reliance on characters inherent in traditional text-based CAPTCHAs and opening up greater design possibilities. Among these, a typical example is the slider CAPTCHA.

The fundamental verification principle and process of slider CAPTCHA comprise the following three components:

- **Server Generates the Slider CAPTCHA:** Based on a cutout template, the server randomly generates sliders of various shapes and background images with shadows from the original image while storing the corresponding coordinates for each slider position.
- **Client Collects User-Related Metrics:** When the user moves the slider on the client side, the client captures data such as the mouse movement trajectory and speed, slider coordinates, and retry count, then sends this information back to the server.

- **Server Validates CAPTCHA Results:** The server calculates the error between the final slider coordinates and the target coordinates. It then combines this with other dimensions like the sliding trajectory and retry count, using a pre-trained machine learning model to comprehensively determine the result.

Compared to traditional text-based CAPTCHAs, slider CAPTCHAs integrate multi-dimensional assessments of user behavior for enhanced security. Simultaneously, users need not type characters—a simple mouse swipe suffices for rapid verification, creating a more user-friendly interaction. Consequently, slider CAPTCHAs are widely adopted.

Despite their high security, ongoing research by the black/gray industry has enabled the bypassing of CAPTCHAs through object detection techniques and simulation of human-like swipe trajectories. The process of black/gray industry countermeasures against CAPTCHAs is illustrated in Figure 5.8.

- **Acquiring CAPTCHA-Related Images:** The black/gray industry extracts the slider image and gap background image from client resource files.
- **Identifying Slider and Gap Coordinates:** Since images extracted from client resources often differ in size from those displayed on the client, the black/gray industry first resizes the images to match before identifying the coordinates of the slider and gap.
- **Calculating the Distance Between the Slider and Gap:** The black/gray industry uses the coordinates of the slider and gap to calculate the distance between them. This distance indicates how far the slider needs to be moved.
- **Simulating Human-Like Sliding Behavior for Matching:** Based on the calculated distance, the black/gray industry simulates human-like sliding behavior to match the slider. For example, they introduce speed fluctuations during sliding to avoid constant uniform motion or add slight errors when aligning the slider with the notch to mimic genuine human behavior.
- **As sliding CAPTCHAs were compromised by the black/gray industry, businesses were forced to upgrade and iterate them, leading to the emergence of click-to-select image CAPTCHAs.** Users must follow the text prompts on the CAPTCHA, clicking the characters in the image sequentially to pass verification. Compared to sliding CAPTCHAs, click-to-select CAPTCHAs incorporate text-based click order, enhancing security to some extent but offering a relatively less user-friendly experience.

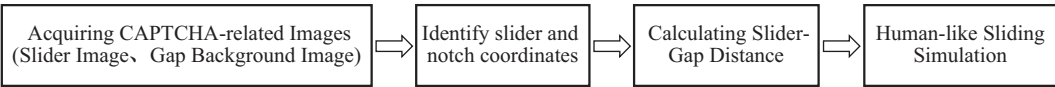


FIGURE 5.8 Process of the black/gray industry circumventing slider CAPTCHAs.

In certain scenarios, click-to-select CAPTCHAs incorporate semantic understanding that machines struggle to replicate. Users must correctly interpret the meaning of the displayed text and click characters in the specified order to pass verification. While this design significantly enhances security, it offers a poor user experience.

At this point, the battle over CAPTCHA design has moved beyond the deadlock of text-based CAPTCHAs, opening up a new “battlefield” and ushering in the era of behavioral CAPTCHAs. To intercept traffic from the black/gray industry, some businesses make trade-offs between security and user experience. They enhance security by introducing highly complex behavioral CAPTCHAs, but this comes at the cost of user experience. Therefore, mainstream platforms predominantly use slider CAPTCHAs. While ensuring user experience, they collect more behavioral data and device fingerprint data to train the server’s machine learning anomaly detection models, thereby identifying anomalous traffic.

5.3.3 Next-Generation CAPTCHA

To strike a balance between user experience and security while addressing nuanced requirements across different scenarios, next-generation CAPTCHAs are evolving in two primary directions: heightened security and user-friendly interaction. Enhanced security focuses on innovations centered around “tasks easily accomplished by humans but challenging for machines,” while user-friendly interaction emphasizes simplicity and engaging design. Widely adopted next-generation CAPTCHAs include intelligent reasoning CAPTCHAs and seamless CAPTCHAs.

Intelligent reasoning CAPTCHAs are primarily designed around two human strengths that machines currently struggle to match: logical reasoning and spatial imagination. Their format is illustrated in Figure 5.17. Users simply follow the textual prompts on the CAPTCHA, applying logical reasoning combined with multi-dimensional spatial imagination to identify the correct elements in the image and then click them. The advantage of this next-generation CAPTCHA type is its high security, while the disadvantage is that it requires a certain level of comprehension ability, making user interaction relatively less user-friendly. Intelligent reasoning CAPTCHAs are commonly used in high-security scenarios such as account unfreezing and password recovery to mitigate the risks of internet traffic fraud.

Seamless CAPTCHA is a next-generation CAPTCHA that employs intelligent human-machine recognition by comprehensively analyzing multi-dimensional information, including user behavior, environmental data, and device fingerprints. Its design principle is illustrated in Figure 5.9.

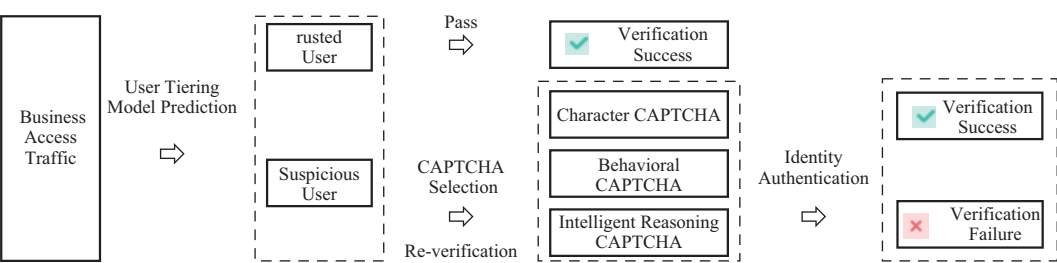


FIGURE 5.9 Design principle of seamless CAPTCHA.

First, for business access traffic, a user classification model assesses user risk levels, categorizing users as either trusted or suspicious. This classification model primarily integrates information from the following dimensions:

- User Environment Information: Access IP, browser user-agent, cookies, operating system details, etc.
- User Behavior Information: Access frequency, button click duration, keyboard typing speed, mouse movement patterns, etc.
- Additional user-related information.

Second, different handling decisions are made for users at varying risk levels. Trusted users can be granted seamless, rapid access to minimize disruption. For suspicious users, CAPTCHAs of varying difficulty levels automatically pop up for secondary verification based on risk severity—such as behavioral CAPTCHAs or intelligent inference CAPTCHAs.

Finally, to enhance security, seamless CAPTCHAs employ image rotation and mutation during storage to prevent brute-force attacks from the black/gray industry. During transmission, images are randomly segmented and transmitted out of order. Additionally, anomaly detection strategies like device and IP correlation analysis guard against bulk attacks from the black/gray industry. Seamless CAPTCHAs offer high security and an excellent user experience.

5.4 SUMMARY

This chapter systematically explores the evolution of attacks and defenses against traffic fraud, focusing on the first line of defense. It begins by introducing foundational knowledge of CAPTCHA verification—covering the origin, application scenarios, and architectural framework of CAPTCHAs—to provide readers with a basic understanding. Subsequently, it discusses the foundational attack–defense dynamics of CAPTCHA verification from both client-side and server-side perspectives within the implementation process. Finally, it traces the evolution of attack–defense strategies from text-based CAPTCHAs and behavioral CAPTCHAs to next-generation CAPTCHAs, thereby establishing a systematic understanding of CAPTCHA verification for readers.

Rules Engine

As the first line of defense for traffic security, CAPTCHA verification primarily intercepts traffic fraud before it enters the platform. For traffic fraud already present within the platform, further countermeasures rely on the rules engine. This engine consists of two main components: risk lists and expert rules, with the latter divided into generic rules and business-customized rules. The operational principle of the rules engine is illustrated in Figure 6.1.

6.1 RISK LIST

In traffic fraud scenarios, black/gray industry operators exploit resources like proxy IPs, SIM bank devices, and group control devices. However, these resources require significant investment, with higher-quality resources commanding greater costs. To reduce expenses, black/gray industry operators repeatedly reuse their existing resources to commit fraud across multiple business platforms until these resources become ineffective on most platforms. Therefore, businesses can accumulate or integrate risk data on black/gray industry resources from multiple sources, compiling a risk list for business risk prevention and control.

6.1.1 Risk List Fundamentals

Risk lists can be categorized into three types based on risk levels, as detailed below:

- **Blacklist:** Refers to users or resources that flagrantly violate platform rules and pose high risks to business operations. Blacklists can be directly applied to block high-risk users, preventing recurring fraud incidents. They are constructed based on elements or resources involved in traffic fraud, encompassing high-risk accounts, high-risk mobile phone numbers, second-dial IPs, and group control devices.
- **Whitelist:** Refers to a collection of users or resources that have a business partnership or require special protection, enabling direct access for high-confidence users. The scope of whitelists mainly includes high-quality accounts, partner mobile phone

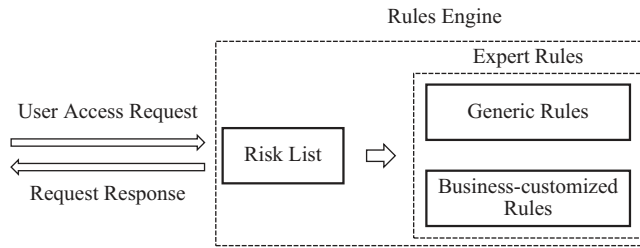


FIGURE 6.1 Working principle of the rules engine.

numbers, base station IPs, and enterprise IPs. For example, a conference product typically designates accounts with verified corporate identities as high-quality users and includes them in the whitelist for enhanced protection.

- **Gray List:** Refers to a collection of users or resources posing certain business risks. Entities on this list are primarily monitored for suspicious activity. The scope of the gray list includes suspicious accounts, questionable proxy IPs, modified devices, and dubious mobile phone numbers from virtual operators. In traffic fraud scenarios, most members of the black/gray industry disguise themselves through device modification to commit fraud. However, ordinary users may also modify devices, so device modification cannot be directly classified as malicious behavior. Instead, it should be treated as a suspected risk and placed on the gray list for monitoring.

Depending on specific business scenarios, the focus of risk lists may vary slightly. The scope outlined above is for reference only and requires adjustment based on actual business conditions. For instance, users engaged in multiple simultaneous loans fall under the gray list in financial traffic fraud scenarios but not in marketing promotion traffic fraud scenarios.

To prevent the same black/gray industry resources from repeatedly committing fraud across multiple scenarios, risk lists are typically constructed by aggregating data from multiple sources, primarily involving the following three aspects.

- **Business-Owned Data Sources:** Primarily based on proprietary business data, accumulated and collected according to operational risk prevention needs. This constitutes the primary source of list data.
- **Cross-Platform and Cross-Business Sources:** Primarily established through collaborations with relevant third-party business platforms or by jointly creating industry risk lists to enhance the security protection capabilities of the entire industry.
- **Public List Sources:** Primarily include blacklists from public security, procuratorate, and judicial authorities, as well as anti-money laundering lists, based on specific business risk prevention and control requirements.

6.1.2 Evolution of Risk Lists in Defense and Offense

Risk list design is tied to business scenarios, operational standards, and security governance objectives. To enhance prevention effectiveness, risk lists must evolve in response to the different stages of a business platform’s lifecycle.

- Risk List Design in the Early Business Phase

During the initial launch phase, the platform is in a cold-start stage without sufficient accumulated business data to build a comprehensive risk control system. At this stage, the platform often collects or introduces third-party list data for rapid deployment and risk prevention. Taking the blacklist for traffic fraud scenarios in the early business phase as an example, the reference scope of the blacklist in this scenario is shown in Table 6.1, illustrating the risk list design in the early business phase.

The primary goal during the initial business phase is to use basic risk lists to intercept obvious activities related to black/gray industry fraud, helping the business operate healthily in its early stages. However, risk lists in the initial phase have a disadvantage in terms of precision. When restrictions are too strict, the coverage rate for detecting malicious activities is relatively low.

- Risk List Design for the Mature Business Phase

After encountering black/gray industry activities during the initial phase, these entities employ various methods to circumvent risk control rules, rendering existing risk lists ineffective. During the mature phase, businesses can leverage accumulated historical data to build specialized risk lists tailored to their operations. The reference scope for blacklists in traffic fraud scenarios during the mature phase is shown in Table 6.2.

Compared to blacklists established during the early stages of business operations, those developed during the mature phase incorporate additional proprietary black data accumulated through operational experience. This further supplements and refines the list data from a business perspective, enhancing the risk prevention capabilities of the business entity. Similarly, the construction and usage logic for whitelists and gray lists parallels that of blacklists. Specific lists can be flexibly adjusted based on the business’s standards and objectives.

TABLE 6.1 Reference Scope for Blacklists in Traffic Fraud Scenarios During Business Initialization

Blacklist Dimensions	Blacklist Reference Scope
Individual list	Public security wanted lists, court-listed dishonest debtors, central bank anti-money laundering lists, etc.
Corporate list	Lists of dishonest enterprises, anti-money laundering sanctioned entities, etc.
Geographic lists	High-risk regions for anti-money laundering as published within the industry, etc.
Common fraud resources	Mobile phone numbers (fraudulent intermediary numbers, SMS verification platform black numbers, virtual carrier black numbers, etc.) IP addresses (malicious proxy IPs, second-dial IPs, etc.) Devices (fake device IDs, emulators, group control devices, etc.)

TABLE 6.2 Reference Scope for Blacklists in Financial Traffic Fraud Scenarios During the Mature Business Phase

Blacklist Dimensions	Blacklist Reference Scope
Individual list	Public security wanted lists, court-listed dishonest debtors, central bank anti-money laundering lists, etc.; business-specific lists of individuals with historical delinquencies, individuals with multiple borrowing risks, etc.
Corporate list	Lists of dishonest enterprises, anti-money laundering sanctioned entities, etc., lists of enterprises with historical business delinquencies, lists of enterprises with multiple borrowing risks, etc.
Geographic lists	High-risk regions for anti-money laundering as published within the industry, regions with concentrated risks specific to one's own business operations, etc.
Common fraud resources	Mobile phone numbers (historically delinquent numbers, fraudulent intermediary numbers, SMS verification platform black numbers, virtual carrier black numbers, etc.) IP addresses (historically blacklisted IPs, malicious proxy IPs, second-dial IPs, etc.) Devices (historically accumulated blacklisted devices, fake device IDs, emulators, group control devices, etc.)

6.1.3 Risk List Deployment and Operations

After completing the design of the risk list, the next step is to deploy and launch it. During deployment, key considerations include the list's update cycle, storage method, and obsolescence mechanism.

1. Update Cycle

The risk list incorporates both internally accumulated data and third-party list data. Due to differing data sources, update cycles may vary. Lists built from proprietary data have controllable update cycles, typically refreshed daily. For third-party list data, update cycles generally align with the characteristics of the source data. For instance, credit blacklist data requires adherence to the update standards set by the third-party provider.

2. Storage Method

The storage method for list data primarily depends on the application scenario and the characteristics of the list. List data applications mainly exhibit the following three key characteristics:

- **High-Frequency Usage:** Risk list screening is a crucial step in upfront risk assessment. Since every user access requires passing through risk list screening, these lists are frequently utilized.
- **Relatively Static:** List data typically has a long validity period with minimal changes over short timeframes.
- **Small Scale:** List data volumes are generally modest, resulting in low storage overhead.

Given these characteristics, list data is typically stored directly in memory or high-performance key-value systems to enable rapid matching and identification of traffic fraud from black/gray industries.

3. Elimination Mechanism

After deploying and launching the list data, closed-loop monitoring of application effectiveness is necessary to prevent false penalties caused by outdated lists. Online performance can be monitored through real-time metrics such as external customer complaint ratios and real-time penalty ratios.

Taking mobile phone numbers as an example, mobile phone numbers are a common fraud resource for the black/gray industry. To minimize costs, these actors typically control a batch of blacklisted mobile phone numbers across multiple platforms for sustained malicious activity until most platforms block them due to risk controls. Telecom operators often recycle these long-dormant mobile phone numbers after abandonment, reissuing them to the market. New users acquiring these mobile phone numbers may be legitimate. If business-side blacklists fail to promptly update and remove these mobile phone numbers abandoned by the black/gray industry, the risk of false positives significantly increases. Therefore, establishing an effective risk list elimination mechanism is essential.

Online risklist elimination mechanisms can generally be considered from two perspectives:

- **Proactive Removal Mechanism:** Set fixed validity periods for risk lists to automatically eliminate expired entries. Examples include daily updates where the N most recent days' data replaces the previous day's entries, or using user profiling to remove matches where user activity appears low risk. Alternatively, synchronize third-party data (e.g., carrier-reported recycled mobile phone numbers) to eliminate historical blacklisted mobile phone numbers.
- **Passive Elimination Mechanism:** Primarily based on real-time metrics like external network complaint ratios triggered by matches on the risk list. Once an alert threshold is reached, corresponding list entries are marked as invalid to prevent large-scale misjudgments in external interactions.

Depending on specific business requirements, both elimination mechanisms can be applied in combination.

6.2 GENERIC RULES

For traffic fraud scenarios, risk lists primarily intercept black/gray industry resources such as accounts, IPs, and devices with prior fraudulent activity. New or unlisted black/gray industry resources require expert rules for further control. Expert rules are derived from historical countermeasure experience and data analysis, offering simplicity and strong interpretability.

Expert rules can be categorized into generic rules and business-customized rules. Generic rules are built around essential black/gray industry resources used in fraud operations, such as IP addresses, devices, and accounts. Business-customized rules are tailored to the unique characteristics of each operation. Next, we will explain generic rules from three dimensions: IP, device, and account. The countermeasure approach for generic rules is illustrated in Figure 6.2.

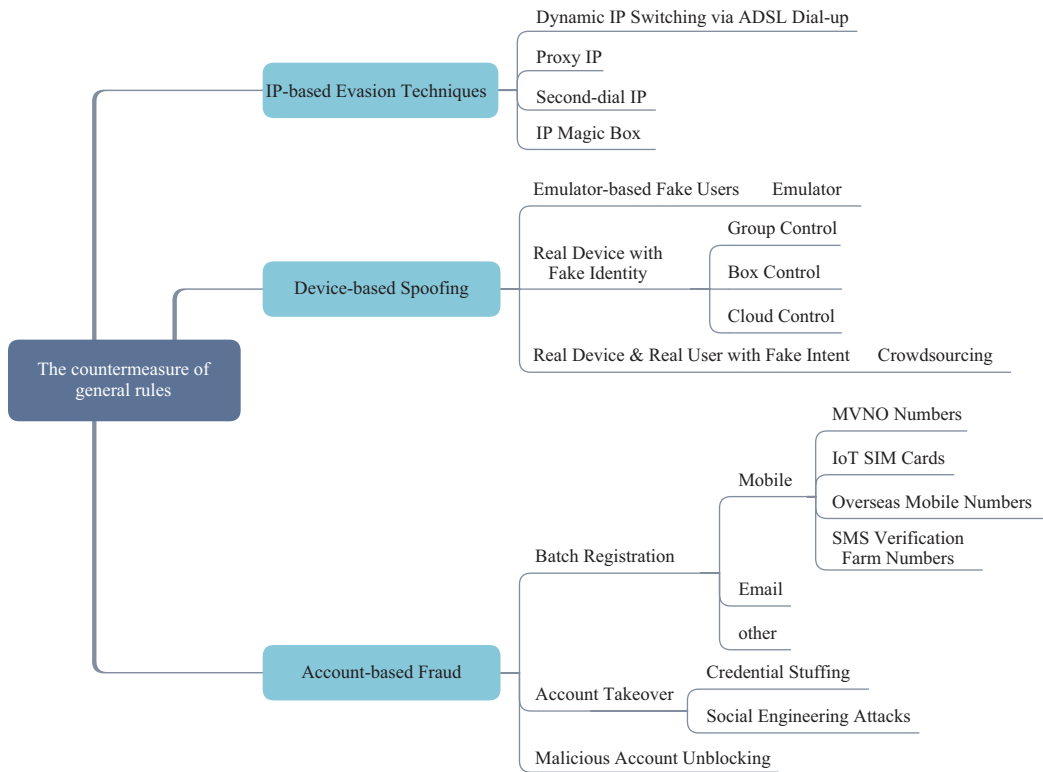


FIGURE 6.2 Adversarial approach for generic rules.

6.2.1 Evolution of IP

As the fundamental identifier for internet traffic access, IP addresses are essential resources for activities in the black/gray industry involving bulk registration, “scraping deals,” chart manipulation, and credential stuffing. Figure 6.3 illustrates the application scenarios of IP resources in black/gray industry operations.

Given the finite nature of IP resources—particularly the scarcity of IPv4 addresses—and their critical role in black/gray industry traffic fraud, IP resources have long been one of the most fiercely contested battlegrounds between legitimate businesses and members of the black/gray industry.

To prevent black/gray industry operators from repeatedly committing traffic fraud using IP resources, businesses often impose restrictions at the IP level. For example, they may limit the number of access attempts from a single IP address to no more than N times within a specified period, thereby reducing the risk of repeated attacks by these operators. In response to these IP-level restrictions, the black/gray industry continuously evolves its IP-based fraud tactics. The progression of these tactics is illustrated in Figure 6.4. Simultaneously, businesses continually upgrade their IP rule strategies to block these attempts. The following sections detail the evolving offensive and defensive strategies between both parties.

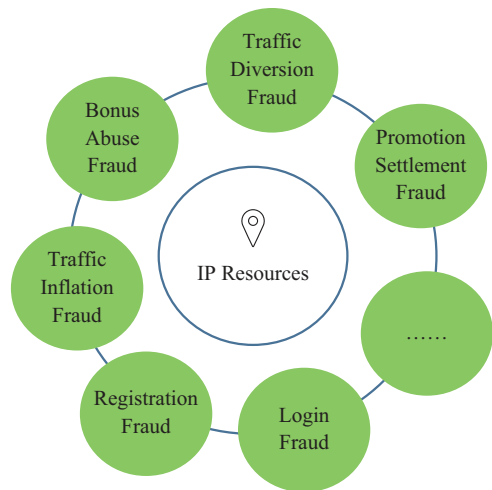


FIGURE 6.3 Application scenarios of IP resources in the black/gray industry.



FIGURE 6.4 Evolution of IP fraud tactics by the black/gray industry.

1. Phase 1: Countermeasures Against ADSL Dial-Up Fraud

The black/gray industry circumvents service providers’ IP frequency control strategies by switching IP addresses. Initially, this was primarily achieved through ADSL dial-up switching. Leveraging the Point-to-Point Protocol over Ethernet principle, they repeatedly disconnected and reconnected to obtain new IP addresses. ADSL dial-up exhibits the following three characteristics:

- Limited IP Acquisition Range: Only adjacent IP segments can be obtained, preventing access to larger network segments or geographic areas.
- Slow IP Switching Speed: Due to the required interval between redial attempts, efficient IP switching cannot be achieved.
- Low Automation: At this stage, IP switching via dial-up cannot achieve a high degree of automation. Defending against these early IP switching techniques is relatively straightforward. Protection can be achieved by building a database of accumulated blacklisted IP addresses and implementing simple IP frequency rules.

2. Phase Two: Countermeasures Against Proxy IP Fraud Techniques

As business IP risk controls tighten, the black/gray industry escalates its IP countermeasures. This phase primarily involves masking real IPs with proxy IPs, bypassing business IP frequency control policies by disguising their origins. The proxy IP mechanism is illustrated in Figure 6.5.

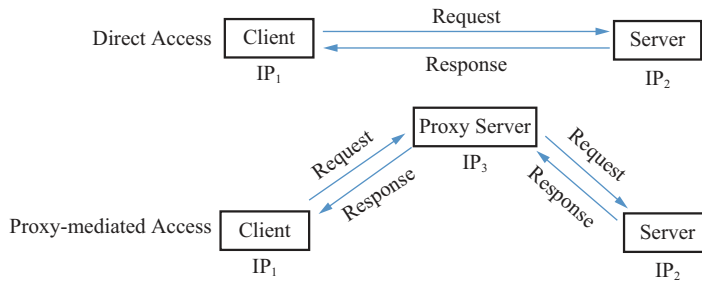


FIGURE 6.5 How proxy IPs work?

As shown in Figure 6.5, when the black/gray industry initiates requests to a target website, the request is normally sent directly to the target website's server. The target website's server then returns the request results directly to the black/gray industry's client. In this scenario, the black/gray industry's IP address is exposed to the target website, leading to interception by risk control systems. However, when a proxy server (i.e., proxy IP) is introduced, it acts as an intermediary—essentially creating a relay station between the black/gray industry and the target website. Instead of directly requesting the target website, the black/gray industry sends requests through the proxy server. The target website returns the response to the proxy server, which then forwards it to the black/gray industry client. This approach ensures that only the proxy server's IP is exposed to the target site while concealing the black/gray industry IP. By continuously rotating among numerous proxy IPs, they can circumvent the target site's IP risk control strategies.

While the black/gray industry can cleverly conceal its real IP addresses using proxy IPs, legitimate businesses can counteract this by targeting the sources of proxy IPs. There are primarily two countermeasures:

- **Countermeasure Based on Proxy IP Blacklists.** This approach targets low-quality proxy IPs by establishing proxy IP reputation databases for interception. For instance, by scraping or scanning the internet to collect publicly available free proxy IPs—which are shared and reused by multiple parties—businesses can compile these IPs into proxy IP reputation databases for blocking.
- **Countermeasure Based on Proxy IP Protocol Detection.** This approach targets high-quality proxy IPs, such as those privately built by proxy providers that offer short-term high anonymity. Since these IPs are not publicly available on the internet, they are generally difficult to collect in proxy IP reputation databases. As proxy IPs are essentially server IPs, they can be blocked using established IDC server IP reputation databases. Additionally, they can be identified by detecting common ports associated with different proxy protocols. Depending on the protocol type, proxy IPs' common ports can be categorized into the following four types, as shown in Table 6.3.

TABLE 6.3 Proxy IP Protocol Types

Proxy Type	Proxy Protocol	Common Ports	Purpose
HTTP proxy	HTTP	Ports 80, 8080, 3128, etc.	Forward HTTP requests, typically for web access
HTTPS proxy	HTTPS	Port 443, etc.	Forward HTTPS requests, typically for HTTPS proxies, but can also be used for HTTP proxies
Socks proxy	Socks4 or 5	Port 1080	Can forward any type of request
FTP proxy	FTP	Ports 21, 2121, etc.	FTP forwarding, jump server, caching

3. Phase Three: Countermeasures Against Fraudulent Tactics Using Second-Dial IP Addresses

Due to limited proxy IP resources and their infrequent use by legitimate users, proxy IPs are easily blocked by service providers’ risk control policies. Consequently, the black/gray industry has shifted its focus to broadband IP pools with more abundant resources. It employs second-dial IP technology to blend illicit traffic with legitimate broadband user traffic, thereby circumventing IP-based risk controls imposed by service providers.

Due to the scarcity of IPv4 resources, operators utilize dynamic IP allocation mechanisms to maximize the use of idle IPs within shared pools. The second-dial IP technology employed by the black/gray industry capitalizes on this dynamic allocation feature. By leveraging virtualization and cloud computing technologies, they package their broadband resources into cloud services, managing and allocating these resources centrally through software-defined routers. Capable of switching IPs within seconds, this technology is termed “second-dial IP.” Integrating second-dial IP resources across multiple provinces and cities creates VPS mixed-dial IP. Current mainstream VPS mixed-dial IP tools can combine IP resources from hundreds of domestic cities, with some even incorporating select overseas IP resources.

Because VPS mixed IPs enable cross-city resource allocation with rapid switching, the traditional IP reputation database becomes ineffective, significantly increasing the difficulty of IP-based business protection. While it is difficult to directly identify black/gray industry activities using VPS mixed IPs from an IP perspective, device-level analysis reveals that a single device switches between numerous IPs within a short timeframe. These IPs span multiple provinces and regions, deviating from normal device IP usage patterns. This anomaly can be leveraged to develop strategies for identifying VPS mixed IP activities by black/gray industry actors. Since VPS mixed IPs are intermingled with legitimate users’ broadband traffic, risk control strategies must be tailored to specific business data conditions.

4. Phase Four: Countermeasures Against IP Magic Box Fraud Tactics

As the battle between the two sides intensified, the black/gray industry gradually shifted its focus to base station IP resources. Since these IPs host numerous legitimate users, they provide an ideal cover for the black/gray industry to conceal its activities

by leveraging the large pool of normal users. Utilizing second-dial IP technology, the black/gray industry has developed tools like the IP Magic Box that rapidly and automatically switch between base station IPs, further circumventing service providers' IP risk control rules.

The IP Magic Box is a compact hardware device featuring core components like USB ports and a SIM card slot. The SIM slot typically supports mobile cards from China's three major carriers—China Unicom, China Mobile, and China Telecom—with some models even accommodating international SIM cards. When connected to a PC via USB, the IP Magic Box enables the PC to access 5G or 4G mobile networks. The PC then utilizes the base station IP of the 5G or 4G network. Combined with the accompanying automated script tool, this allows for rapid, automatic switching of base station IPs.

Since the IP Magic Box utilizes base station IP resources—which typically impact a broad user base—service providers hesitate to block these hidden IPs used by members of the black/gray industry. Even if identified, they avoid indiscriminate blocking to prevent collateral damage to legitimate users. In such cases, data from the IP Magic Box can serve as training samples for models, providing suspicious data points for further verification through device-level analysis. Next, we will elaborate on the practical process through the evolution of defense and attack strategies at the device level.

6.2.2 Evolution of Device Strategy: Offense and Defense

As the hardware carriers of internet information, devices are indispensable resources for black/gray industry traffic fraud. To maximize profits while minimizing hardware costs, members of the black/gray industry typically push device usage to its limits. For instance, a single device may simultaneously log into multiple fraudulent accounts. However, businesses often impose restrictions on the number of accounts a device can access. To circumvent these device-level restrictions, members of the black/gray industry have developed a series of fraudulent tactics. The evolution of device-level fraud techniques is illustrated in Figure 6.6.

Next, we will elaborate on the offensive–defensive evolution between business risk control measures and black/gray industry circumvention tactics.

1 Phase 1: Countermeasures Against Fake Devices and Fake Users in the Black/Gray Industry

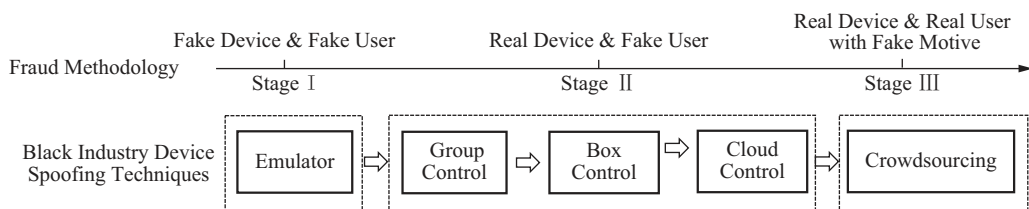


FIGURE 6.6 Evolution of fraud tactics at the device level of the black/gray industry.

Due to business restrictions on device registrations or account logins, the black/gray industry first attempted to circumvent these by creating fake devices and using non-genuine user accounts. This phase involved two primary types of fake devices:

- **Device Modification:** Primarily achieved through tools that alter device information at the system level. This causes the business platform to receive fabricated data, rendering device fingerprints based on system parameters partially ineffective. Examples include altering the device's IMEI, MAC address, model, and other system information.
- **Emulator:** Primarily involves creating multiple virtual mobile devices on a PC via an emulator. Each virtual device mimics a real device by spoofing system information such as IMEI and MAC addresses, achieving a convincing level of authenticity.

To combat traffic fraud involving counterfeit devices from the black/gray industry, service providers implement countermeasures based on specific types of fraud. Two common approaches are as follows:

- **Countermeasures Against Device Tampering by the Black/Gray Industry.** Prior to Android Q, the primary unique device identifier was the IMEI. The black/gray industry counterfeits IMEI information through device tampering. To address this, service providers primarily employ two methods: checking against common IMEI ranges or utilizing the inherent validity checks built into the IMEI itself. For example, calculating the check digit using the Luhn algorithm on the first 14 digits of the IMEI and verifying it against the 15th digit can identify invalid IMEIs forged by the black/gray industry. After Android Q, due to restrictions like IMEI reporting requiring user authorization, major manufacturers began developing decentralized unique device IDs—the device OAID. However, since OAID cannot be validated for legitimacy like the IMEI, alternative solutions are required for preventing device modification under the OAID identification system.
- **Countermeasures against Black/Gray Industry Emulators:** Emulators are more prevalent on PCs than on mobile devices. For emulators used by members of the black/gray industry on PCs, prevention can be approached in two ways: First, examine the application-level information reported by the virtual device to the service provider. Check for any emulator-specific built-in applications, such as the emulator's input method or app store. Second, analyze the system information reported by the virtual device to the service provider. Look for specific keywords related to "emulators," such as model numbers unique to emulators or system information like Wi-Fi and Bluetooth configurations.

2 Phase Two: Countermeasures Against Fake Users on Genuine Devices

Traces of traffic fraud perpetrated by the black/gray industry using fake devices and fake users are relatively obvious and easily detected by business platforms. Consequently, these actors have shifted from fake devices and users to genuine

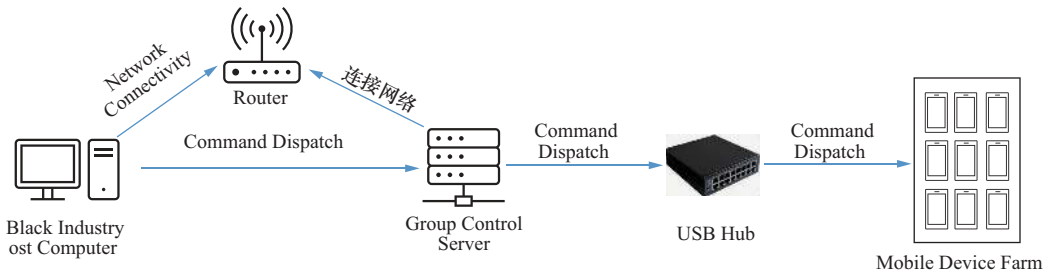


FIGURE 6.7 Principle of real-device group control.

devices with fake users—replacing fake devices with real device groups for group control to increase detection difficulty. Real device groups primarily involve a single PC controlling multiple mobile devices in bulk to execute fraudulent operations. The principle of real device groups is illustrated in Figure 6.7.

As shown in Figure 6.7, physical connections are required between the PC and mobile devices, precluding remote control.

When the black/gray industry needs to scale up, it requires investing in more genuine devices. However, this presents challenges: a large number of devices occupy significant space, making operations and management inconvenient and costly. Consequently, box control gradually emerged for the black/gray industry.

Unlike group control systems that directly manage real devices, box control unifies management by integrating the motherboards of multiple phones. This approach removes bulky components like phone casings, retaining only the core hardware. The display information from these phones is then transmitted to a PC for centralized visualization. This significantly reduces the physical footprint and substantially lowers operational costs for the black/gray industry.

Both real-device group control and box control utilize actual devices. To reduce costs, members of the black/gray industry typically purchase inexpensive used phones in bulk from second-hand markets. Some of these devices have even been phased out of the market and are rarely used by regular consumers. Additionally, real-device group control systems are typically mounted on phone stands and remain stationary, constantly connected to power sources. Based on these characteristics, service providers can implement comprehensive identification through system information such as phone model, gyroscope, accelerometer, and battery status.

As countermeasures evolve, cloud-based botnets have emerged as an advanced iteration of group control, enabling remote, bulk device management. Cloud-based botnets eliminate the need for physical connections between phones and PCs. Once a phone gains root access and installs the cloud-based botnet system, remote control becomes possible. Compared to traditional group control, cloud-based botnets are not geographically restricted and do not require devices to be physically centralized in one location. This aligns more closely with typical user behavior, increasing the difficulty for service providers to detect and prevent such activities. At this stage,

service providers can monitor whether a phone has obtained root access or installed suspicious tools related to the cloud-based botnet system.

3 Phase Three: Countermeasures Against the Black/Gray Industry Using Genuine Devices and Users with Fake Intentions

The black/gray industry replaced fake devices with real ones, which somewhat increased the difficulty of device identification. However, since these were still not genuine users, businesses could establish behavioral strategies for detection and prevention. Consequently, criminals shifted to using real devices with real users but fake motivations, further complicating identification. This “real devices, real users, fake motives” approach involves crowdsourcing tasks to ordinary part-time users who complete them for monetary rewards. Since these users are genuine and their devices are legitimate, only their motives are questionable, significantly complicating detection for businesses.

To counter the black/gray industry crowdsourcing approach, businesses primarily focus on device-level detection and use tools to identify crowdsourcing activities.

6.2.3 The Evolution of Account Strategy Offense and Defense

In internet traffic scenarios, accounts serve as the unique identifier for users within business systems—functioning as the “ID card” for platform activities. Consequently, accounts are indispensable resources for black/gray industry traffic fraud. Risky accounts primarily originate from three sources: bulk registration, account theft, and malicious account unblocking. The following sections detail countermeasure strategies at the account level based on these source types, with the overall account-level defense framework illustrated in Figure 6.8.

6.2.3.1 Preventing Risk Accounts from Batch Registration
Sources of the Black/Gray Industry

Users primarily access business platforms through two methods: quick login via third-party accounts and login via registered accounts. Quick login via third-party accounts involves users authorizing the binding of third-party accounts (e.g., WeChat and QQ) directly to

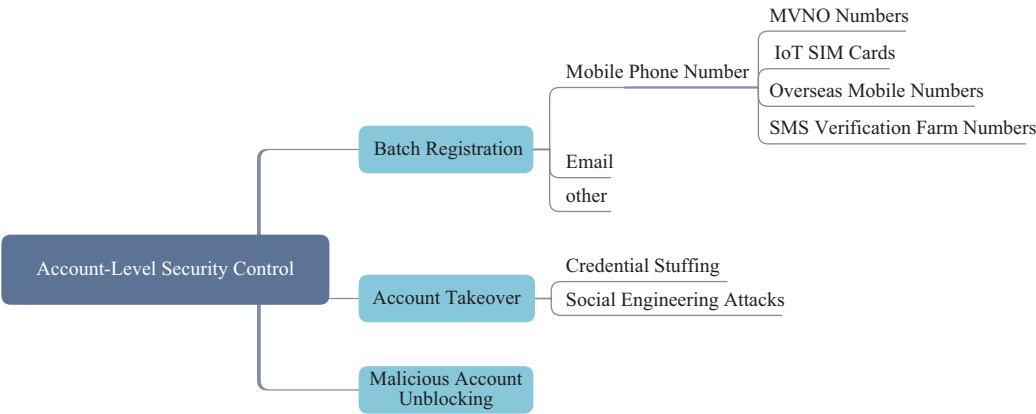


FIGURE 6.8 Account-level prevention approach.

their current business account ID for swift access. Accounts logged in this manner are primarily controlled through business-customized rules. Account registration login involves creating an account using key user information like mobile phone numbers or email addresses before logging in. Accounts created this way can be controlled using generic rules. The following section focuses on bulk registration via mobile phone numbers as an example.

Bulk registration of black/gray industry accounts using mobile phone numbers can be categorized into four primary methods: virtual operator numbers, IoT numbers, overseas mobile numbers, and SMS verification platforms. The following sections detail prevention strategies for each of these four bulk registration methods.

6.2.3.1.1 Countermeasures for Virtual Operator Numbers When the black/gray industry aims to obtain large numbers of service accounts through bulk registration, its primary challenge is securing sufficient mobile phone numbers for the registration process. Regularly obtained mobile phone numbers from service centers require real-name verification, which presents high barriers to entry and quantity restrictions. Consequently, the black/gray industry turns to obtaining mobile phone numbers from virtual network operators (VNOs). These mobile phone numbers, typically starting with prefixes like 162, 165, 170, and 171, can be obtained online, making it easier to bypass real-name verification and relatively inexpensive to acquire—attracting the black/gray industry. Service providers counter bulk registration using MVNO mobile phone numbers primarily by implementing MVNO prefix-based strategies for identification and prevention, thereby raising the barriers for black/gray industry activities.

6.2.3.1.2 Preventing IoT Number Abuse Beyond virtual operator numbers, IoT numbers represent another low-cost channel for the black/gray industry to acquire mobile phone numbers. IoT numbers are pure data SIM cards that start with prefixes like 146 or 148, primarily used for connecting and managing smart hardware. Since pure data plans are significantly cheaper than standard SIM cards, they are highly favored by the black/gray industry. To counter bulk registration using IoT numbers, service providers primarily implement IoT number range strategies for identification and prevention, thereby raising the barriers for the black/gray industry to register and utilize these numbers.

6.2.3.1.3 Countermeasures Against Overseas Mobile Phone Numbers As service providers tighten controls on domestic MVNO and IoT numbers, the limited pool of mobile phone numbers available for bulk registration by the black/gray industry has prompted them to shift focus overseas. Mobile SIM cards in Southeast Asian countries support GSM networks, require no real-name verification for immediate use, and offer inexpensive, low-cost plans. Consequently, members of the black/gray industry have started using overseas mobile phone numbers to replace their previous blacklisted resources. To counteract the bulk use of overseas mobile phone numbers during registration, service providers primarily rely on comprehensive identification methods, including maintaining blacklists of overseas mobile phone numbers and assessing the risk levels of overseas geographic locations.

6.2.3.1.4 Countermeasures Against Mobile Phone Numbers from SMS Verification Platforms In traffic fraud scenarios, upstream black/gray industry supply chains provide abundant mobile phone numbers from SMS verification platforms. These mobile phone numbers can be directly purchased for bulk registration. Businesses can monitor and collect SMS verification platform blacklisted numbers to build a database, thereby achieving greater control with less effort.

6.2.3.2 Preventing Black/Gray Industry Risk Accounts from Account Theft Sources

Currently, the mainstream account theft methods employed by the black/gray industry fall into two primary categories: credential stuffing attacks and social engineering attacks. The following sections will provide detailed prevention strategies for both of these account theft methods.

6.2.3.2.1 Preventing Account Theft via Credential Stuffing As internet platforms proliferate, users face an increasing number of passwords to remember across various services. For simplicity and convenience, most users habitually reuse the same password across different platforms, exposing themselves to the risk of account theft through credential stuffing by the black/gray industry. The basic principle of this attack method is as follows: the black/gray industry first breaches a less secure Platform A to obtain user credentials, then uses the stolen information to brute-force match credentials on Platform B and steal accounts. The black/gray industry chain for credential stuffing attacks, as shown in Figure 6.9, primarily consists of five stages: database dumping, data laundering, credential stuffing, monetization through sales, and social engineering databases.

- Database Dumping: This stage primarily involves the black/gray industry exploiting vulnerabilities through methods like vulnerability scanning or SQL injection to attack business databases and illegally obtain user personal information in bulk. Large internet platforms typically possess robust risk control capabilities, making them difficult to breach. Consequently, databases of smaller internet platforms with weaker security often become targets.
- Data Laundering: The black/gray industry processes the user information obtained in the previous step through cleaning, filtering, and categorization to extract valuable

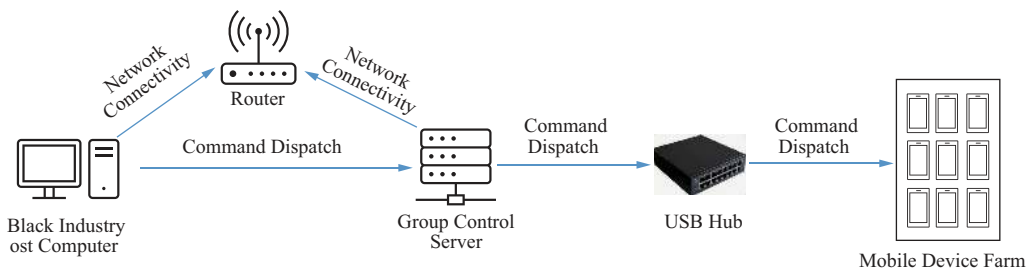


FIGURE 6.9 Black/gray industry chain for account theft via credential stuffing.

personal details or accounts, such as gaming accounts containing virtual currency or equipment.

- **Credential Stuffing:** Using the user information compiled during the data laundering process, the black/gray industry employs automated tools to brute-force match or crack accounts on other platforms.
- **Monetization:** The black/gray industry sells the valuable personal information or accounts it has screened to downstream players in the industry chain to generate revenue.
- **Social Engineering Database:** User personal information extracted through database dumping, data laundering, and credential stuffing is ultimately consolidated into the black/gray industry's social engineering database. They leverage user profiling within this database to target specific demographics for fraud or extortion.

Accounts compromised via credential stuffing are typically vulnerable due to weak initial password strength, the reuse of credentials across multiple platforms, lax account management, or low account activity. Therefore, business entities can adopt a two-pronged prevention strategy: first, pre-breach prevention—business platforms should require password complexity based on their security classification. Second, post-breach prevention—addressing traffic fraud by the black/gray industry, which exploits stolen credentials. This involves considering the four stages a compromised account may undergo upon reuse, as illustrated in Figure 6.10.

Business units must develop targeted strategies to address these four stages.

First, during new device login, the black/gray industry may attempt to log in through account scanning or credential stuffing. The defense strategy for new device login is illustrated in Figure 6.11. This strategy is structured as a funnel: first, the password is verified. If the password is entered incorrectly too many times, it may indicate repeated brute-force



FIGURE 6.10 Four stages of reusing stolen accounts.

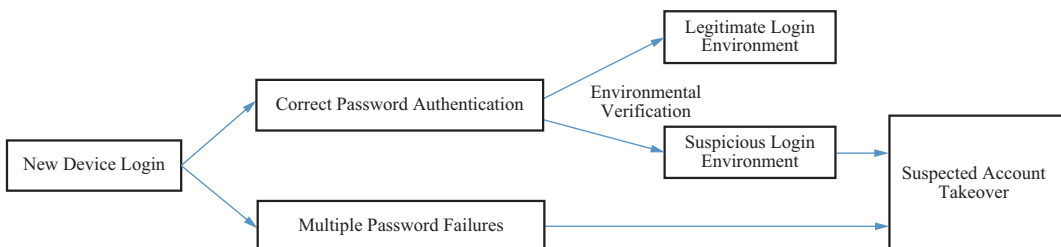


FIGURE 6.11 Prevention strategy for new device login.

attempts by the black/gray industry, posing a suspected account theft risk. If the password is correct but the login environment verification stage shows anomalies, there is also a suspected account theft risk. This is because the black/gray industry, while knowing the correct password, often exhibits clustered abnormal environments, such as logging into multiple stolen accounts from the same shared IP address.

Second, during the secondary verification stage, verification strategies of varying difficulty should be implemented based on risk levels. Accounts flagged as suspicious in the initial step must undergo secondary verification. This stage establishes verification questions of differing complexity according to the account's risk level, further raising the login threshold and increasing the cost of login fraud for the black/gray industry.

Finally, within N hours after login, the account's behavior must be closely monitored and subjected to risk detection. After bypassing the new device login risk control and successfully accessing a compromised account, black/gray industry cybercriminals' activities often deviate from normal account behavior. The primary objective of individuals from the black/gray industry accessing compromised accounts is financial gain. Their activities frequently involve transferring assets from the original account, altering profile pictures and signatures for traffic diversion, posting large volumes of non-compliant content, and executing fraudulent schemes. Consequently, detecting abnormal behavior within N hours post-login can further mitigate black/gray industry activities.

6.2.3.2.2 Preventing Account Theft via Social Engineering The black/gray industry employs pre-scripted scenarios to gradually lure users into revealing critical information like passwords or CAPTCHA codes, ultimately stealing accounts. Due to the higher cost of social engineering attacks, the black/gray industry primarily targets high-value social media accounts or premium business accounts.

Service providers' prevention strategies can be divided into two main aspects: first, pre-theft prevention, where providers educate users on security and issue risk warnings based on currently prevalent fraud scripts; and second, post-theft prevention, which mirrors the approach for credential stuffing attacks, focusing on preventing the reuse of stolen accounts.

6.2.3.3 Preventing Malicious Unlocking of Accounts from the Black/Gray Industry

As internet service providers tighten their security measures, a large number of accounts from the black/gray industry have been suspended, leading to a shortage of available accounts for these operations. Consequently, the black/gray industry has resorted to batch account unblocking requests to regain access to suspended accounts. This malicious unblocking method primarily leverages the vast resources of ID cards and mobile phone numbers from virtual operators controlled by the black/gray industry. It bypasses the strategies established by service providers during the unblocking process, enabling malicious bulk unblocking and the reacquisition of large numbers of unblocked account resources.

To counteract the risk of malicious account unblocking by members of the black/gray industry, service platforms can build their prevention strategies around the following three aspects.

- **Frequency Control Strategy for Unlocking:** A single account can only be unlocked using the same mobile phone number or ID within a specified timeframe. Similarly, a single mobile phone number or ID can unlock only one account within the same timeframe. This raises the barrier for black/gray industry actors attempting malicious account unlocks.
- **Guarantor-Assisted Prevention Strategy:** When an account requests unblocking, the applicant must obtain confirmation from N normal users to proceed. This increases the unblocking threshold and raises the cost for the black/gray industry to unblock accounts.
- **Facial Verification Prevention Strategy:** When an account applies for unblocking, the applicant must undergo real-time dynamic facial verification. Approval is granted only when the applicant's facial features match the facial information associated with the ID document linked to the account.

6.3 BUSINESS-CUSTOMIZED RULES

Generic rules are primarily constructed from the perspective of essential resources required for black/gray industry fraud operations, utilizing relatively few data dimensions, making them easily circumvented by fraudsters. In contrast, business-customized rules are derived from hundreds or thousands of business data dimensions. While this significantly increases the cost for black/gray industries to counteract, it also imposes high labor costs on business units for rule extraction. Consequently, the effectiveness of countermeasures is largely constrained by the level of human resources invested. Therefore, establishing an intelligent rules engine can reduce labor costs and enhance the effectiveness of combating black/gray industries.

The intelligent rules engine system, as shown in Figure 6.12, consists of six core modules: rule intelligent preprocessing module, rule intelligent construction module, rule intelligent

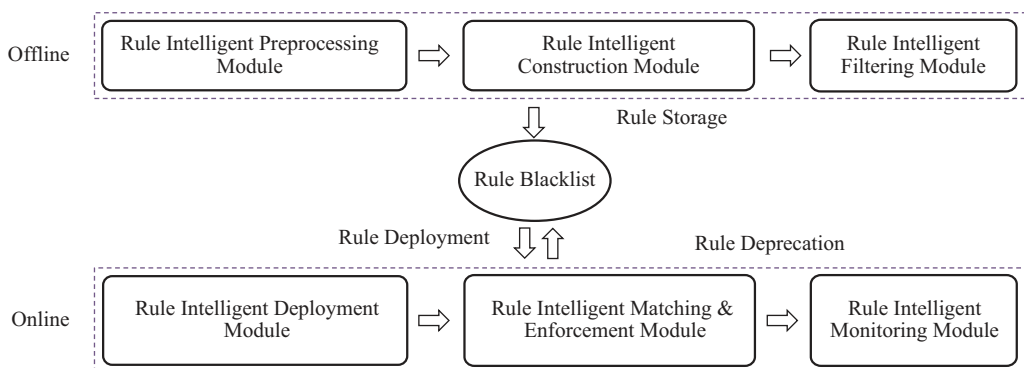


FIGURE 6.12 Intelligent rules engine system.

screening module, rule intelligent deployment module, rule intelligent matching and penalty module, and rule intelligent monitoring module.

6.3.1 Rule Intelligent Preprocessing Module

Given the diverse types of raw logs generated by internet services and the varied formats of their fields, input formats must be standardized before entering the intelligent rules engine system. The following explanation uses device-level business-customized rules as an example.

- Time Range Trimming: To detect fraud promptly, the time range for cloud-based data selection should not be too long; it is best to filter data from the most recent K hours.
- Column-Based Pruning: To reduce computational load, unnecessary fields with no obvious malicious traces are removed.
- Row-Based Pruning: Since client data submissions may fail, some rows may contain null values. Rows with null values that carry no meaningful information can be discarded.
- Field value discretization.

After preprocessing the data using the above methods, standardized results can be obtained. Device profiling features data for traffic fraud environments is shown in Table 6.4.

6.3.2 Rule-Based Intelligent Construction Module

6.3.2.1 Step 1: Initial Screening of Rule Dimensions Based on 1-Gram Analysis

Although we have already eliminated redundant fields in the rule intelligent preprocessing module, this is only a preliminary pruning based on business experience. Some redundant dimensions still require computational analysis for removal. Here, we primarily assess and eliminate redundant dimensions by calculating the risk level of user groups associated with a single dimension. For example, by calculating attribute values such as the proportion of blacklisted users, fraud report ratios, and whitelisted user ratios, we comprehensively evaluate and eliminate single dimensions corresponding to user groups with very low risk levels. This achieves dimensionality reduction, minimizing the computational load in subsequent steps. Assuming a threshold of blacklist ratio ≥ 0.1 and whitelist ratio ≤ 0.6 , the preliminary screening results based on 1-gram rules are shown in Table 6.5.

TABLE 6.4 Device Profile Features Data in Traffic Fraud Scenarios

Type	Profile Feature	Meaning
Device	F_1	Describes the fundamental attributes of the device, such as model type
	F_2	Characterizes the tool-related attributes of the device, such as crowdsourcing
		
	F_N	Characterizes the tool-class attributes of risk devices, such as multi-instance operation

TABLE 6.5 Preliminary Screening Results Based on 1-gram Rules

Device Profile Features	Profile Feature Discrete Values	Blacklist Ratio of Associated Device Groups	Whitelist Ratio of Associated Device Groups
F_5	F_5 = Automated Script Risk Level A	0.36	0.41
F_7	F_7 = Modified Device Risk Level C	0.14	0.61
F_N	$F_{(N)}$ = Risk Level B for Multiple Openings	0.17	0.54

TABLE 6.6 2-Gram Rule Combination Results

No.	2-gram Dimension Combination Rules	Associated Device Groups
Rule 1	F_5 = Automated Script Risk Level A & F_7 = Device Modification Risk Level C	Device Group 1
Rule 2	F_5 = Automated Scripting Risk Level A & F_N = Multi-Instance Risk Level B	Device Group 2
Rule 3	F_7 = Modified Device Risk Level C & $F_{(N)}$ = Multi-Instance Risk Level B	Device Group 3

TABLE 6.7 Screening Results

No.	2-gram Dimension Combination Rule	Blacklist Ratio of Associated Device Groups	Whitelist Ratio of Associated Device Group	Malicious Rule Screening
Rule 1	F_5 = Automated Script Risk Level A & F_7 = Modified Device Risk Level C	0.65	0.21	Non-compliant
Rule 2	F_5 = Automated Scripting Risk Level A & F_N = Multi-Instance Risk Level B	0.85	0.08	Compliant
Rule 3	F_7 = Modified Device Risk Level C & $F_{(N)}$ = Multi-Instance Risk Level B	0.33	0.45	Non-compliant

6.3.2.2 Step 2: Automatic Rule Combination Based on n -Grams

Based on the risk dimensions preliminarily screened in Step 1, further construct n -gram rules through n -gram combination methods. Finally, associate user groups based on the constructed n -gram rules. Using a simple 2-gram combination as an example, we automatically pair the discretized values from the three dimensions identified in Step 1. This yields a total of three possible combinations. The results of the 2-gram rule combinations are shown in Table 6.6.

6.3.3 Rule Intelligent Screening Module

Based on the rules and associated user groups generated by the intelligent rule construction module, further calculate attribute values for user groups (e.g., blacklist ratio, whitelist ratio, and fraud report ratio). Then evaluate and filter out rules corresponding to high-risk user groups, storing them as malicious rules in the rule blacklist database. Assuming a blacklist ratio ≥ 0.8 and a whitelist ratio ≤ 0.1 are selected as criteria for malicious rules, the screening results are shown in Table 6.7. Ultimately, Rule 2 is identified as a malicious rule and added to the rule blacklist.

6.3.4 Other Modules

- **Rule Intelligent Deployment Module:** Automated tools for rule deployment that automatically upload newly added malicious rules from the rule blacklist to production via a gray-scale approach.
- **Rule Intelligent Matching and Penalty Module:** Generates n-gram rules in real-time for low-trust users using the same method as offline rule generation. These rules are matched against live malicious rules; successful matches trigger penalties.
- **Rule Intelligent Monitoring Module:** Since deployed malicious rules may inevitably cause false positives, an automated rule elimination mechanism is required. This primarily involves two approaches: first, a passive elimination mechanism: real-time calculation of the user complaint ratio for false positives per online malicious rule. Upon triggering an alert threshold, the corresponding rule's penalties are immediately invalidated, and the rule is removed from the malicious rule database.

6.4 SUMMARY

This chapter further elaborates on the evolution of business risk control and the battle against black/gray industries from the perspective of the rules engine. It begins by introducing risk lists—critical information for upfront risk screening—detailing their evolution across different product phases and deployment management approaches. Subsequently, it explores how businesses create generic rules for prevention and control by focusing on three essential resources related to black/gray industries: IP addresses, devices, and accounts. Finally, it presents the intelligent rules engine system from the perspective of business-customized rules, further enhancing the prevention efficiency of business units.

Machine Learning Countermeasures

As a core countermeasure in the early stages of business protection, the rules engine features low resource overhead, deep integration with business scenarios, and rapid identification and blocking of fraudulent traffic. Consequently, it is an indispensable component in traffic fraud scenarios. However, rules engines heavily rely on the expertise of rule creators. Additionally, individual rules typically utilize a limited number of feature dimensions, making them susceptible to exploitation by the black/gray industry, which identifies rule boundaries and circumvents them, ultimately rendering the rules ineffective.

To address the low detection coverage of black/gray industry traffic caused by circumventing early-stage solutions like rules engines, this chapter introduces machine learning countermeasures. These solutions fundamentally start from traffic data, extracting its inherent patterns (feature representations) to train predictive models. These models score traffic and, combined with business evaluation criteria, filter out suspicious fraudulent traffic. This chapter focuses on practical application, explaining how to implement these machine learning countermeasures in traffic fraud scenarios.

Data serves as the foundation of machine learning algorithms. Without sufficient data, these algorithms cannot effectively identify traffic fraud. Once data is acquired, it must first be stored compliantly and appropriately through data governance solutions. Subsequently, feature engineering transforms raw data into features data that models can interpret. The core path for constructing traffic fraud detection models is illustrated in Figure 7.1, outlining key stages from governing raw log data to building traffic fraud detection models.

Within this core path, numerous machine learning algorithms exist, each suited to distinct scenarios. Practical selection requires first considering the fraud techniques, manifestations, and impacts; second, evaluating the platform's data sources and data pattern characteristics; and finally, assessing business evaluation criteria. This holistic approach enables comparative analysis of algorithms and their training parameter choices. Therefore, before selecting an algorithm, it is crucial to clarify several key questions: What are the

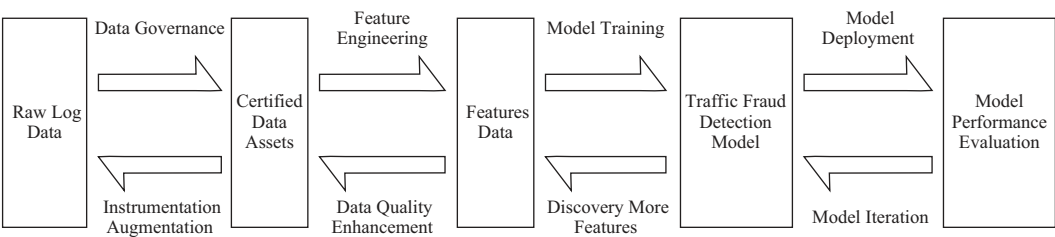


FIGURE 7.1 Core path for building traffic fraud detection models.

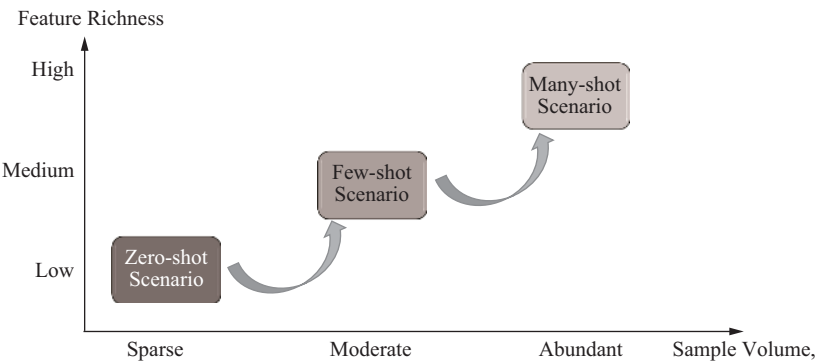


FIGURE 7.2 Evolution of machine learning countermeasure solutions.

fraud techniques? What are the manifestations of fraud? What are the harms? And what are the governance standards? In traffic fraud scenarios, building a machine learning solution from scratch typically involves the following three stages, as illustrated in Figure 7.2.

- Phase One: In the early stages of adversarial learning, with limited samples available, anomaly detection must be performed in a zero-shot scenario.
- Phase Two: As governance progresses, a sample pool gradually accumulates. However, early samples are incomplete, containing only partial accurate anomalous traffic samples. At this stage, single-sample detection frameworks can be employed.
- Phase Three: As the dimensions of sample collection expand and evaluation criteria mature, sufficient positive samples and negative samples will accumulate, representing most fraudulent behaviors. At this stage, various complex supervised learning algorithms can be implemented.

Therefore, this chapter categorizes traffic fraud as a sample-centric scenario-based problem. It first introduces unsupervised machine learning algorithms for identifying anomalous traffic in zero-shot scenarios; then covers semi-supervised machine learning algorithms for few-shot scenarios; and finally addresses supervised machine learning algorithms for many-shot scenarios.

7.1 ZERO-SHOT SCENARIO

In the pre-traffic phase of identifying traffic fraud, the absence of fraudulent traffic samples prevents the construction of positive and negative samples, characterizing this phase as an unsupervised scenario. During this phase, there is a lack of understanding of fraudulent traffic patterns. Additionally, the sheer volume of traffic makes it challenging to manually identify anomalies through pattern extraction. However, since fraudulent traffic is inherently anomalous and differs from normal traffic, it can be detected through anomaly detection approaches.

From an anomaly detection perspective, anomalous traffic monetization typically manifests in two forms: one involves data values that are significantly higher or lower than normal traffic, which can be identified using traditional statistical detection frameworks; the other appears as outliers after clustering, which can be detected through unsupervised learning methods.

In certain fraud detection scenarios, fraudulent traffic exhibits data values that are either higher or lower than normal traffic, allowing fraud to be identified through these metrics. For example, registration or login pages often employ sliding CAPTCHAs to detect fake clicks. Typically, completing the pattern puzzle by sliding passes the CAPTCHA verification. However, the underlying logic of the CAPTCHA involves capturing the user's trajectory data during puzzle completion. This data, combined with information such as sliding fluctuations, sliding speed, and pattern overlap, is used to comprehensively determine whether the action constitutes a fraudulent click. Details regarding traditional statistical testing approaches will be elaborated in Section 7.1.1.

Simple maxima and minima are insufficient to distinguish fraudulent traffic from normal traffic. In such cases, unsupervised learning approaches can be employed for identification, as fraudulent traffic often manifests as outliers. From the perspective of feature distribution in dimensional space, normal traffic exhibits centralized patterns, whereas fraudulent traffic forms distinct clusters of outliers. Figure 7.3 illustrates both outliers and

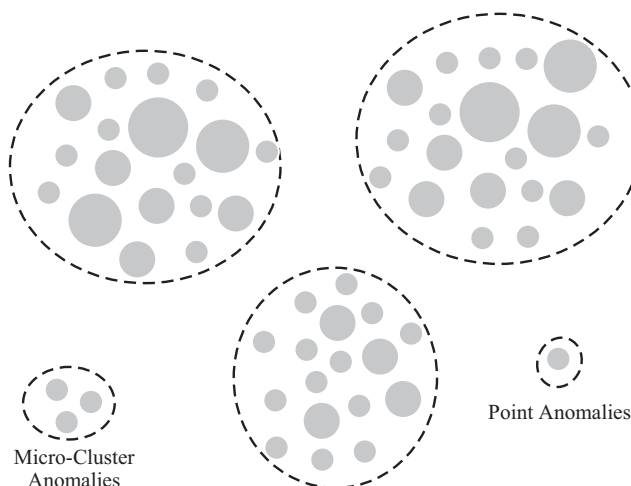


FIGURE 7.3 Outliers and outlier clusters.

outlier clusters, demonstrating that detecting outliers can also identify fraudulent traffic. Details regarding the unsupervised learning approach will be elaborated in Section 7.1.2.

7.1.1 Traditional Statistical Testing Approach

Using the example of identifying fraudulent clicks through sliding CAPTCHAs mentioned earlier, we explain how to detect fraudulent traffic based on traditional statistical testing methods. The overall process can be divided into the following three steps.

- 1. The user completes the sliding CAPTCHA operation on the front-end page.
- 2. The backend captures information such as the user’s mouse swipe trajectory, total swipe duration, and the overlap between the slider and the gap.
- 3. The backend performs human–machine discrimination on the data. If the action is determined to be a fraudulent click, the login or registration process is terminated, or the user is redirected to another method for CAPTCHA verification.

The following section uses the total sliding time metric as an example to explain the analysis process for identifying traffic fraud based on traditional statistical testing methods.

First, define the evaluation criteria: what constitutes machine-generated sliding behavior. Most normal users take 1–2 seconds to complete a slide. If the sliding time is too short, it may indicate machine behavior. The threshold for identifying machine sliding can be statistically derived using a hypothesis testing approach. This section focuses on confirming thresholds using Tukey boxplots.

The distribution of user swipe durations is shown in Figure 7.4. Analysis of collected swipe durations reveals that the majority of users swipe for 1–2 seconds, with most concentrating within the 0–3-second range.

Next, for user swipe duration, calculate the first quartile (Q1) and third quartile (Q3). Simultaneously compute the interquartile range ($IQR = Q3 - Q1$), along with outlier thresholds ($Q1 - 1.5IQR$ and $Q3 + 1.5IQR$). By applying these calculations to the data in Figure 7.4, the statistical values for the Tukey boxplot are obtained, as shown in Table 7.1. The boxplot for user swipe duration is depicted in Figure 7.5.

Based on the threshold classification, traffic with user swipe durations between 0.152 and 2.800 seconds is identified as normal traffic. Traffic with swipe durations shorter than 0.152

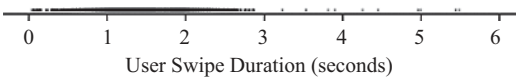


FIGURE 7.4 Distribution of user swipe duration.

TABLE 7.1 Statistical Values for Tukey Boxplots				
Q1	Q3	IQR	$Q1 - 1.5IQR$	$Q3 + 1.5IQR$
1.145	1.807	0.662	0.152	2.800

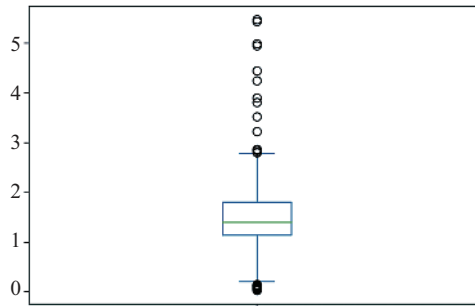


FIGURE 7.5 Boxplot of user swipe duration.

seconds or longer than 2.800 seconds is identified as abnormal verification traffic. However, based on business experience, machine-generated swipes typically occur in very short durations. Therefore, we only flag traffic with durations under 0.152 seconds as suspected machine activity, subjecting it to secondary verification or rejection. In real-world usage scenarios, we do not rely solely on user swipe duration for detection but employ multiple indicators to identify fake traffic. Nevertheless, the fundamental approach remains consistent.

7.1.2 Unsupervised Learning Approach

Section 7.1.1 explained how traditional statistical tests can identify fake traffic in sliding CAPTCHA scenarios. Such scenarios typically occur in environments like login and registration, requiring rapid backend responses to determine whether traffic is fake traffic immediately after the user completes the slider. When maximum and minimum values are not clearly defined, simple statistical rules cannot identify fake traffic. Therefore, unsupervised learning approaches can be used to further filter out fake traffic.

Next, using traffic fraud activities in marketing campaigns as a case study, we explain how unsupervised learning approaches can identify fraudulent traffic.

In marketing campaigns, members of the black/gray industry bulk-claim prizes to resell them for profit. Since these operators control vast resources like devices and accounts (e.g., mobile phone numbers and QQ IDs), statistical anomalies often appear less pronounced across multiple dimensions. This makes it difficult to detect anomalous traffic using simple statistical threshold tests. In such cases, clustering algorithms from unsupervised learning can be employed. These algorithms group users into clusters, identify outliers, and then combine this with business context to determine anomalous traffic.

This section uses clustering algorithms from unsupervised learning as an example to explain detecting “freeloaders” obtaining marketing activity coupons. The process of detecting anomalous traffic based on clustering algorithms is shown in Figure 7.6.

7.1.2.1 Collecting Profile Data

The first step is to collect user profiling data, which can be primarily categorized into device-level, account-level, and group-level dimensions, as shown in Figure 7.7. In specific business scenarios, the black/gray industry typically employs resources like SIM pooling,



FIGURE 7.6 Process for detecting anomalous traffic using clustering algorithms.

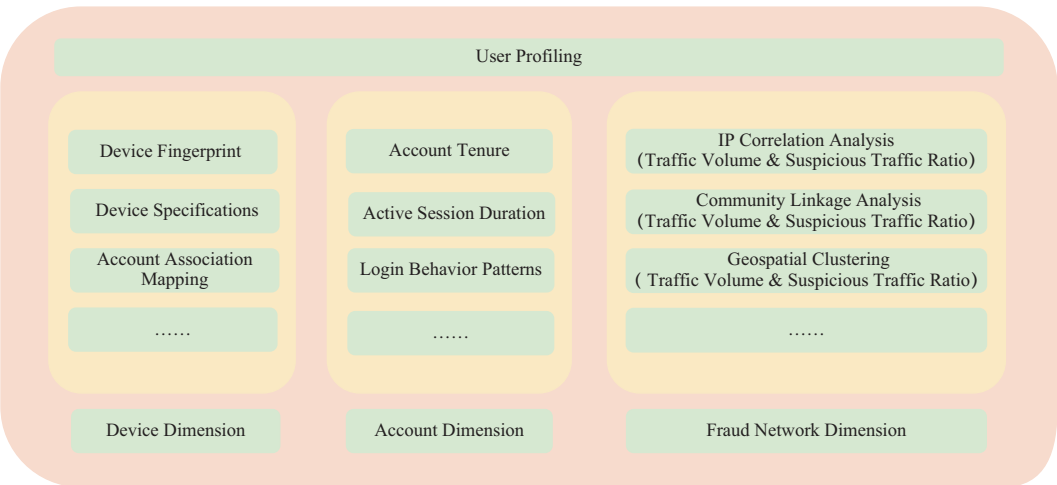


FIGURE 7.7 Three dimensions of user profiling.

group control, secondary accounts, and proxy IPs for fraudulent purposes. Therefore, user profiling at the device and account dimensions can be characterized based on the manifestations of these illicit activities. The device-level profile includes information such as device ID, model, and login account status. The account-level profile encompasses metrics like registration duration, activity period, and login history. Group-level user profiling encompasses traffic volume and the proportion of suspicious traffic originating from shared IPs, communities, or geographic regions. “Freeloaders” typically operate in organized groups. If a sudden surge in traffic or an unusually high proportion of suspicious traffic is detected within a specific cluster (shared IPs, communities, or geographic regions), the traffic within that cluster may originate from freeloaders.

7.1.2.2 Features Data Processing

The second step involves processing the user profiling data collected in the first step. Feature engineering transforms profile data into features data.

7.1.2.3 Cluster Analysis

The third step involves clustering using clustering algorithms. Several clustering algorithms are available for selection. The following three are common clustering algorithms:

TABLE 7.2 Advantages and Disadvantages of Clustering Algorithms

Algorithm	Advantages	Disadvantages
Partition-based clustering algorithms, with common examples including <i>k-means</i> clustering	Low computational complexity	Requires prior determination of cluster number
Density-based clustering algorithms, with common examples including the DBSCAN algorithm	Less sensitive to noisy data	Multiple parameters are involved, and sensitivity to parameter values is high
Hierarchical clustering algorithms	Computational intensity is high, making them unsuitable for high-dimensional data	No need to pre-determine the number of clusters

- **Partition-Based Clustering Algorithms:** These algorithms feature low computational complexity and high partitioning efficiency, making them suitable for relatively regular data patterns.
- **Density-based clustering algorithms** are relatively insensitive to noisy data and effectively identify isolated nodes. However, they involve numerous parameters and are sensitive to those parameters, requiring iterative threshold tuning to achieve optimal clustering results.
- **Hierarchical clustering algorithms** offer greater flexibility and require fewer parameter definitions. However, due to their high computational demands and relatively low clustering efficiency, they are less suitable for high-dimensional data.

Comparing these three clustering algorithms, their respective advantages and disadvantages are summarized in Table 7.2.

7.1.2.4 Identifies Outliers and Detects Anomalous Traffic

The final step involves extracting the small clusters formed after the clustering in the third step. These clusters are then prioritized for verification and review. The review metric typically focuses on the proportion of anomalous users within each cluster. Clusters with a low proportion of anomalous users are disregarded, while those with a high proportion are flagged as anomalous traffic for mitigation. During the initial phase of countermeasures, a manual analysis of sampled users within subclusters is required to identify clusters with high user anomaly rates for targeting. However, as the volume of anomalous users accumulates, a blacklist can be generated from these users. The probability of a subcluster being anomalous traffic is then confirmed by verifying the proportion of its users matching the blacklist. Generally, a higher blacklist match rate indicates a higher probability of the subcluster being anomalous traffic. Specific thresholds must be set for each business scenario. When the blacklist hit rate exceeds the configured threshold, users within that cluster can be blocked or banned. Actions executed at different blacklist hit rates are illustrated in Figure 7.8. Depending on the business context, disposal methods can be flexibly customized.



FIGURE 7.8 Actions executed at different blacklist hit rates.

7.2 FEW-SHOT SCENARIO

A few-shot scenario refers to a situation with only samples from a single category, commonly seen in scenarios with a large number of normal samples. This occurs because the business has clear definitions and standards, along with well-established data engineering practices, making it relatively easy to distinguish normal user samples.

Section 7.1 described how to detect anomalous traffic during the pre-traffic phase of adversarial attacks when no samples are available. However, as adversarial attacks intensify and business standards evolve, a large volume of normal samples and a small number of abnormal samples accumulate. At this stage, directly training a supervised model remains challenging, primarily due to the insufficient number of abnormal samples to support building a model with good generalization performance. Yet, with a sufficient quantity of normal samples, this phase is well-suited for constructing models using semi-supervised learning algorithms.

Before introducing anomalous traffic detection models built using semi-supervised learning algorithms, it is essential to clarify the distinction between outlier detection and singular value detection. Outlier detection occurs when the training set is “contaminated,” meaning it contains both normal and anomalous traffic. In this scenario, detecting anomalous traffic falls under outlier detection. Singular value detection, however, applies to uncontaminated training sets containing only normal traffic. Models built from this normal traffic are then used to determine whether unknown traffic is normal. Therefore, all anomalous traffic detection mentioned in this section is outlier detection, while the anomalous traffic detection discussed in Section 7.1 is also outlier detection.

7.2.1 Traditional Semi-Supervised Learning Approaches

In a few-shot scenario, numerous semi-supervised learning algorithms exist for constructing anomaly traffic detection models, such as One-Class SVM, AutoEncoder, and Isolation Forest. Their core approach involves training models exclusively on normal traffic samples to predict unknown traffic. Traffic exhibiting significant deviation from normal patterns is classified as anomalous traffic.

Next, using the common One-Class SVM algorithm as an example, we will explain how to build an anomalous traffic recognition model using only normal traffic.

The One-Class SVM algorithm can perform both singular value detection and outlier detection; this section focuses on singular value detection. The principle of identifying

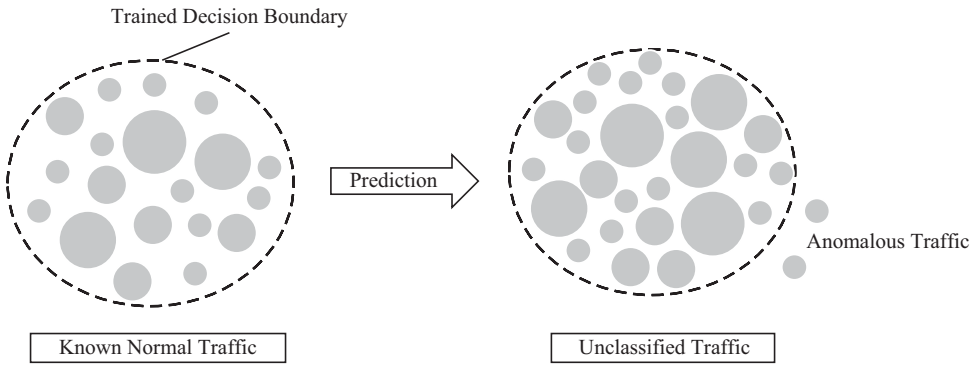


FIGURE 7.9 Principle of anomalous traffic detection using the One-Class SVM algorithm.



FIGURE 7.10 Overall process for anomalous traffic detection using the One-Class SVM algorithm.

anomalous traffic using the One-Class SVM algorithm is illustrated in Figure 7.9. First, by fitting the feature space distribution of normal traffic, a boundary circle for normal traffic can be established. This boundary circle manifests as a hyperplane in the high-dimensional feature space. This hyperplane is then used to classify unknown traffic. If unknown traffic falls within the boundary circle, it is classified as normal traffic; if it lies outside the circle, it is classified as anomalous traffic. As shown in Figure 7.9, the traffic represented by the two points outside the circle constitutes anomalous traffic.

The overall process of identifying anomalous traffic using the One-Class SVM algorithm is shown in Figure 7.10. The core components of the model primarily consist of the following four steps.

7.2.1.1 Normal Traffic Purification

Here, it is essential to ensure that the training dataset consists of normal traffic with minimal anomalous traffic, as the presence of anomalous traffic may distort the learning of normal traffic boundaries. In this step, the method introduced in Section 7.1 can also be applied to further filter traffic, i.e., first identifying and then removing anomalous traffic. However, based on operational experience, it is recommended not to remove a small amount of traffic that is difficult to classify. This approach helps ensure that the subsequent model training maintains a certain level of robustness.

7.2.1.2 Data Feature Engineering

Further elaboration is omitted here.

7.2.1.3 Model Training

For the One-Class SVM algorithm, three hyperparameters are particularly critical: kernel, gamma, and nu. The kernel options are “rbf,” “linear,” “poly,” and “sigmoid,” with “rbf” as the default. In a specific anomalous traffic monitoring model, we tested all four kernel parameters. Figure 7.11 shows the decision boundaries learned using different kernels. For datasets with elliptical spatial distributions, the “rbf” kernel yields the best results. The default gamma value is “scale.” Under normal circumstances, gamma is set to its default value.

Next, the nu value requires adjustment. Nu represents the training error; generally, the smaller the training error is set, the stricter the boundary surface obtained through training becomes. In a certain anomalous traffic monitoring model, learning was attempted with $\nu = 0.01$, $\nu = 0.1$, $\nu = 0.2$, and $\nu = 0.5$. The boundary surfaces learned using different nu values are shown in Figure 7.12. Smaller nu values improve the boundary’s inclusiveness for data but also increase the likelihood of misclassifying abnormal traffic as normal traffic. The appropriate nu value must be selected based on the specific business scenario. In this scenario, $\nu=0.1$ is prioritized.

7.2.1.4 Model Validation and Tuning

After training the model, it must be tested on the test set to select the optimal threshold for identifying anomalous traffic. If the model was trained using the One-Class SVM algorithm, the predict function returns 1 and -1, where 1 represents normal traffic and -1 represents anomalous traffic, making it challenging to determine the threshold. If the

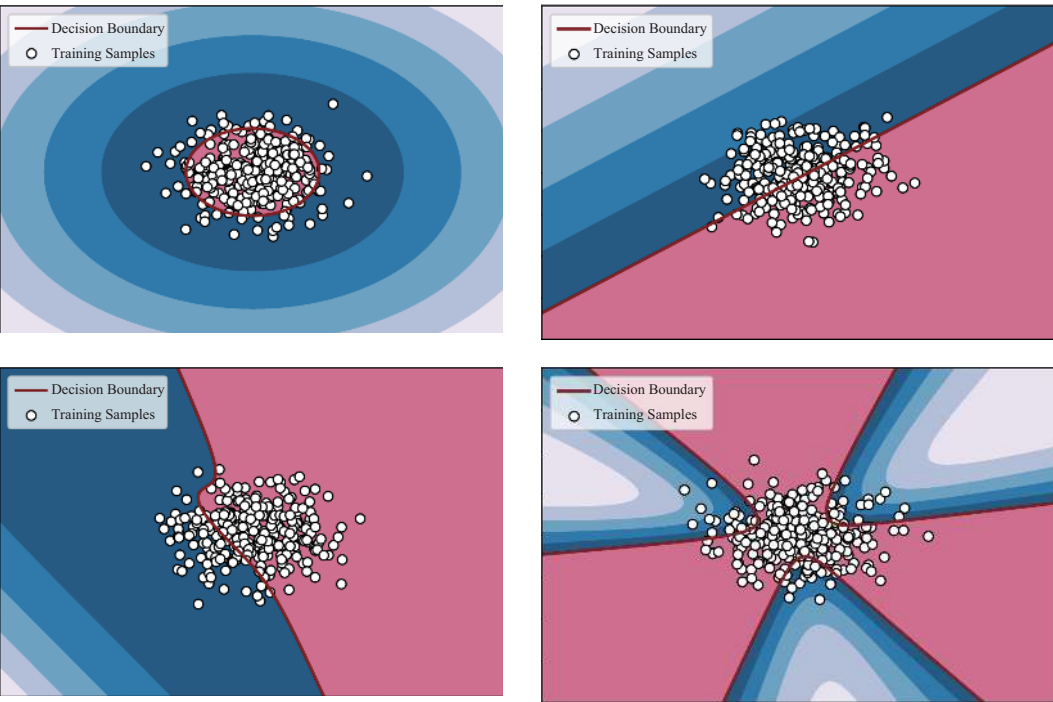


FIGURE 7.11 Boundary surfaces learned using different kernel parameters.

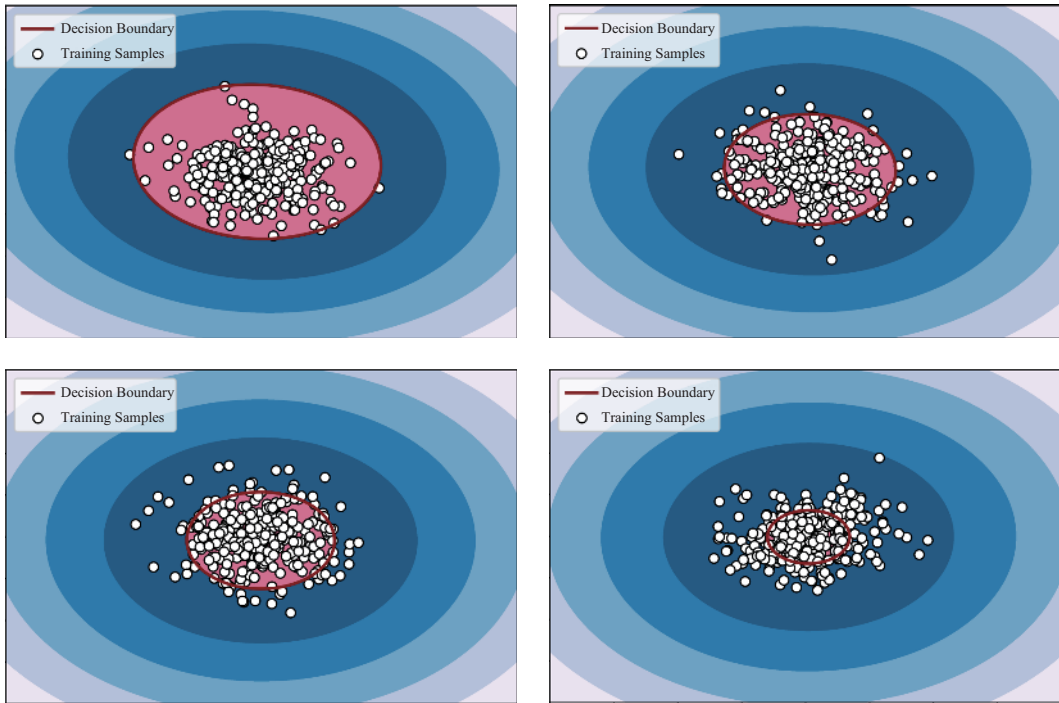


FIGURE 7.12 Boundary surfaces learned using different ν values.

`score_samples` function is used for prediction, it returns the distance of samples to the hyperplane, with a larger distance indicating a higher probability of normal traffic. In anomalous traffic monitoring scenarios, threshold selection must balance the false positive rate (normal traffic misclassified as anomalous traffic) and the coverage rate (anomalous traffic missed). The optimal threshold is chosen by weighing these trade-offs.

7.2.2 Behavioral Sequence Learning Approach

Unlike traditional semi-supervised learning approaches, the behavioral sequence learning scheme is primarily based on the patterns of users' operational behavior sequences, detecting abnormal accounts that perform large-scale, uniform operations—such as group control or scripted actions—by analyzing patterns in user operation sequences. This method offers high interpretability, simplified feature engineering, and low training costs for detecting anomalous traffic behaviors.

While normal user operation sequences are highly variable, black/gray industry groups exhibit consistent patterns due to their targeted objectives and tool-based batch operations. Taking a short-video app as an example, user actions primarily include login, check-in, search, browsing, liking, commenting, video posting, messaging, shopping, and payment settlement. When normal users engage with this app, they typically log in first, followed by more random subsequent actions. In contrast, black/gray industry groups engaged in like and comment boosting have a clear objective upon logging in, leading to a series of repetitive and similar actions such as repeated likes and comments.

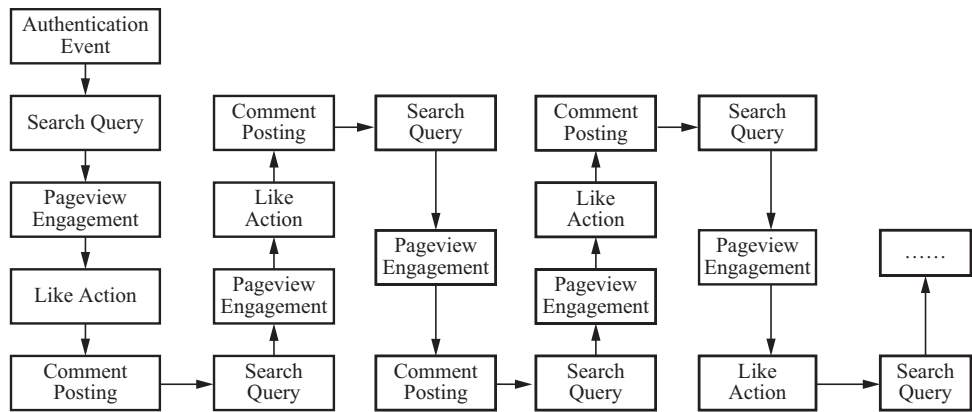


FIGURE 7.13 Behavioral sequence of a brushing account.

The fraud group that offers the interface for a like-boosting service sells services like “double-taps on posts” and “post comments” on their website. After purchasing a service, users submit relevant post information. Upon receiving an order, the bot farm uses a large pool of accounts to automatically search for the specified content, like and comment with pre-prepared material. It then moves on to the next piece of content, repeating the process until the order is fulfilled. Throughout this operation, the bot farm collectively controls numerous devices and accounts. After logging into the app, these accounts continuously repeat the cycle of searching, browsing, liking, and commenting. The behavioral sequence of one such bot account is illustrated in Figure 7.13.

Consequently, fake accounts exhibit repetitive behavioral patterns and group-level similarity in behavioral sequences.

- Behavioral Repetition: For predetermined actions like searching, browsing, liking, and commenting, fraud groups program automated scripts that control accounts to endlessly repeat predefined behavioral paths.
- Group Behavioral Similarity: Accounts under group control by the same bot group exhibit nearly identical behaviors, continuously liking and commenting on target content. Accounts under group control by different bot groups may also show behavioral similarities since their shared objective is to inflate metrics.

7.2.2.1 Principles of Generic Behavioral Sequence Anomaly Detection

After accumulating a large number of normal samples and a small number of abnormal samples, a behavioral sequence model for anomalous traffic can be trained. The process of identifying anomalous traffic based on user behavioral sequences is shown in Figure 7.14, which consists of four steps: behavioral sequence representation, similarity assessment, threshold verification, and online identification.

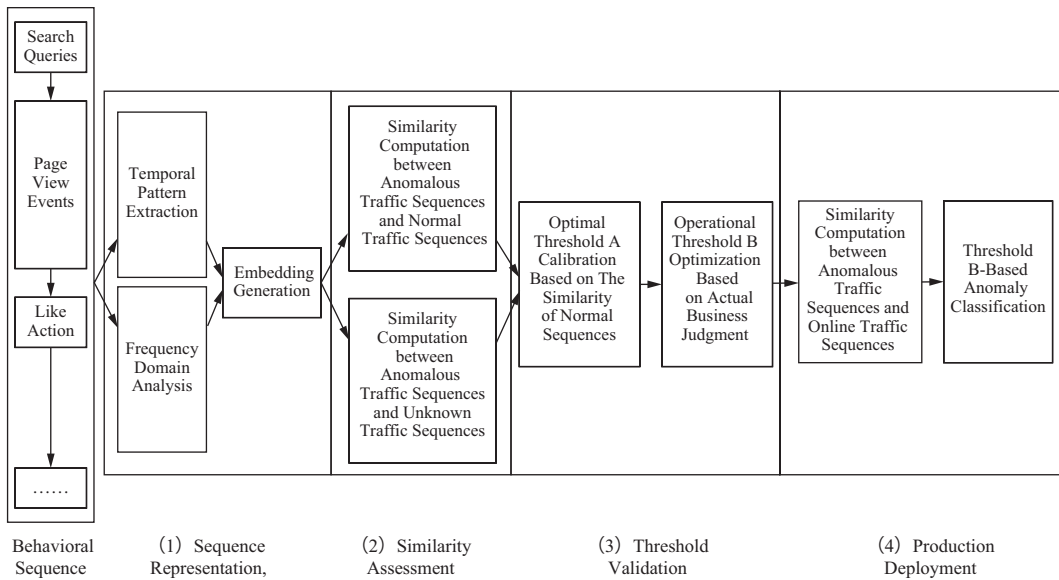


FIGURE 7.14 Process for identifying anomalous traffic based on user behavioral sequences.

7.2.2.1.1 Behavioral Sequence Representation By mining information from users' behavioral sequences within the app, a vector representation of the behavioral sequence is generated. This vector representation can be approached from both time domain and frequency domain perspectives.

7.2.2.1.2 Similarity Assessment Unknown accounts and normal accounts are extracted from a period of online activity, and corresponding behavioral sequences are constructed for each. The ratio of unknown accounts to normal accounts is adjusted to be close to 1:1. The similarity between the behavioral sequences of normal accounts, unknown accounts, and a small number of abnormal account samples is then calculated.

7.2.2.1.3 Threshold Validation By calculating the similarity thresholds between normal and abnormal behavioral sequences, the highest threshold A is obtained (excluding outliers). Unknown accounts are then screened against the similarity threshold of abnormal behavioral sequences. Behavioral sequences with similarity greater than A and exceeding 0.8 (depending on actual business and data) are placed into set S. Accounts in set S undergo further validation through other dimensional user profiling. When training thresholds, the core purpose of using normal user data is to obtain the reference threshold A. If A is too low, it may lose its reference value; if A is sufficiently high, it can assist the business in rapidly converging to a reasonable security threshold.

7.2.2.1.4 Online Detection After training with the above threshold, a threshold B is derived for business monitoring or mitigation. The similarity between known anomalous traffic and live traffic is then calculated, with accounts exceeding B classified as anomalous traffic.

The process described above focuses on the principles and steps for detecting anomalous traffic in behavioral sequences. Various sequence construction and modeling approaches exist across different business scenarios.

Next, we focus on constructing behavioral sequence models using temporal and frequency domain dimensions. First, we define the User Operation Sequence ID. All actions triggered by a user after entering the product are distinguished by a unique ID. For example, login operation: 101; check-in operation: 102; search operation: 103; browse content operation: 104; like operation: 105; comment operation: 106; post video operation: 107; bookmark operation: 108.

7.2.2.2 *Anomaly Detection in the Time Series Dimension*

Bulk operations by actors from the black/gray industry most readily exhibit varying similarity patterns across time series dimensions. Next, we describe the core process of constructing an anomaly detection model for behavioral sequences using a simple similarity calculation method.

7.2.2.2.1 Defining User Time Series Define a time window T , such as one day. Within the T -time range, define all users' behavioral sequences (without merging duplicate actions). For example, User 1's time-ordered behavioral sequence is 101, 102, 105, 104, 108, 106, 104, 107.

7.2.2.2.2 Define the Behavioral Sequence Define the window size N for splitting, for example, $N=4$. For a four-element combination, there are two splitting methods: one is overlapping sliding splits, for example, sequence 1 is 101–102–105–104, then sequence 2 is 102–105–104–108. This approach leverages the concept of maximum set coverage to comprehensively cover all combinations. The second method is non-overlapping partitioning. For example, if sequence 1 is 101–102–105–104, then sequence 2 is 108–106–104–107. This method is suitable when no redundant information exists between different sequences.

7.2.2.2.3 Training Suspicious Sequence Sets By analyzing historical anomalous traffic patterns, a set of suspicious behavioral sequences can be constructed. This involves identifying combinations of behavioral sequences that appear with high probability in black samples but low probability in white samples, and defining these as the suspicious sequence set.

7.2.2.2.4 Anomalous Traffic Identification By matching the number and frequency of sequences within the suspicious sequence set and setting thresholds, anomalous traffic can be identified. The behavioral sequence comparison between normal and abnormal user groups is shown in Figure 7.15.

7.2.2.3 *Frequency Sequence Dimension Anomaly Detection*

As countermeasures evolve, black/gray industry groups mimic normal user operations—such as actively refreshing videos to simulate viewing behavior. When normal behavior is intermixed with actions like searching, liking, or commenting, it becomes difficult to identify consistent patterns solely through temporal behavior analysis. However, since the core

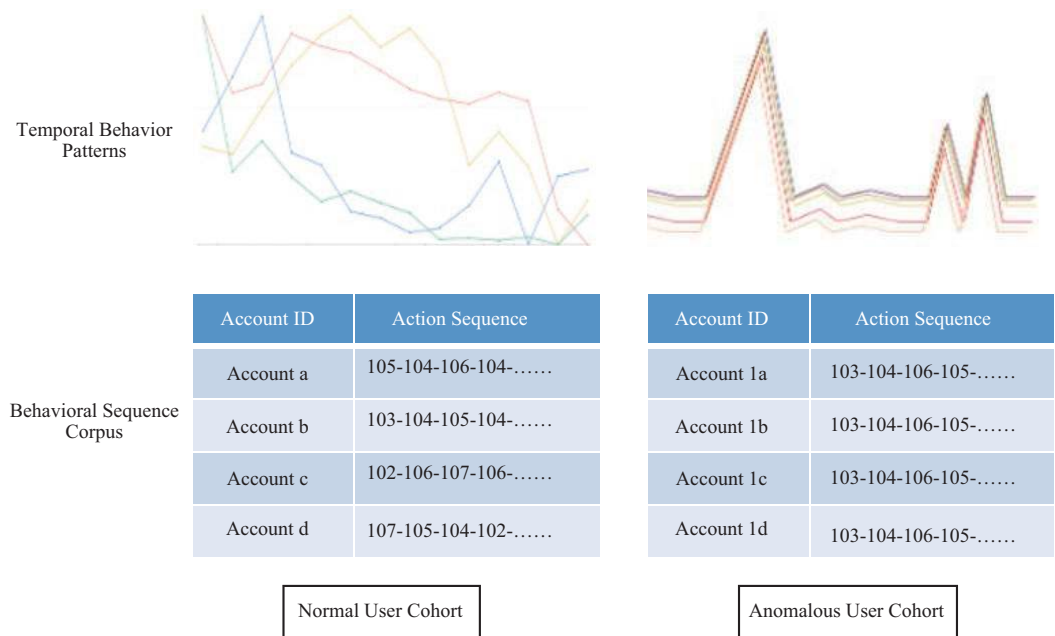


FIGURE 7.15 Comparison of behavioral sequences between normal and abnormal user groups.

objective of these groups remains unchanged—using group control devices to artificially inflate views and likes for target content—their various operational behaviors may exhibit similar frequency patterns.

Next, by converting the behavioral sequence from the time domain to the frequency domain, we obtain a vector representation of the behavioral sequence in the frequency domain space. Then, through similarity calculations, we construct an anomaly detection model for behavioral sequences. The core process is detailed below.

7.2.2.3.1 Defining User Time Series Define a time window T . Within the T -time range, aggregate all user behavioral sequence chronologically into time series. For example, the time-domain behavioral sequence for anomalous User 1 is: 101, 103, 104, and for anomalous User 2: 101, 104, 106.

7.2.2.3.2 Transforming Behavioral Sequences into Frequency-Domain Vector Representations Convert the user’s behavioral sequence into a vector representation in the frequency domain. Each bit of the vector represents an action or a combination of actions, and the numerical value at each bit indicates the frequency of occurrence of that action or the preceding and subsequent consecutive combinations of actions. For example, “search + like” represents performing a search immediately followed by liking. Another example is the vector [38, 158, 76, 65, 112], which represents 38 saves, 158 likes, 76 comments, 65 likes + comments, and 112 searches + likes. Sort all actions in descending order and extract the top N actions, such as the top 20 actions. Define these as fixed-length vectors, where each bit represents a specific action or action combination.

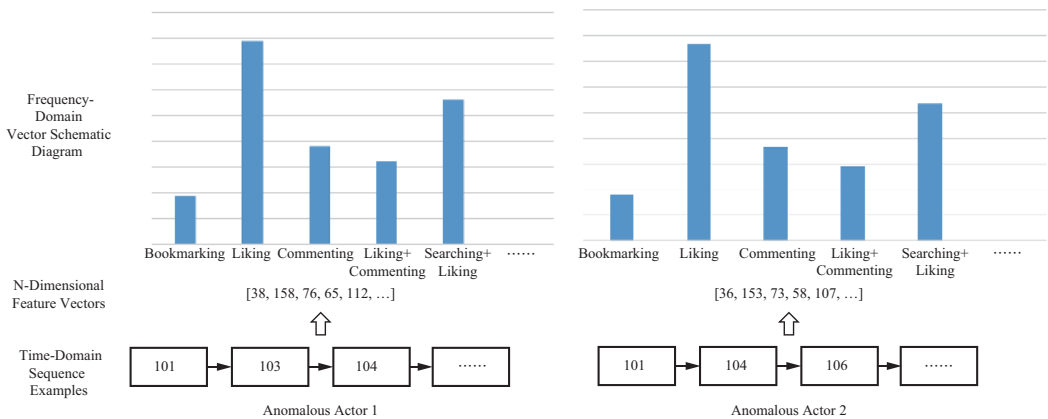


FIGURE 7.16 Frequency domain vector representations of different anomalous behavioral sequences.

7.2.2.3.3 Training Suspicious Sequence Sets By leveraging historically accumulated anomalous traffic, a set of suspicious frequency domain vectors can be constructed. This yields thresholds with high similarity to black samples and low similarity to white samples. After business validation, the final screening thresholds can be established.

7.2.2.3.4 Anomalous Traffic Identification Using thresholds determined by trained abnormal behavioral sequences, a set of anomalous traffic can be filtered out for subsequent monitoring or mitigation. Frequency domain vector representations of different abnormal user behavioral sequences are shown in Figure 7.16.

7.3 MANY-SHOT SCENARIO

As business standards become increasingly refined, definitions grow clearer, and data engineering matures, the battle against black/gray industries intensifies. User complaints and reports also improve, enabling businesses to accumulate substantial abnormal samples. With a sufficient number of normal and abnormal samples available, supervised learning can be fully leveraged.

When applying supervised learning for anomalous traffic detection in practice, model selection and construction are typically considered from the following three dimensions:

- First, decide whether to build a binary classification model or a regression model. This choice primarily depends on the nature of the samples and the modeling scenario, as detailed in Section 7.3.1.
- Second, consider whether the modeling scenario requires interpretability. If interpretable model scoring explanations are needed, interpretable models such as logistic regression should be used for anomalous traffic detection. Section 7.3.2 will provide further details.

- Third, as data continues to accumulate, training ensemble models becomes necessary to further enhance model performance.

Section 7.3.3 will provide a detailed introduction.

7.3.1 Binary Classification Models vs. Regression Models

In supervised learning, the question of model selection frequently arises. This chapter addresses model choice from the perspective of business scenarios and operational experience.

In anomaly traffic detection, when samples include clearly defined positive samples and negative samples—where positive samples represent anomalous traffic and negative samples represent normal traffic, typically labeled with discrete values 1 or 0—in scenarios where quantifying the severity of malicious activity is unnecessary, building a binary classification model is preferred. During training, all anomalous traffic is treated as samples with the same level of maliciousness. However, in specific scenarios requiring quantitative grading of maliciousness, constructing a regression model is preferable. This is because regression model training sets the sample value of negative samples to 0, then quantifies the sample values of positive samples into values between 0 and 1 based on maliciousness—higher values indicating stronger maliciousness. Compared to binary classification models, regression models typically employ mean squared error as their loss function. This allows them to learn distinctions between anomalous traffic samples across varying degrees of maliciousness. Their predicted values directly represent maliciousness levels, with higher predictions indicating more severe anomalies. A comparison between binary classification and regression models is illustrated in Figure 7.17.

7.3.2 Explainability Scenarios

When constructing models to detect anomalous traffic, business stakeholders often require interpretable discrimination—meaning they need to understand the reasons for classification decisions from a business perspective. For example, when a user’s behavior is

Binary Classification Model	Regression Model
☐ Samples: Positive Samples /Negative Samples	☐ Samples: Continuous-valued samples (range [0,1])
☐ Loss Function: Cross-entropy loss (log loss)	☐ Loss Function: Mean Squared Error (MSE)
☐ Use Cases: Unable to Grade Positive Samples, as The Prediction Results Represent Probabilities of Belonging to The Positive Class	☐ Use Cases: Quantifiable grading is possible, and higher prediction values indicate more malicious anomalous traffic

FIGURE 7.17 Comparison of binary classification and regression models.

classified as “freeloading,” the model should provide an explanation for why that behavior was labeled as such. In such cases, the logistic regression algorithm is used. Logistic regression trains parameters $\theta = [\theta_1, \theta_2, \theta_3, \dots, \theta_N]$, then applies the following logistic function for nonlinear mapping to calculate the probability that the sample traffic is anomalous traffic.

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

By examining the magnitude of the training parameter θ , we can determine whether the relationship between the N features and the label is positive or negative. If θ_i yields a positive value, it indicates a positive contribution between the i feature and the label, with a larger value signifying a stronger positive relationship. Conversely, if θ_i yields a negative value, it indicates a negative contribution between the i feature and the label, with a smaller value signifying a stronger negative relationship.

The overall process of constructing an anomalous traffic detection model based on logistic regression is shown in Figure 7.18.

7.3.2.1 Data Preparation

The first step is the data preparation phase. For an introduction to profiling data, refer to Section 7.1.2. Unlike the unsupervised learning approach in Section 7.1.2, the supervised learning approach requires preparing samples and their corresponding labels. Typically, target samples (i.e., anomalous traffic samples) are treated as positive samples, while normal traffic samples are treated as negative samples. Anomalous traffic samples represent the specific types of anomalous traffic that need to be intercepted. For detecting “freeloading” behavior, historical anomalous traffic associated with obtaining activity benefits is used as positive samples. For negative samples, based on the author’s years of experience combating the black/gray industry, randomly extracting normal traffic as negative samples is not the most appropriate approach. Figure 7.19 illustrates the initial sample construction approach, where positive samples and negative samples exhibit significant differences. During model training, the loss function value decreases rapidly, and the test set area under the curve (AUC) is very high. However, the model trained at this stage is not optimal. The straight line in Figure 7.19 represents the decision boundary, which is a hyperplane in high-dimensional space. Figure 7.19 shows three decision boundaries, each capable of completely separating positive samples and negative samples. However, such



FIGURE 7.18 Overall process for constructing an anomalous traffic detection model based on logistic regression.

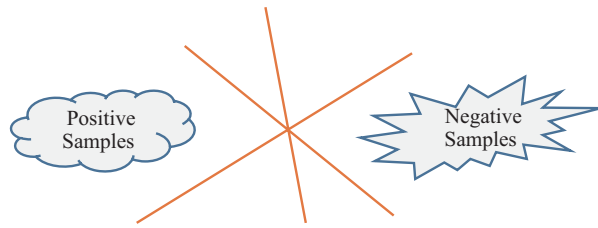


FIGURE 7.19 Initial sample construction approach.

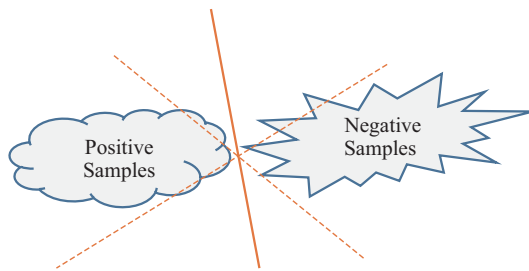


FIGURE 7.20 Adjusted sample construction method.

perfect decision boundaries rarely exist in real-world scenarios. Models trained using this sample construction method exhibit poor practicality in actual applications.

Therefore, during the sample screening phase, it is necessary to first eliminate obviously normal traffic. After all, completely normal traffic does not require model classification and can be identified through simple strategies. Following this screening, the positive and negative samples are found to be distributed relatively close together in the feature space. The adjusted sample construction method is shown in Figure 7.20. After the positive and negative samples are brought closer together, the two decision boundaries marked with dashed lines become difficult to use for sample segmentation. This allows for training a more ideal and effective decision boundary at the sample level.

7.3.2.2 Data Feature Engineering

The second step involves data feature engineering. Further details are omitted here.

7.3.2.3 Model Training

The third step involves training the model. This process requires algorithm engineers to optimize the model's performance by adjusting hyperparameter values. Different algorithmic models have distinct hyperparameters. The following section uses the logistic regression algorithm as an example to illustrate hyperparameter tuning.

Logistic regression involves numerous hyperparameters, but three aspects primarily impact training outcomes: First, adjusting algorithm iteration parameters—mainly setting the learning rate and training termination conditions. Second, modifying the loss function—primarily by incorporating regularization terms to find models that minimize

prediction error while maintaining simplicity. Third, selecting an optimization algorithm—primarily choosing an optimizer to refine the loss function and ensure effective model training. Next, we will discuss how to adjust model training hyperparameters based on practical experience from anomalous traffic detection models.

Selecting appropriate hyperparameters involves evaluating model performance. To thoroughly validate the data, k -fold cross-validation can be employed to assess model effectiveness. For k -fold cross-validation, the same set of hyperparameters yields k validation datasets. Across these k datasets, k AUC values are obtained, where AUC represents the area under the ROC curve. Figure 7.21 displays the ROC curve for an anomalous traffic detection model, with an AUC value of 0.75. The final evaluation calculates the mean and standard deviation of the AUC across these k validation datasets.

Based on the performance of the mean and standard deviation of AUC across the k validation datasets, the corresponding hyperparameter tuning measures can be summarized into the following three points:

- When the mean AUC on the test set is low and below expectations, the primary hypothesis is that the model may be underfitting. In this case, attempts can be made to increase the learning rate or extend the number of training iterations. If these adjustments effectively raise the mean AUC, it indicates that the previous model was underfitting and did not learn the data sufficiently.
- If the mean AUC on the training set continues to rise during training iterations while the mean AUC on the test set first increases and then decreases, this indicates

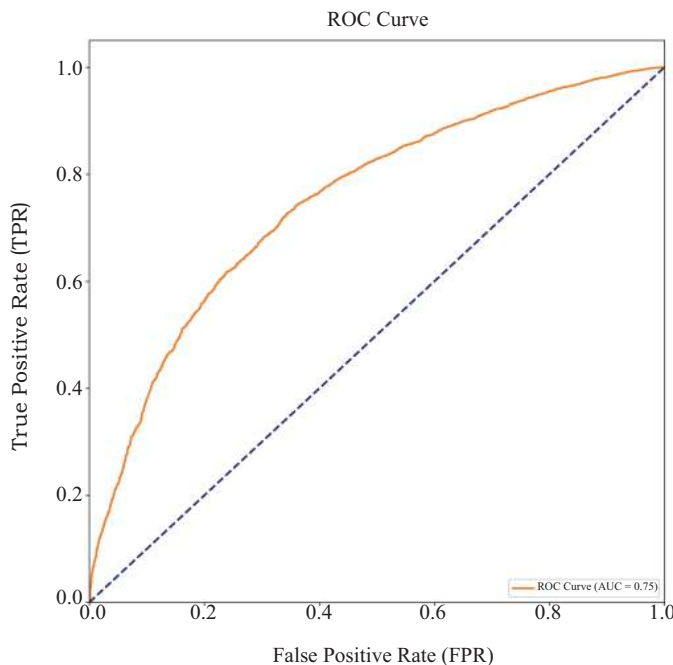


FIGURE 7.21 ROC curve of an anomalous traffic detection model.

overfitting on the training set. Adjust the model termination condition, such as reducing the maximum iteration count to stop training earlier.

- A large mean and variance in the test set AUC indicate significant performance variation across different folds of the training dataset. This strongly suggests overfitting during model training. One approach is to incorporate a regularization penalty term into the objective function. If a regularization penalty already exists, consider increasing its weight. Alternatively, this could stem from insufficient sample size, inadequate data coverage, or uneven sample distribution. In such cases, hyperparameter tuning is typically not the first solution. Instead, prioritize increasing the sample size or expanding the data collection scope to achieve a more uniform distribution of samples across the feature space.

7.3.2.4 Model Validation

After training the model, it must undergo validation. As mentioned earlier, the logistic regression algorithm is employed in scenarios requiring interpretability, where the trained model parameters θ serve as an explanatory basis. When the parameter θ yields a positive value, it indicates that the corresponding feature value contributes positively to label prediction. At this point, it is necessary to verify whether the relationship between this feature and the label aligns with business logic. If it is consistent with business logic, the magnitude of the feature value explains its contribution to model performance. For variables inconsistent with business logic, further analysis is required—such as examining whether correlations exist between feature variables that cause mutual cancellation—to determine if additional feature processing is necessary.

7.3.2.5 Model Deployment

Once model validation is complete, the next step is to deploy the model for use while ensuring proper operational management. After deployment, it is essential to continuously monitor its performance in traffic classification and adjust the model as necessary. For detailed guidance on this process, refer to Chapter 11 of this book.

7.3.3 Ensemble Models

Section 7.3.2 introduced logistic regression models for interpretable scenarios, though other similar models are available. While these models offer strong interpretability and straightforward business understanding, they struggle with complex adversarial scenarios. For instance, in later stages of adversarial attacks, the black/gray industry may disguise its behavior across traffic stages to mimic normal users. In such cases, ensemble models can enhance overall recall.

Three common ensemble approaches are adaptive boosting, bagging, and stacking. All three use simple models as base learners and then integrate multiple base learners to form ensemble models with stronger fitting capabilities. These approaches differ in their underlying principles and methodologies. Table 7.3 compares the algorithms of adaptive boosting, bagging, and stacking: Adaptive boosting trains each base learner iteratively based on

TABLE 7.3 Comparison of Adaptive Boosting, Self-Voting, and Stacking Algorithms

	Adaptive Boosting	Self-Voting	Stacking
Base learner requirements	Base learners' training dependencies	Independent training of individual base learners	Primary learner training independence
Training set	Dependencies	Independent	Independent
Algorithm framework	Serial structure	Parallel structure	Primary learners run in parallel, while secondary learners run in series
Representative algorithms	XGBoost, GBDT, etc.	Random forest	An algorithmic framework

the previous one's results, with the final decision made collectively; self-voting involves training each base learner independently and then aggregating their outputs through voting or weighting to determine the final decision; self-voting shares similarities with stacking in that it also employs independent training, but it further trains a secondary learner using the predictions from the primary learners as features, ultimately using the secondary learner's prediction as the final output.

This section focuses on XGBoost as an example to illustrate its practical application in anomalous traffic detection.

In 2016, Dr. Tianqi Chen introduced the XGBoost algorithm in his paper “*XGBoost: A Scalable Tree Boosting System*.” The XGBoost algorithm has achieved strong results in major data competitions and delivers effective outcomes in routine modeling, making it widely adopted. The process of building a malicious traffic monitoring model using XGBoost largely follows the overall workflow depicted in Figure 7.18. The primary differences lie in feature engineering and model hyperparameter tuning.

- Regarding feature engineering, since XGBoost's base learners utilize tree models, feature normalization or extreme value removal is unnecessary. However, outlier removal remains essential. Additionally, when converting non-numeric features to numerical values, one-shot encoding is not recommended. This is because high-dimensional features can become overly sparse, necessitating deeper tree models. Embedding encoding is thus preferred for better control over feature dimension and sparsity.
- For adjusting model hyperparameters, the XGBoost algorithm has three important hyperparameters: `max_depth`, `n_estimators`, and `scale_pos_weight`. `max_depth` represents the maximum depth of the tree. A deeper tree enhances fitting capability but also increases the risk of overfitting. `n_estimators` denotes the number of trees; more trees improve fitting capability. `scale_pos_weight` represents the weight assigned to positive samples. A higher value increases the weight of positive samples, leading to a greater increase in the loss function when samples are misclassified. Adjusting sample weights can also accelerate algorithm convergence and promote a more balanced overall scoring distribution.

During XGBoost model training, pay close attention to the following three aspects.

7.3.3.1 Adjusting the `scale_pos_weight` Parameter

In anomalous traffic detection scenarios, positive samples (anomalous samples) are typically far fewer than negative samples (normal samples). For example, when training a traffic anomaly detection model with a positive-to-negative sample ratio of 1:4, `scale_pos_weight` defaults to 1. When predicting the proportion of malicious traffic distributed across different ranges on the test set (with a positive-to-negative ratio of 1:4), as shown in Figure 7.22, the overall score distribution is skewed toward 0–0.5, with the majority concentrated in the 0–0.2 range (accounting for 56.5%). This distribution is unreasonable because the training set contains fewer positive samples, leading to overall training bias toward negative samples and a tendency to predict low scores. After setting `scale_pos_weight` to 4 and retraining the model, we predicted the proportion of traffic maliciousness scores across different intervals on the test set (with a positive-to-negative ratio also set to 1:4), as shown in Figure 7.23. The overall score distribution appears more balanced, with a noticeable increase in the proportion of predicted scores within the 0.5–1 interval.

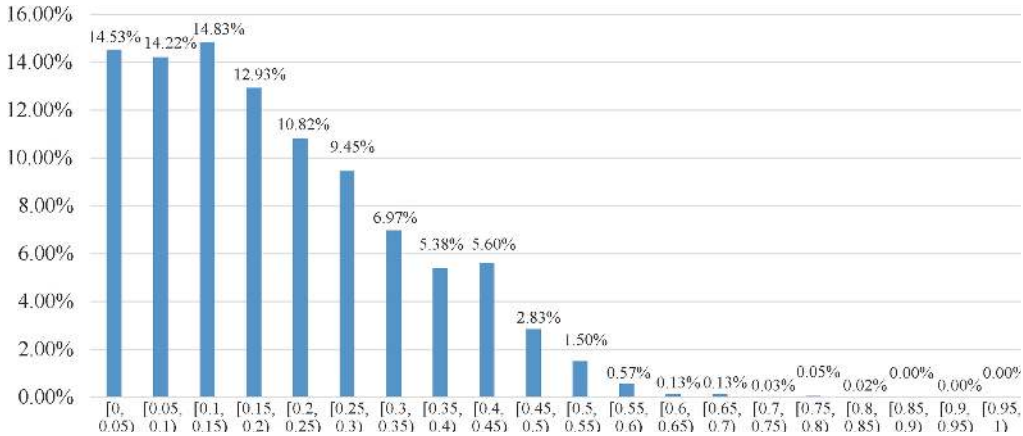


FIGURE 7.22 Proportion of predicted traffic maliciousness across different intervals (`scale_pos_weight` set to 1).

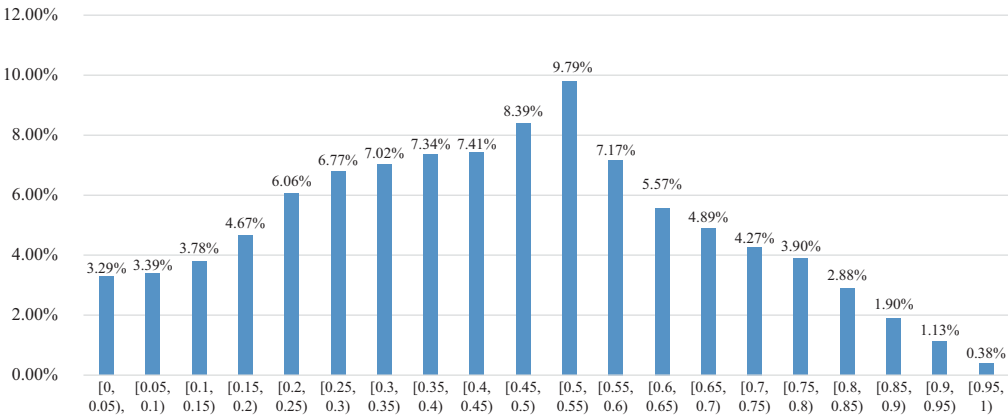


FIGURE 7.23 Proportion of predicted traffic maliciousness across different scale intervals (`scale_pos_weight` set to 4).

7.3.3.2 Adjusting the Parameters max_depth and n_estimators

To optimize the parameters max_depth and n_estimators, cross-validation can be employed to identify their optimal values. First, adjust max_depth. When constructing models for anomalous traffic detection, this parameter typically ranges from 3 to 10, with higher values indicating stronger fitting capabilities. The cross-validation results for max_depth=3, max_depth=5, and max_depth=8 are shown in Figure 7.24. When max_depth=3, the average test AUC for the first tree is 0.66, and the average test AUC after training 21 trees is 0.74. When max_depth=5, the average test AUC is 0.68 for the first tree and 0.75 after training up to the 21st tree. This demonstrates that max_depth=5 yields better results than max_depth=3. However, while max_depth=8 shows a slight overall improvement in average test AUC compared to max_depth=5, its average training AUC reaches 0.85 at the 21st tree—significantly diverging from the test AUC. Additionally, the standard deviation of the test AUC increases. These comparisons indicate that overfitting occurs when max_depth is set to 8, confirming that max_depth=5 yields the best results. Next, we adjust the n_estimators value. After confirming the optimal max_depth, we observe how the average test AUC changes as the number of trees increases. Generally, once the average test AUC stabilizes to within three decimal places, the number of trees reached at that point can be set as the n_estimators value.

7.3.3.3 Feature Importance Evaluation

After training the model built with the XGBoost algorithm, it is first necessary to evaluate and rank the feature importance. Subsequently, we verify whether the features with higher importance rankings align with business logic. Figure 7.25 displays the feature importance ranking for an anomalous traffic detection model, indicating the number of times each feature was used in splitting decisions across subtree models. As shown in Figure 7.25, feature “F55” ranks highest in importance. Validation confirms this feature correlates with group control device profiles, aligning with business expectations.

Compared to simple single-classification models, using the XGBoost algorithm to build anomalous traffic detection models significantly improves the F1-score. To further enhance coverage of traffic fraud, stacking techniques can be employed to boost model performance. Table 7.3 does not list representative stacking algorithms, as stacking is both a methodology and an algorithmic integration framework.

Next, we focus on how to enhance the effectiveness of models for detecting anomalous traffic based on the stacking algorithmic concept.

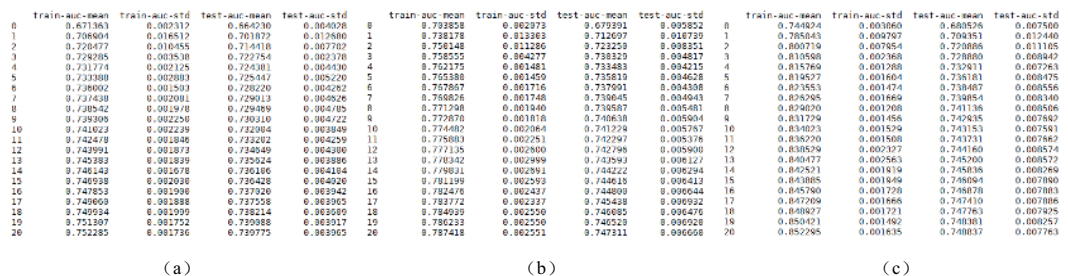


FIGURE 7.24 Cross-validation results for max_depth=3, max_depth=5, and max_depth=8.

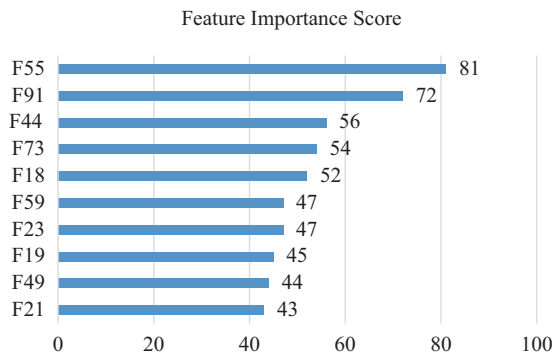


FIGURE 7.25 Feature importance ranking for an anomalous traffic detection model.

When constructing models for anomalous traffic detection using the XGBoost algorithm, the algorithm selects effective features for model building. The remaining features are not selected by the algorithm due to their less pronounced discriminative power. However, these weaker features still possess some discriminative capability for the samples. Continued exploration of these weaker features can further enhance the algorithm's effectiveness.

The stacking-based approach to anomalous traffic detection model construction involves categorizing features by attributes and building primary learners using features from different attribute groups. Training distinct primary decision models yields varying prediction scores. These scores can be interpreted as evaluating the probability of a sample being anomalous traffic from multiple dimensions. Simultaneously, these scores represent aggregated strong features derived from weak feature mining. These prediction scores are then fed as feature values into secondary learners to train secondary decision models. The output of these secondary models serves as the final traffic classification result. Therefore, stacking can be used to uncover weak features, better utilize data, and enhance overall model performance.

The overall framework of the stacked anomalous traffic detection model is shown in Figure 7.26, comprising M primary classification models and one secondary classification model. These M primary classification models are based on features ranging from “Feature 11 to Feature 1N” and “Feature 21 to Feature 2N”. When constructing an anomalous traffic detection model, “Features 11 to 1N” represent device-level characteristics, “Features 21 to 2N” represent IP-based evasion techniques, while “Features 31 to 3N” represent account-level features. Other features are fed into additional primary discriminant models. This approach ensures separate and sufficient learning from device, IP-based evasion techniques, account, and other dimensions. The discriminant values obtained from these submodels are then used as strong features to train the secondary discriminant model. The secondary discriminant model is trained using sample set 2. It is crucial to ensure no overlap between sample set 1 and sample set 2, as overlap could lead to label leakage, thereby compromising the training effectiveness of the secondary discriminant model. Finally, the discriminant values produced by the secondary discriminant model serve as the output of this anomalous traffic detection model to identify whether traffic is anomalous.

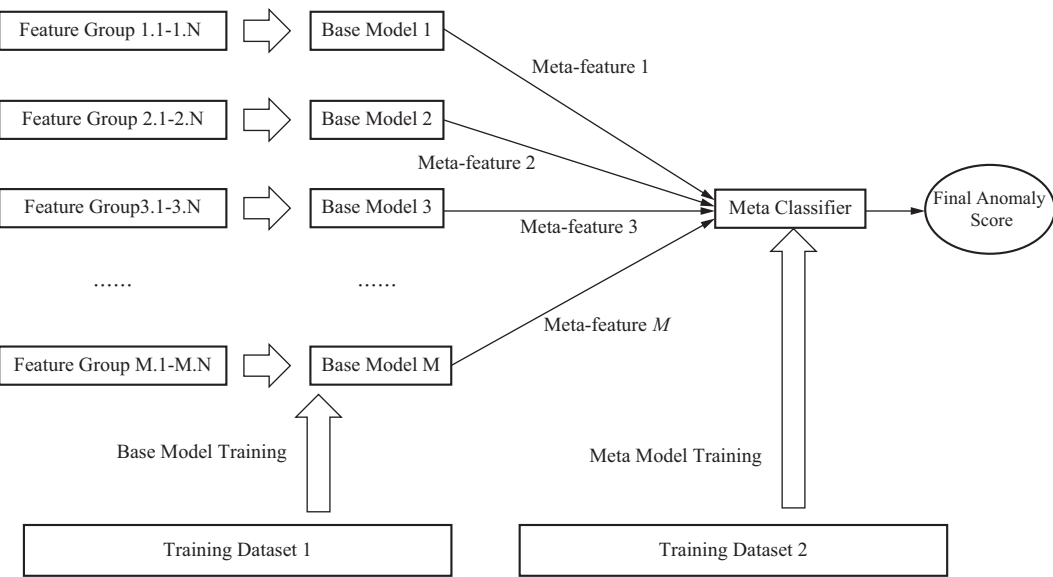


FIGURE 7.26 Overall framework of the stacked anomalous traffic detection model.

TABLE 7.4 Comparison of AUC on the Test Set across Different Modeling Approaches

	Direct Modeling Using the XGBoost Algorithm	Model Built Using Stacking
AUC on test set	0.742	0.772

When selecting modeling approaches for primary and secondary discriminant models, stacking-based models generally outperform those built directly using the XGBoost algorithm. In a specific anomalous traffic monitoring scenario, both modeling approaches were applied, and their AUC comparisons on the test set are shown in Table 7.4. The comparison reveals that the AUC on the test set for the model built directly using the XGBoost algorithm is 0.742, while the AUC for the stacking-based model is 0.772. This indicates that the stacking-based model achieves a 3.0% improvement in AUC compared to the direct XGBoost model. In practical model evaluation, it is important to consider both the AUC metric and the upper bound dimension. When a certain accuracy threshold is reached (e.g., 99%), the proportion of improvement in the recall rate for anomalous traffic should be examined.

7.4 SUMMARY

This chapter uses the anomalous traffic detection scenario as a case study to introduce different machine learning adversarial approaches adopted over various time periods. In the zero-shot scenario, an unsupervised learning approach is employed to build the anomalous traffic detection model; in the few-shot scenario, a semi-supervised learning approach or behavioral sequence approach is utilized to develop the anomalous traffic detection model; in the many-shot scenario, a supervised learning approach is applied to create the anomalous traffic detection model.

Complex Network Adversarial Approaches

With the rapid development and increasing sophistication of the black/gray industry, traffic fraud has evolved from initial “individual operations” to organized, division-of-labor “gang operations,” becoming more covert and rapidly changing. Consequently, the single-point countermeasures discussed in previous chapters face significant challenges, necessitating countermeasures at the group level through complex networks. The core advantages of complex network countermeasures are as follows:

- **Early Detection:** When individual fraud traces are subtle, known fraudulent neighbor nodes within complex networks can be identified through coloring techniques, enabling early detection.
- **Enhanced Coverage:** When single-point fraud traces are evident, complex network-based gang detection provides more comprehensive coverage.

As shown in Figure 1.7, traffic fraud issues and characteristics vary across different stages of the traffic lifecycle, with distinct patterns of black/gray industry data traces emerging at each phase. Consequently, distinct complex network types can be constructed for fraud detection at different stages, primarily categorized into the following three archetypes:

- **Pre-Traffic Phase:** A complex network type based on shared resources among black/gray industries.
- **Mid-Traffic Phase:** Building upon the pre-traffic phase, complex network types can be constructed based on shared behavioral paths and attributes among black/gray industries.
- **Post-Traffic Phase:** Building on the pre-traffic phase and mid-traffic phase, complex network types can be constructed based on shared contacts among black/gray industries.

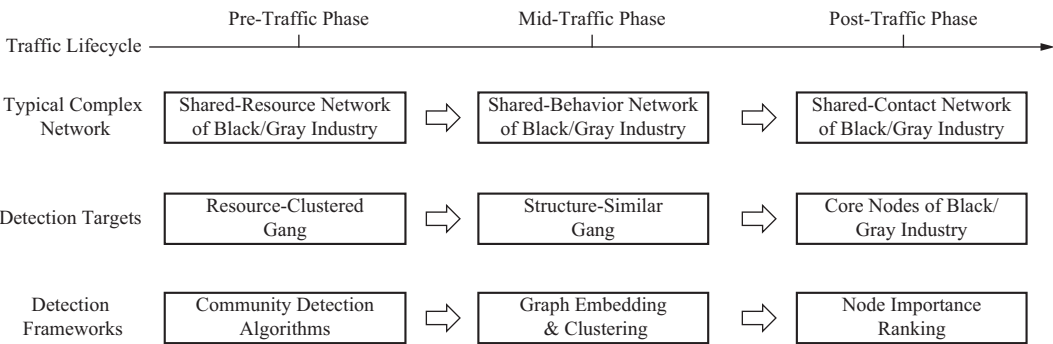


FIGURE 8.1 Evolution of complex network countermeasure schemes.

The evolution of complex network countermeasures across different traffic lifecycle stages is illustrated in Figure 8.1. During the pre-traffic phase, community detection algorithms can identify resource-aggregating black/gray industry groups within the shared-resource complex network formed by traffic fraud activities. In the mid-traffic phase, graph representation and clustering detection frameworks can detect structurally similar black/gray industry groups within the shared-behavior path and attribute complex network. In the post-traffic phase, for the shared contact networks formed by black/gray industry fraud, node importance evaluation detection frameworks can identify core nodes within these networks and propagate detection to fraudulent user nodes.

This chapter focuses on illustrating the evolution of complex network countermeasures throughout the entire traffic lifecycle within the business context of traffic fraud.

8.1 PRE-TRAFFIC PHASE SOLUTIONS

During the pre-traffic phase, black/gray industries are just beginning download, registration, and login operations and have not yet entered platform activities. The data generated at this stage is extremely limited. However, due to the finite nature of essential resources required by black/gray industries, the data generated during this phase can be used to construct a complex network of shared resources among these industries. Its structure, as shown in Figure 8.2, primarily involves the sharing of essential resources such as accounts, IP addresses, devices, phone numbers, and email addresses.

The core elements of network construction involve establishing reliable nodes and high-confidence edges. For common types of traffic fraud, the information for network nodes and edges that can be constructed in the pre-traffic phase is as follows.

1. Network Nodes

- **Account Node:** A unique ID that identifies users on the business platform, typically obtained during registration by linking personal information such as phone numbers and email addresses. Users must then log in to the business platform using this account on their personal devices to begin activities.

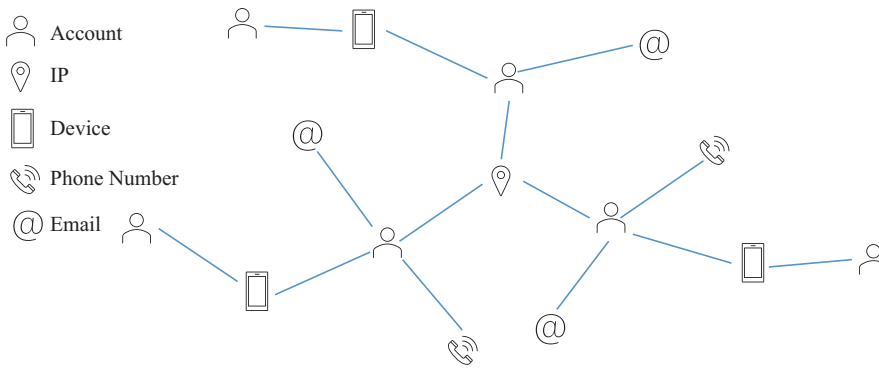


FIGURE 8.2 Complex network structure of shared resources in black/gray industries.

- **IP Nodes:** The IP address of the network environment used when accessing the business platform, which can be a base station IP, Wi-Fi IP, server IP, etc.
- **Device Node:** The device used by the user to log into their account, such as a mobile phone, PC, or tablet.
- **Number Node:** The phone number linked to the account during registration.
- **Email Node:** The email address associated with the user's account during registration.

2. Network Edge

- **Account-Phone Number:** A single phone number can only be linked to one account on the same service platform, but it can be registered and linked to multiple accounts across different platforms. Therefore, the degree of the same service phone number node is 1, while the degree of cross-platform service phone number nodes can be greater than 1.
- **Account-Email:** As personal information linked during account registration, the same email address can only be associated with one account on the same service platform, but it can register and link to multiple accounts across platforms.
- **Account-Device:** A single account can log in via multiple devices, and a single device can log in to multiple different accounts. To reduce costs, black/gray industry operators utilize techniques like multi-instance operation to control devices for batch logging into multiple accounts and simultaneous operations.
- **Account-IP:** A single account can switch between multiple IP addresses for login, and multiple accounts can log in from the same IP address. Since accounts controlled in bulk by black/gray industries share the same network environment, IP clustering occurs.

Based on the complex network of shared resources within black/gray industries, we will now construct subgraphs according to actual business scenarios to detect resource-aggregating criminal groups.

8.1.1 Detecting Syndicates based on Single-Dimensional Resource Aggregation

The cost of fraud is a critical factor determining whether black/gray industries can maximize profits in traffic fraud. To reduce fraud costs, black/gray industries often control a pool of high-risk devices to conduct business fraud through bulk account operations. In this process, legitimate users exhibit relatively dispersed resource usage, while black/gray industry groups tend to cluster the use of various resources such as black/gray industry IPs, devices, and phone numbers. The following section illustrates the detection of account groups based on the clustering of single-dimensional resources from the black/gray industry.

IP addresses are essential yet limited resources. Consequently, the black/gray industry typically operates multiple devices and accounts under the same IP address for fraudulent activities, forming complex networks of shared IP resources, as illustrated in Figure 8.3.

Detection Frameworks for Black/Gray Industry Account Syndicates Based on IP Aggregation: The core steps include the following four phases:

1. Node and Edge Pruning

- Time Range Pruning. IP allocation is dynamic; an IP assigned to Region A in the morning may be reassigned to Region B in the afternoon. Selecting an overly broad time span for node data can introduce excessive noise. Given that the required number of nodes and edges for graph construction cannot be too small, the time range is set to approximately N hours. However, for proxy IPs, longer-term aggregated data may be selected.
- Pruning based on Node Attributes. Given the wide range of base station IPs, which involve usage by numerous users within a region, including these can introduce a large number of normal users that distort the network visualization.

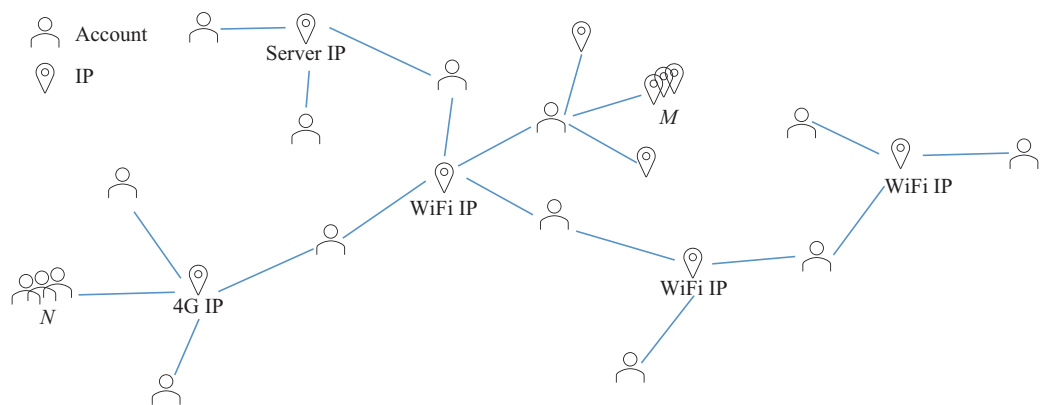


FIGURE 8.3 Complex network of shared IP resources in the black/gray industry.

Therefore, large IP nodes such as base station IPs and enterprise Wi-Fi IPs need to be excluded. Additionally, from an account perspective, nodes associated with a large number of IPs should be removed during the early stages of visualization to prevent the emergence of anomalous communities in the final segmentation results.

The core objective of pruning is to retain high-confidence edges, with confidence thresholds defined by varying business standards. The pruned complex network of shared IP resources for black/gray industries is illustrated in Figure 8.4.

2. Transforming Heterogeneous Graphs into Homogeneous Graphs

Due to the dynamic nature of IP allocation, even after the initial pruning step based on time range—which shortens the time span of IP nodes to approximately N hours—it remains challenging to avoid short-term IP fluctuations. Consequently, the confidence level that IP-aggregated account clusters represent actual criminal groups is low.

Assuming two accounts belong to the same real-world group, regardless of IP switching, these accounts will frequently appear under the same IP. Therefore, the number of times two accounts appear under the same IP can serve as a confidence metric for their belonging to the same real-world group—the more often they

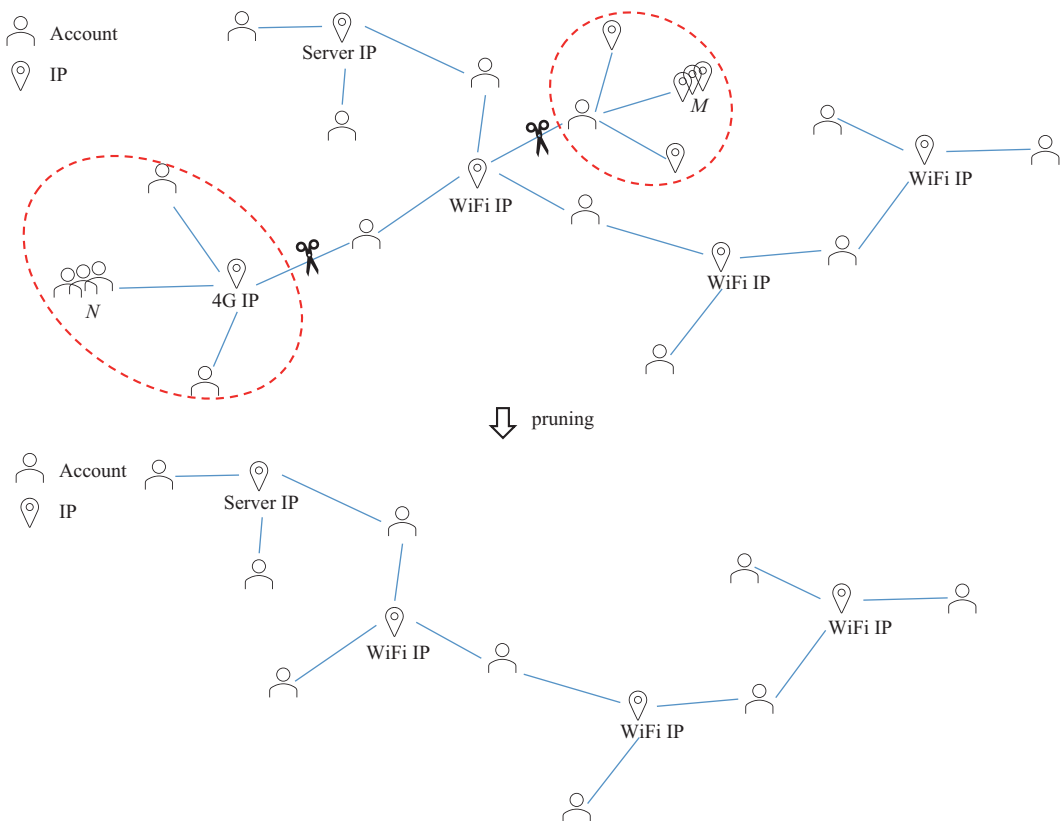


FIGURE 8.4 Complex network of shared IP resources in the black/gray industry after pruning.

co-occur, the higher the confidence. Based on this principle, for accounts sharing the same IP within a time slice, virtual edges can be constructed between each pair. The frequency of these virtual edges appearing within the past N hours serves as their corresponding confidence weight. This transforms the IP-account heterogeneous graph into an account-account homogeneous graph, as shown in Figure 8.5. The upper part of Figure 8.5 shows the heterogeneous graph before conversion, while the lower part displays the homogeneous graph after conversion, where the edge values represent confidence weights. Building upon the principle of converting heterogeneous graphs to homogeneous ones in Step 2, the data range of N hours can be extended to N days in practical applications.

3. Identifying Syndicates Using Community Detection Algorithms

The converted (account, account, confidence) edges are input into the community detection algorithms. After several rounds of iteration, the results converge, ultimately producing the gang partitioning results, as shown in Figure 8.6.

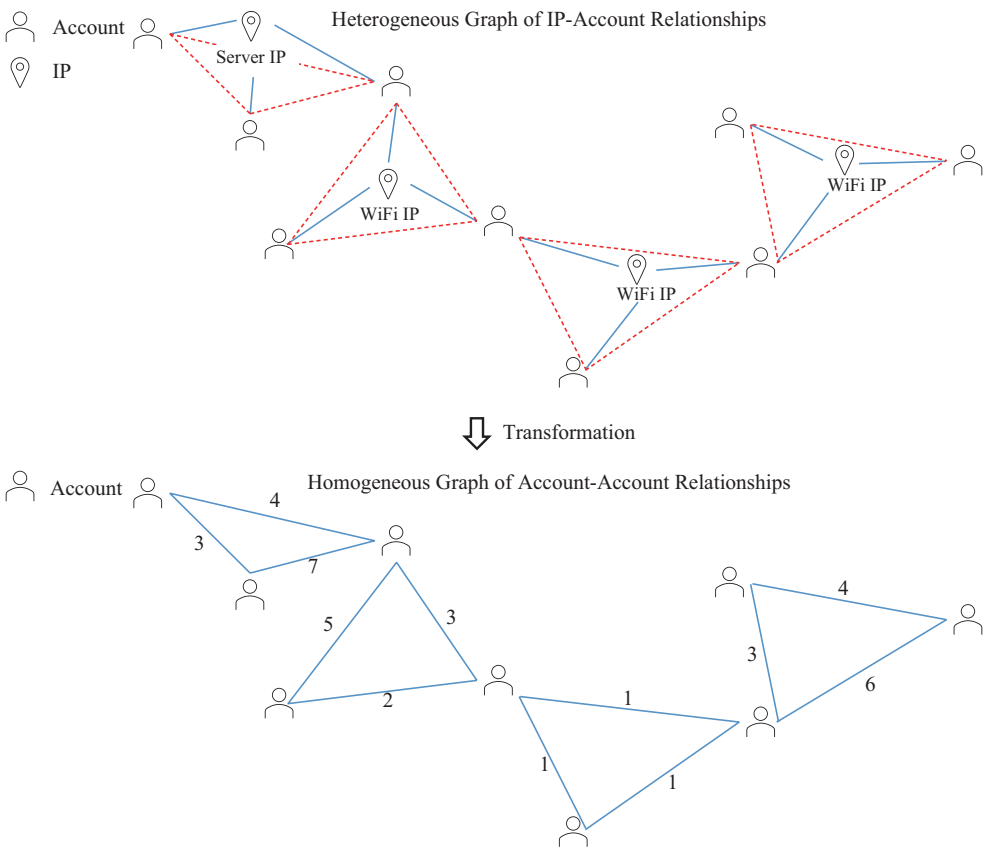


FIGURE 8.5 Conversion of heterogeneous graph between IP and accounts to homogeneous graph between accounts.

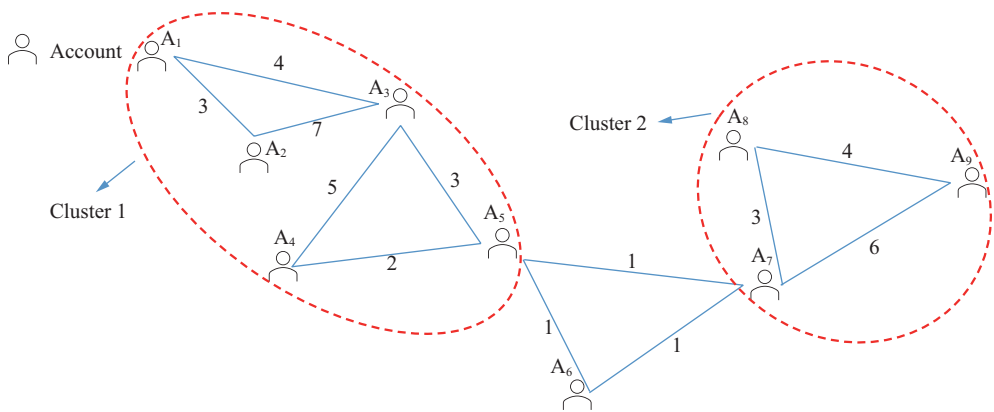


FIGURE 8.6 Gang partitioning results.

TABLE 8.1 Calculated Attribute Values for Groups

Group ID	Group Size	Number of Fraud-Labeled Nodes	Blacklist Ratio	Is Malicious Gang
1	5	3	0.6	Yes
2	3	0	0	No

4. Screening Malicious Account Groups Aggregating IP Resources

The gang partitioning results obtained through the community detection algorithms in the preceding steps reveal which gangs each node belongs to but do not indicate whether the gang is malicious. Therefore, further screening is required based on gang attribute values (such as the proportion of blacklisted and whitelisted nodes within the gang). Assuming fraud labels are known for account nodes A1, A2, and A3, while all other account nodes are normal, the calculated gang attribute values are shown in Table 8.1. By verifying the basic attributes and behavioral data of unknown nodes within gangs with high blacklist ratios, automated convergence rules can be further refined. Ultimately, based on a blacklist ratio of ≥ 0.5 , Group 1 is identified as a malicious account group aggregating IP resources.

The detection of account groups aggregating other single-dimensional resources such as black/gray industry devices, numbers, and email addresses follows a process similar to IP detection. The key consideration is developing a reasonable pruning strategy aligned with business standards.

8.1.2 Detection of Groups Aggregating Multi-Dimensional Resources

The above primarily addresses account group detection for single-dimensional resources of the black/gray industry. When encountering multiple resources of the black/gray industry, the complex network exhibits more pronounced heterogeneous graph properties. Converting heterogeneous graphs into homogeneous graphs for processing becomes overly complex and generally ineffective in such scenarios. For detecting account groups

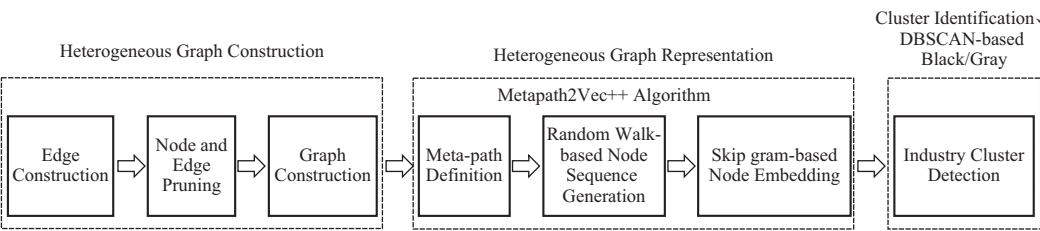


FIGURE 8.7 Detection frameworks for account groups aggregating multiple resources from the black/gray industry.

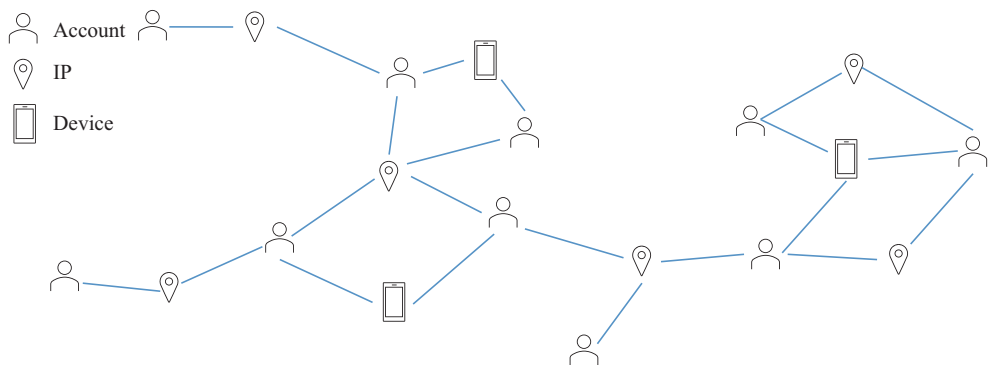


FIGURE 8.8 Heterogeneous graph aggregating multiple resources from the black/gray industry.

aggregating multiple resources of the black/gray industry, processing can be handled by constructing heterogeneous graphs, as illustrated in Figure 8.7.

8.1.2.1 Construction of Heterogeneous Graphs

8.1.2.1.1 Edge Construction Traffic fraud scenarios involve numerous essential resources for black/gray industries, such as black/gray industry IPs, devices, phone numbers, and email addresses. Business entities must extract relationships based on specific operational contexts. This section primarily illustrates edge construction using IP-to-account and device-to-account relationships.

8.1.2.1.2 Node and Edge Pruning IP node pruning methods can be referenced in Section 8.1.1. Device node pruning removes exceptionally anomalous device nodes. For example, if a device is associated with a large number of accounts, the device node may exhibit data anomalies and should be excluded.

For anomalous account nodes linked to a large number of IPs or devices, manual verification is required to confirm that no data collection anomalies exist before use; otherwise, they should be removed.

8.1.2.1.3 Graph Construction After the above two preprocessing steps, a heterogeneous graph aggregating various resources from the black/gray industry is ultimately obtained, as shown in Figure 8.8.

8.1.2.2 Representation of the Heterogeneous Graph

Due to the prominent heterogeneous graph properties within complex networks formed by aggregated resources from the black/gray industry, along with the presence of multiple node types, graph representation learning algorithms like node2vec—designed for homogeneous graphs—can only capture the structural information linking nodes. They fail to differentiate the information associated with different node types, severely limiting their effectiveness in identifying account syndicates. By pre-specifying metapaths (i.e., node type sequences), the metapath2vec++ algorithm not only learns the structural information of node associations but also effectively distinguishes the semantic information of node types. This provides a unique advantage in handling representation learning for heterogeneous graphs. The specific representation learning process can be divided into the following three steps.

8.1.2.2.1 Definition of Metapaths The black/gray industry resource clusters in Figure 8.8 primarily fall into three categories: account groups aggregating black/gray industry device resources; account groups aggregating black/gray industry IP resources; and account groups aggregating multiple resources, including both black/gray industry devices and IPs. The corresponding three metapaths for these scenarios are illustrated in Figure 8.9.

Specifically, Metapath 1 focuses on learning representations of IP aggregation structures; Metapath 2 focuses on learning representations of device aggregation structures; Metapath 3 focuses on learning representations of multi-resource aggregation structures. Additionally, when specifying metapaths, ensure node types exhibit left-right symmetry to enable repeated traversal.

8.1.2.2.2 Generating Node Sequences via Random Walks Conventional random walk algorithms (e.g., DeepWalk) do not distinguish between node types, randomly selecting the next node for traversal. This strategy ensures fairness in node selection within homogeneous graphs. However, for heterogeneous graphs with multiple node types, if significant node type skew exists, random walk algorithms like DeepWalk tend to favor high-visibility node transitions, potentially losing information from low-visibility nodes. The metapath2vec++ algorithm addresses this issue in heterogeneous graphs by pre-specifying metapaths for node types. During the random walk, it generates node sequences by traversing these predefined metapaths. This approach increases the “exposure” of low-visibility node types in the walk sequence, resulting in richer node sequence information.

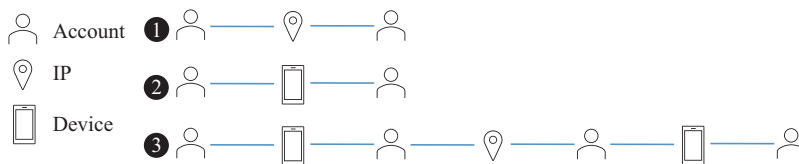


FIGURE 8.9 Three corresponding metapaths.

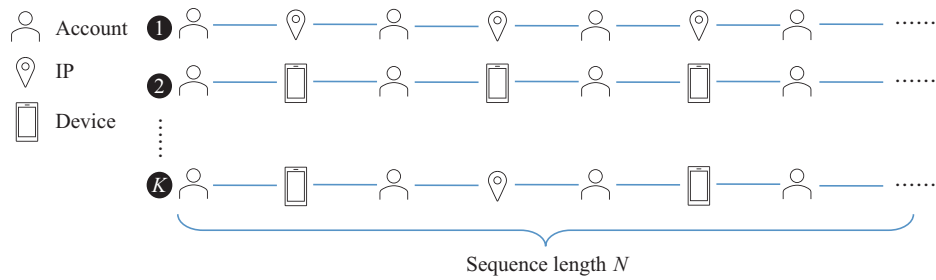


FIGURE 8.10 Node sequence generated by random walk.

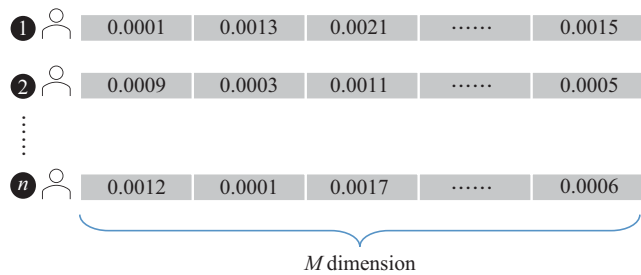


FIGURE 8.11 Generating M -dimensional account node vectors using the improved Skip-gram algorithm.

Set the maximum walk length N for the `metapath2vec++` random walk process (parameter N can be adjusted based on specific business needs). Perform the random walk according to the predefined metapath from the previous step to obtain a K -row node sequence. The node sequence generated by the random walk is shown in Figure 8.10.

8.1.2.2.3 Generating Node Vectors Using the Skip-Gram Algorithm The traditional Skip-gram algorithm does not distinguish between node types, treating the entire node sequence as a single type during training, thereby losing type-specific semantic information. The `metapath2vec++` algorithm improves upon Skip-gram by incorporating the characteristics of heterogeneous graphs. This enables the generated node vectors to contain not only structural information between nodes but also semantic information about node types.

Based on the node sequence generated by the previous step's random walk, an M -dimensional vector for the account node can be generated using the improved Skip-gram algorithm, as shown in Figure 8.11.

8.1.2.3 Cluster Identification of Groups

In the previous step, all account nodes in the heterogeneous graph were sampled via metapath random walks and projected into the same vector space using the Skip-gram algorithm. At this stage, business accounts sharing IP addresses and devices tend to cluster closely in the vector space. During traffic fraud operations, black/gray industry cybercriminals often share essential resources to reduce costs. This results in black/gray

industry accounts being closer together than normal accounts and exhibiting high-density clustering. Therefore, density-based clustering can effectively identify high-density clusters of black/gray industry accounts. Next, the DBSCAN algorithm is selected for clustering and identifying groups of black/gray industry accounts. The specific process can be divided into the following two steps.

1. Based on the DBSCAN algorithm, various clustered groups can be obtained. The group partitioning results based on the DBSCAN algorithm are shown in Figure 8.12.
2. Further screening of black/gray industry account groups is performed based on group attribute values. By associating the clusters obtained from DBSCAN with account blacklists, it is found that Group 1 has a blacklist ratio of 0.57, while neither Group 2 nor the outliers are associated with any blacklist entries. The screening of black/gray industry account groups based on group attribute values is illustrated in Figure 8.13. Assuming a blacklist ratio of ≥ 0.5 as the screening threshold for black/gray industry account groups, Group 1 is ultimately identified as such.

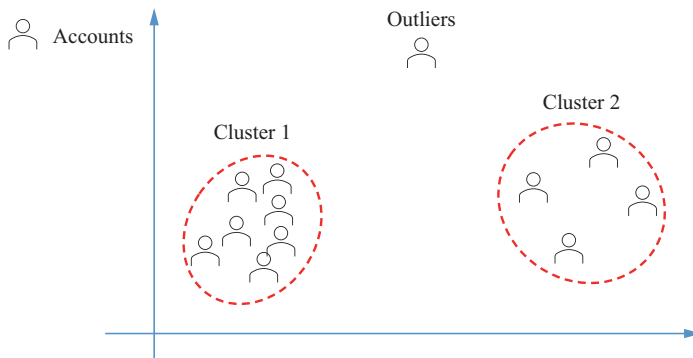


FIGURE 8.12 Group partitioning results based on the DBSCAN algorithm.

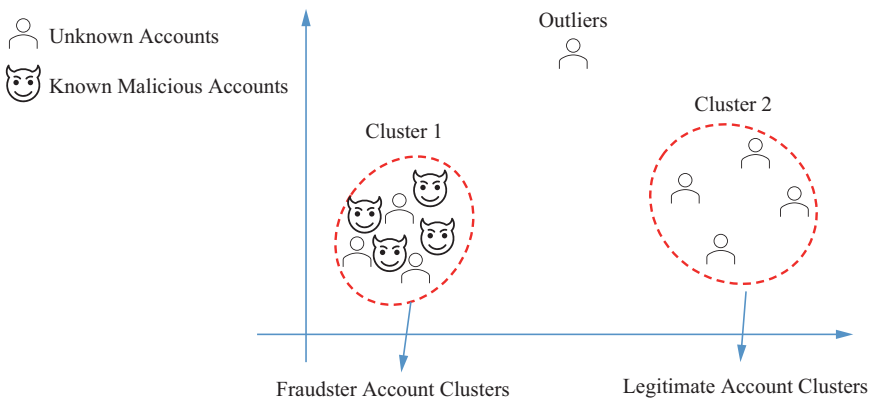


FIGURE 8.13 Screening black/gray industry account groups based on group attribute values.

8.2 MID-TRAFFIC PHASE STRATEGY

During the mid-traffic phase, black/gray industry actors begin operating on the platform, generating similar behavioral paths and attributes such as malicious clicks, comments, traffic diversion, and the propagation of fraudulent APKs and URLs. At this stage, traffic fraud forms a complex network of shared black/gray industry behavioral paths and attributes. During the pre-traffic phase of the complex network, nodes carry limited attribute information. During the mid-traffic phase of black/gray industry activities, substantial data is generated, leading to richer account node attributes. Consequently, the heterogeneous graph formed during the mid-traffic phase can enhance gang identification accuracy by incorporating node attribute information. The countermeasure for the mid-traffic phase is illustrated in Figure 8.14.

Next, we first identify structurally similar black/gray industry account groups based on the heterogeneous graph. Then, we construct a complex network of propagation resources to identify additional black/gray industry fraudulent resources such as apps and URLs. This enhances the fraudulent attribute information of account nodes, further improving the effectiveness of identifying black/gray industry account groups.

8.2.1 Structural Similarity Gang Detection

Section 8.1.2 described the pre-traffic phase approach using the metapath2vec++ heterogeneous graph algorithm to identify account groups, which lacked node attribute information and yielded limited effectiveness in node vector representation. As activities of the black/gray industry expand, node attributes in the mid-traffic phase become richer. Incorporating node attributes into the heterogeneous graph further enhances structural similarity-based account group detection. The structural similarity account group detection scheme for the mid-traffic phase is illustrated in Figure 8.15.

8.2.1.1 Construction of Heterogeneous Graphs with Node Attribute Features

Constructing a heterogeneous graph with node attribute features involves four steps: edge construction, node and edge pruning, node attribute feature construction, and graph construction.

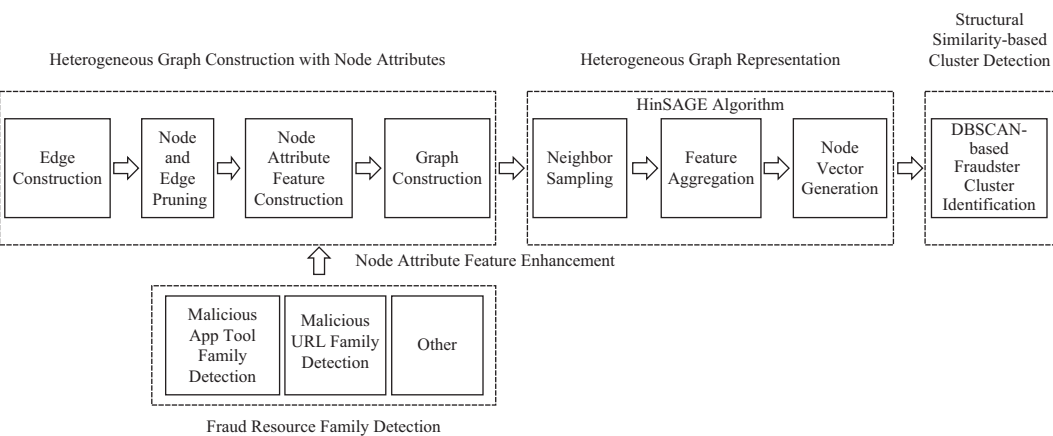


FIGURE 8.14 Complex network countermeasure for the mid-traffic phase.

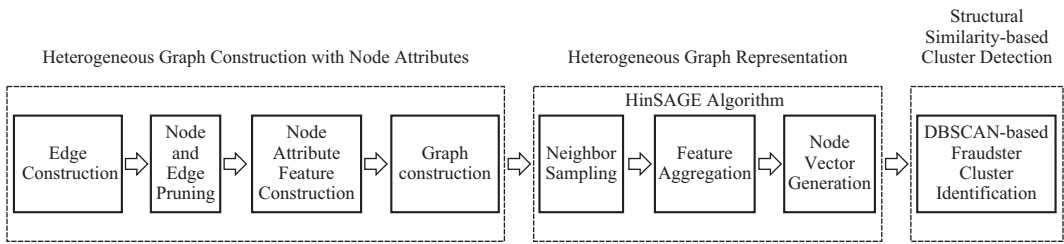


FIGURE 8.15 Mid-traffic phase structural similarity account group identification approach.

TABLE 8.2 IP Node Attribute Features

Feature Type	Feature Dimension
IP attributes	Is it a server IP?
	Is it a proxy IP?
	
	Is it an overseas IP?
IP-associated business data	Number of logins within N hours
	Number of associated accounts within N hours
	
	Number of associated devices within N hours

8.2.1.1.1 Edge Construction The mid-traffic phase involves numerous complex network relationships, such as shared connections between accounts and resources from the black/gray industry, propagation links between accounts and apps/tools or URLs from the black/gray industry, and purchase relationships between accounts and products. In practical operations, relationship extraction must be tailored to specific scenarios. For illustrative purposes, this section focuses on extracting IP-to-account and device-to-account relationships to construct edges, thereby explaining the graph construction process.

8.2.1.1.2 Node and Edge Pruning For IP node pruning, the primary objective is to eliminate public IPs that significantly skew user counts—such as base station IPs and corporate Wi-Fi networks—as these provide little value in identifying black/gray industry account syndicates. Pruning these IP nodes minimizes interference and reduces computational overhead.

For device node pruning, the main objective is to eliminate anomalous noise devices. For instance, if a single device is associated with a large number of accounts, this device node may exhibit data anomalies. Manual verification is required to confirm whether data reporting is normal, and such anomalous devices should be excluded.

Account node pruning follows a similar approach to device node pruning. For anomalous account nodes associated with a large number of IPs or devices, manual verification is necessary to confirm the absence of data reporting anomalies before they can be utilized. Otherwise, such account nodes should be excluded.

8.2.1.1.3 Node Attribute Feature Construction IP node attribute features are primarily constructed from IP properties and IP-associated business data, as shown in Table 8.2.

TABLE 8.3 Device Node Attribute Features

Feature Type	Feature Dimension
Device attributes	Is it a low-cost model?
	Is it an older operating system version?
	
	Root access status
Device-associated business data	Number of logins within N hours
	Number of associated accounts within N hours
	
	Number of associated IP addresses within N hours

TABLE 8.4 Attribute Features of Account Nodes

Feature Type	Feature Dimension
Account attributes	Is it a newly registered account?
	Whether it is registered with an overseas card?
	
Behavioral data within account operations	Is it a zombie account?
	High-frequency access within a short timeframe
	Whether access intervals are highly regular within a short timeframe?
	Number of Likes during late-night hours
	...
	Number of comments within N hours

The attribute characteristics of device nodes are primarily constructed from feature types such as device attributes and device-associated business data, as shown in Table 8.3.

The attribute characteristics of account nodes are primarily constructed from feature types such as account attributes and behavioral data within account operations, as shown in Table 8.4.

8.2.1.1.4 Graph Construction Based on the extracted relationships and node attribute features, the complex network heterogeneous graph during the mid-traffic phase is shown in Figure 8.16. It contains three types of nodes (account nodes, IP nodes, and device nodes), with varying dimensions of the attribute features for each node type.

8.2.1.2 Characterization of Heterogeneous Graphs

Based on the constructed heterogeneous graph, the HinSAGE algorithm is then employed to learn and generate node vectors. The HinSAGE algorithm supports both supervised and unsupervised learning modes. The core process of the unsupervised learning approach can be divided into the following two steps.

8.2.1.2.1 Neighbor Node Sampling Neighbor node sampling is an inward-to-outward process. Starting from the current target node, it first samples the first-layer nodes and then samples second-layer neighbors based on the first-layer results. Before sampling begins, the node sampling depth and the number of nodes per layer are set. Here, the sampling depth is set to 2, with the number of sampled neighbors in the first layer set to 2. Similarly,

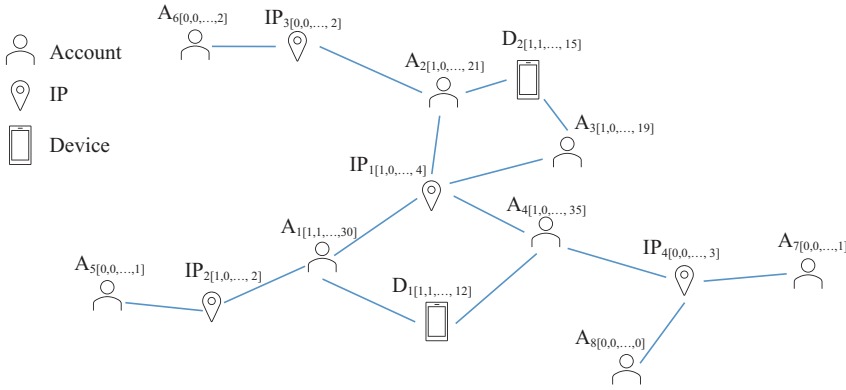


FIGURE 8.16 Complex network heterogeneous graph in the mid-traffic phase.

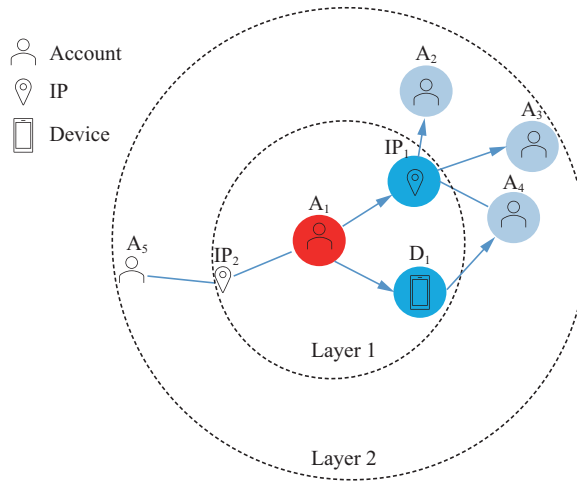


FIGURE 8.17 Sampling results of neighbor nodes.

the number of sampled neighbors per node in the second layer is also set to 2. Thus, the total number of sampled nodes in the second layer is 4. The sampling results for the neighbor nodes are shown in Figure 8.17. Here, A1 is the current target node. The sampled nodes in the first layer are IP1 and D1. In the second layer, the sampled neighbors of IP1 are A2 and A3, while both samples of D1 in the second layer are from A4. During sampling, missing node types may occur; in such cases, default values are used as fillers.

8.2.1.2.2 Feature Aggregation and Node Vector Generation Feature aggregation fundamentally involves aggregating neighbor node attribute features and then concatenating them with the current node's own attribute features. To optimize aggregation effectiveness in traffic fraud scenarios, mean aggregation is generally recommended for nodes of the same type, yielding the overall average risk profile of neighbor nodes. For nodes of different types, maximum value aggregation is typically advised. If a single node type in a traffic fraud scenario is risky, the likelihood of overall risk is also high.

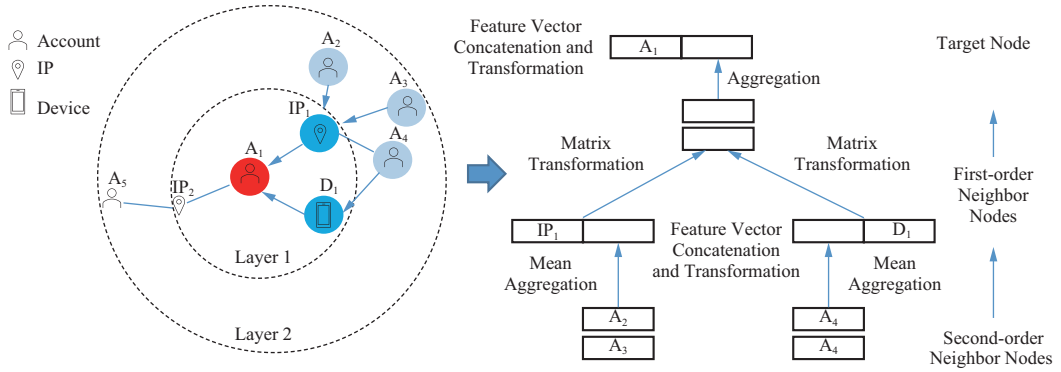


FIGURE 8.18 Feature aggregation process.

Feature aggregation is the inverse process of node sampling, proceeding from the outside in. The feature aggregation process is illustrated in Figure 8.18. First, the sampled nodes A2 and A3 of the same type from Layer 2 undergo mean aggregation to obtain the neighbor feature vector of node IP1. This vector is then concatenated and transformed with node IP1's own attribute feature vector to yield the node's updated feature vector. Similarly, the latest feature vector for node D1 is obtained. Subsequently, aggregation is performed based on the feature vectors of the first-layer nodes IP1 and D1. However, since IP1 and D1 belong to different types, their attribute feature vectors have varying dimensions. Therefore, matrix transformation is required to convert them into vectors of the same dimension. The transformation formula is as follows:

$$h_v^{k-1} = h_v^{k-1} \cdot W_{\phi(v)}^k$$

where v is the node type, k is the current depth and $W_{\phi(v)}^k$ is the transformation matrix at the current depth, which can be learned through training.

After transforming the attribute feature vectors of nodes IP1 and D1 into vectors of the same dimension via matrix transformation, maximum value aggregation is performed to obtain the feature vector of the neighboring nodes of target node A1. This vector is then concatenated and transformed with the attribute feature vector of target node A1 itself, ultimately yielding the updated feature vector of target node A1.

8.2.1.3 Structure Similarity Gang Identification

After obtaining node vectors of various types, the DBSCAN algorithm is further applied to identify structurally similar account clusters. The specific process can be referenced in Section 8.1.2 under the cluster identification step.

8.2.2 Fraud Resource Family Detection

The effectiveness of identifying structurally similar groups based on heterogeneous graphs primarily depends on the quality of node vector representations. The richness of node attribute information significantly influences the quality of these vector representations.

During the mid-traffic phase, black/gray industry actors begin disseminating fraudulent resources such as malicious apps and deceptive URLs for traffic diversion fraud. Information about these resource propagation activities serves as the most direct and effective indicator for distinguishing black/gray industry actors from legitimate users. This data can be incorporated as node attribute features into the complex network heterogeneous graph during the mid-traffic phase, thereby enhancing the representational power of node vectors.

Fraudulent resources disseminated through the black/gray industry come in many forms. The following discussion will primarily focus on fraudulent app resources as an example. By detecting families of fraudulent app resources spread through these channels, we can identify and track more fraudulent apps. This enhances the fraud attribute information associated with account nodes, thereby improving the recognition effectiveness of heterogeneous graphs.

In highly competitive environments, various variants of fraudulent app families often emerge, exhibiting a long-tail distribution. Examples include apps with identical package names but different software names, identical software names but different package names, or matching software and package names but differing file SHA1 values. These variations make it challenging for businesses to fully identify and cover them. However, from a complex network perspective, these variants share certain connections—such as common software names, package names, or certificates. Therefore, we can construct a heterogeneous graph based on shared relationships among SHA1 values, certificates, package names, and software names to identify black/gray industry app families. This process involves five key steps:

8.2.2.1 Edge Construction

Numerous relationships exist between a fraudulent app's SHA1, certificate, package name, and software name. Here, we primarily extract three types of edge relationships: SHA1-certificate, SHA1-package name, and SHA1-software name. Among these, SHA1 uniquely corresponds to a single app file; any identical app file will yield the same SHA1. Meanwhile, the black/gray industry can modify certificates, package names, and software names.

8.2.2.2 Node and Edge Pruning

First, we perform pruning based on time range, primarily extracting app relationship data from the past N days.

Second, we conduct pruning based on node attributes. To avoid introducing excessive legitimate nodes during graph construction, high-frequency SHA1, certificates, package names, and software names are primarily filtered out.

8.2.2.3 Weighting Design

Fraudulent apps are primarily used or disseminated by the black/gray industry. Due to their rapid mutation and lower popularity, apps with lower popularity scores are more likely to be fraudulent apps from the black/gray industry. Here, we calculate the popularity values (ranging from 0 to 1 million, with values exceeding 1 million capped at 1 million) for three types of edges: SHA1-certificate, SHA1-package name, and SHA1-software

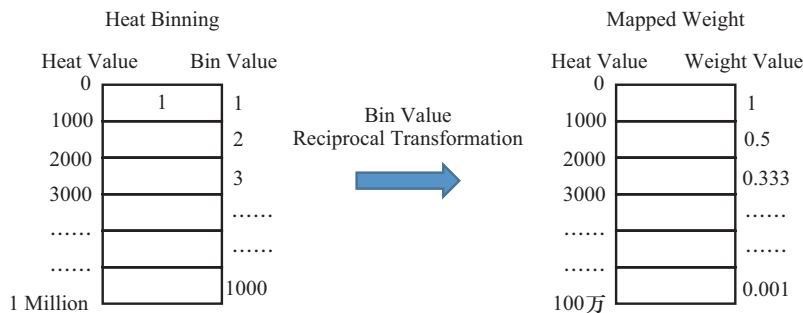


FIGURE 8.19 Weight design process.

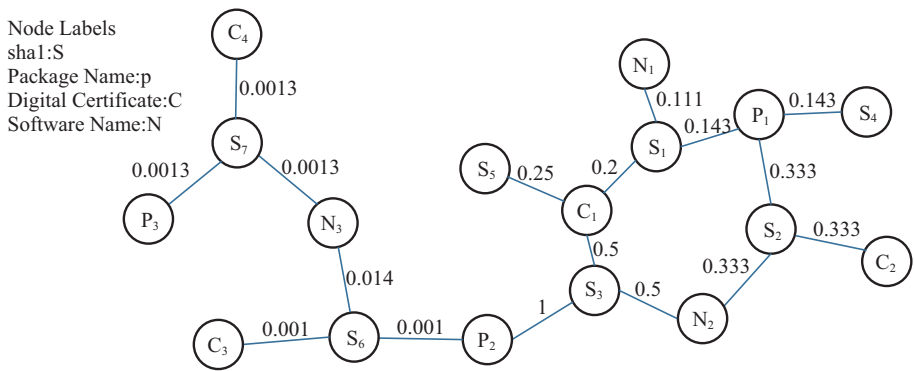


FIGURE 8.20 Genealogy of black/gray industry fraud app families.

name. First, the data is divided into 1,000 equal-interval bins, each spanning a 1,000-point range. The reciprocal of each bin’s range width is then used as its weight, as illustrated in Figure 8.19. A higher weight indicates a greater likelihood that the app belongs to the black/gray industry fraud family.

8.2.2.4 Graph Construction

Based on the extracted relationships and after pruning nodes and edges, the lineage relationships within the black/gray industry fraud app family are shown in Figure 8.20. Specifically, S1 and S2 share package name P1; S1 and S3 share certificate C1; S2 and S3 share software name N2; S3 and S6 share package name P2; S6 and S7 share software name N3.

8.2.2.5 Community Detection Algorithms for the Black/Gray Industry Fraud App Families

Based on the obtained lineage relationships of black/gray industry fraud app families, multiple iterations of community detection algorithms achieve equilibrium. The final family partitioning results are output, and the ultimate black/gray industry fraud app families are filtered based on family attribute values (such as the proportion of seed apps), as shown in Figure 8.21.

Through the detection of the black/gray industry fraud app family, more fraudulent apps can be identified and introduced as account node attribute information into the heterogeneous graph in Section 8.2.1. This further enhances node vector representation and improves the effectiveness of identifying account syndicates.

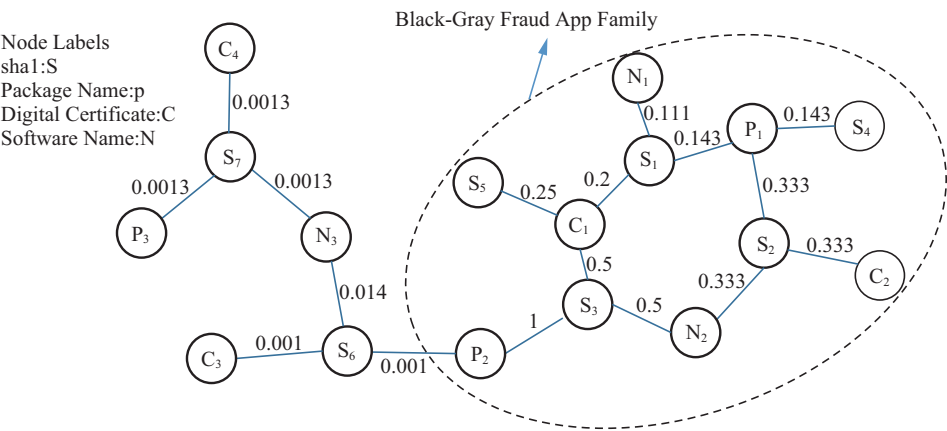


FIGURE 8.21 Final app families for the black/gray industry fraud.

8.3 POST-TRAFFIC PHASE SOLUTIONS

In the post-traffic phase, social attributes become more prominent. Following the mid-traffic phase traffic diversion, black/gray industries in the post-traffic phase begin establishing connections with users to carry out fraud. Examples include black intermediaries forming financial ties with users to commit loan fraud and fake merchants establishing transactional links with users to perpetrate transaction fraud. At this stage, the black/gray industry fraud primarily forms complex networks centered around shared intermediaries (e.g., black intermediaries), as illustrated in Figure 8.22. Here, A5, A6, and multiple accounts establish relationships, representing typical intermediary nodes.

The post-traffic phase primarily targets fraud backbone nodes identified through complex network analysis, spreading to suspicious user nodes to enhance fraud detection coverage. The complex network countermeasure scheme for the post-traffic phase is illustrated in Figure 8.23.

Next, we illustrate the graph construction and modeling process using the example of identifying black intermediaries in loan fraud scenarios.

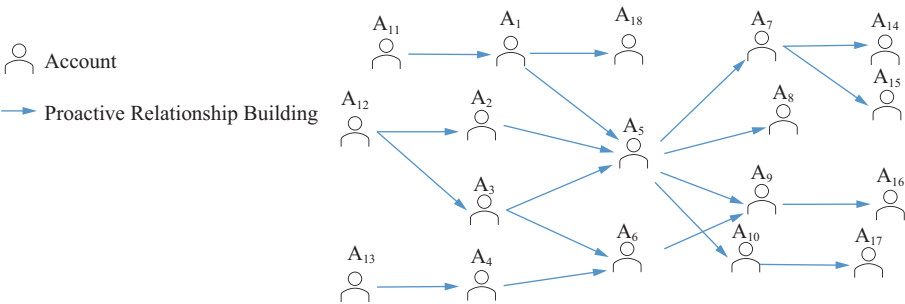


FIGURE 8.22 Complex network of shared intermediaries.

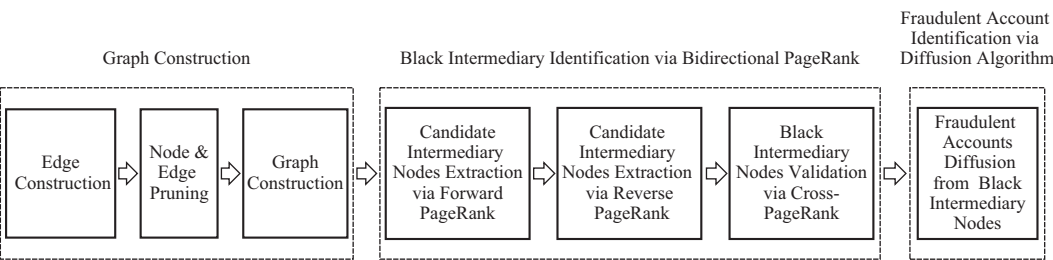


FIGURE 8.23 Complex network countermeasure scheme for the post-traffic phase.

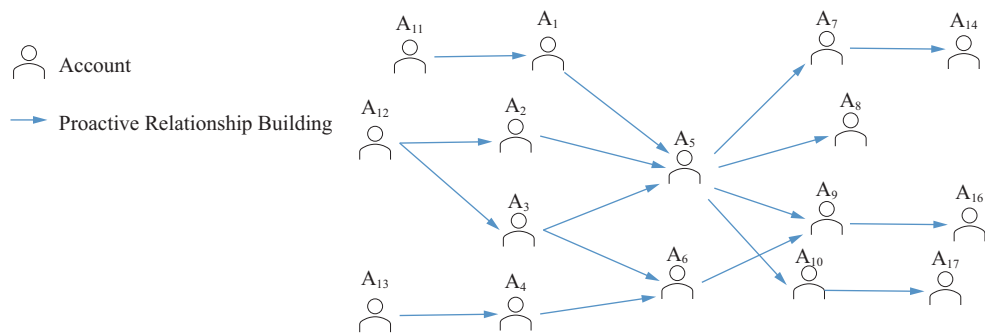


FIGURE 8.24 Financial attribute subgraph in the post-traffic phase.

8.3.1 Graph Construction

8.3.1.1 Edge Construction

Here, we primarily extract relationships established between accounts to construct edges.

8.3.1.2 Node and Edge Pruning

First, pruning based on time range. We primarily extract account relationship data from the past N days. Second, pruning based on node attributes. Given the loan fraud scenario, we filter nodes with financial attributes and remove those without, thereby obtaining a financial attribute subgraph.

8.3.1.3 Graph Construction

Based on the extracted relationships and after pruning nodes and edges, the final financial subgraph for the post-traffic phase is shown in Figure 8.24. Here, A₁₅ and A₁₈ are excluded due to the absence of financial attributes. The direction of the arrows indicates that the source node actively establishes a relationship with the destination node. Therefore, applying PageRank to propagate node PR values along the direction of the arrows is called the forward PageRank scheme, while propagating node PR values against the direction of the arrows is termed the reverse PageRank scheme. The following details the bidirectional PageRank-based approach for identifying black intermediaries.

8.3.2 Bidirectional PageRank Cross-Identification of Black Market Brokers

Within the complex network of relationships established by loan fraud black intermediaries, individuals with borrowing needs but problematic credit histories proactively contact multiple unofficial loan brokers. These users hope to successfully apply for loans by having their documents packaged through these intermediaries. From the black broker's perspective, to maximize profits, they aggressively expand their client base by proactively connecting with numerous account nodes possessing potential borrowing needs. On one hand, they charge high fees for packaging materials for users with poor credit histories. On the other hand, they deceive clients by offering to directly borrow from blacklisted accounts to package loan materials as a gimmick to obtain fees.

The distinction between black intermediary nodes and ordinary nodes lies in their role as pivotal intermediaries. Black intermediary nodes serve as core hubs where numerous borrowing-needy nodes proactively establish connections. Simultaneously, these black intermediary nodes actively cultivate relationships with other nodes harboring potential borrowing needs. From the perspective of node influence, since black intermediaries are influential backbone nodes within complex networks, and PageRank is a typical algorithm for measuring node influence, black intermediary backbone nodes can be identified based on PageRank. The specific process for cross-identifying black intermediaries using bidirectional PageRank can be divided into the following three steps.

8.3.2.1 Extracting Candidate Intermediary Nodes Using Forward PageRank

A large number of user account nodes with borrowing needs actively establish relationships with black intermediary nodes. Considering this direction of user-initiated connections as the positive direction, from the perspective of PageRank propagation, the PR values of numerous user nodes are transmitted to black intermediary nodes. This results in black intermediary nodes possessing higher PR values than ordinary nodes. Consequently, nodes with high positive PR values can be filtered as candidate black intermediary nodes based on a threshold. The forward PageRank propagation process is illustrated in Figure 8.25.

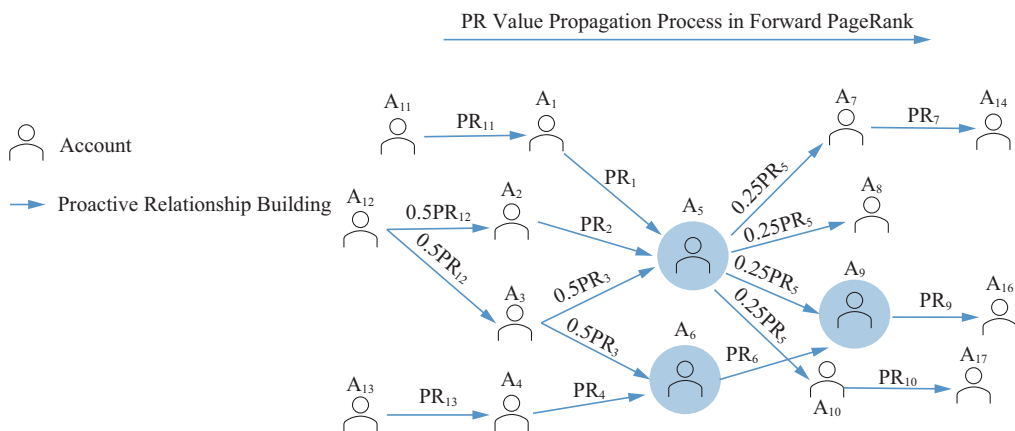


FIGURE 8.25 Forward PageRank PR value propagation process.

Specifically, in the forward PageRank propagation, since user nodes actively establish relationships with nodes A5, A6, and A9 in relatively large numbers, when PageRank iterations reach equilibrium, the PR values of these three nodes are higher than those of other nodes, making them influential backbone nodes. Furthermore, after the initial pruning phase, the remaining user nodes predominantly possess financial attributes—that is., they are influential nodes that have actively established relationships with numerous users having financial attributes. Such account nodes are more likely to be loan black intermediaries. Therefore, during the forward PageRank process, nodes A5, A6, and A9 are selected as candidate loan black intermediary nodes.

8.3.2.2 *Extracting Candidate Broker Nodes via Reverse PageRank*

The candidate loan black intermediary nodes identified through the forward PageRank process primarily represent intermediate influential nodes actively connected to numerous users with borrowing needs. However, some gambling users who proactively establish relationships with other intermediaries (e.g., gambling intermediaries) also carry debt risks and borrowing demands, exhibiting similar characteristics in the forward PageRank process.

To more accurately identify loan black intermediaries, further screening at the business level is required. Loan black intermediaries differ from other intermediaries, like gambling ones, in how they proactively establish relationships with target users. While gambling and other intermediaries primarily target users without borrowing needs, loan black intermediaries specifically target users with borrowing needs. Consequently, to expand their customer base, loan black intermediary nodes may also proactively establish relationships with numerous user account nodes expressing borrowing needs. In Figure 8.25, candidate intermediary node A5 has proactively established relationships with nodes A7, A8, A9, and A10. However, during the forward PageRank (PR) value propagation process, user account nodes A7, A8, A9, and A10—which may have borrowing needs—failed to propagate PR values to the black intermediary node. Consequently, these nodes were not effectively utilized in the identification process of black intermediary nodes.

Further business analysis reveals that if PageRank values propagate in the opposite direction of active relationship establishment, user account nodes A7, A8, A9, and A10—which may have borrowing needs—can transmit PR values to the subsequent black intermediary node A5. This method is termed reverse PageRank propagation, as illustrated in Figure 8.26. Through reverse PR propagation, PR values from numerous passively connected user nodes are transmitted to black intermediary nodes, elevating their PR values. This enables the identification of candidate black intermediary nodes by filtering those with high reverse PR values based on a threshold.

From the perspective of reverse PR propagation, since nodes A3, A5, and A12 have relatively more user nodes passively connected to them, when PageRank iterations reach equilibrium, these three nodes exhibit higher PR values than others. They thus constitute influential backbone nodes in the reverse PR propagation process. After initial pruning, these user nodes retain accounts with financial attributes—influential nodes passively connected to numerous financially active users. Such accounts are more likely to be loan black intermediaries. Thus, nodes A3, A5, and A12 are selected as candidate black intermediary nodes during reverse PageRank.

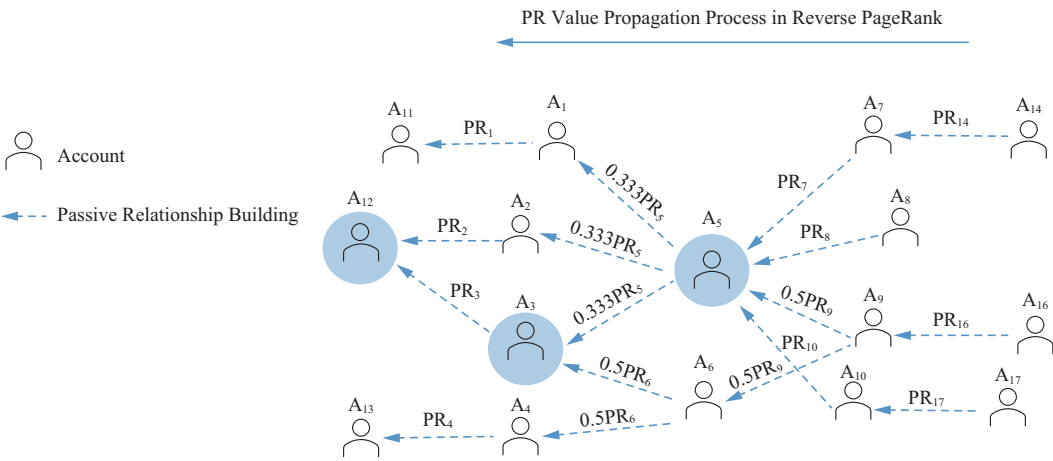


FIGURE 8.26 Reverse PageRank PR value propagation process.

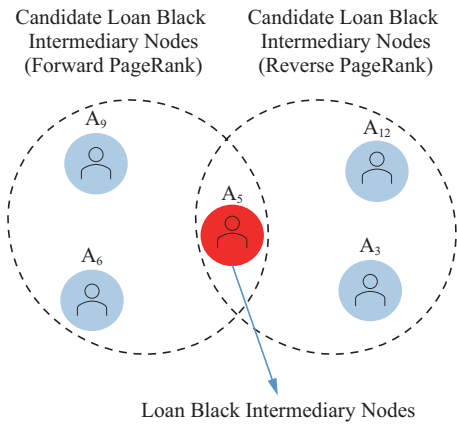


FIGURE 8.27 Loan black intermediary nodes cross-verified by positive and negative PageRank.

8.3.2.3 Cross-Verification of Black Intermediary Nodes via Positive and Reverse PageRank Based on the results from the above two steps, the forward PageRank process identified candidate loan black intermediary nodes as A5, A6, and A9, while the reverse PageRank process identified candidates as A3, A5, and A12. The loan black intermediary nodes cross-validated by both forward and reverse PageRank are shown in Figure 8.27, with A5 ultimately confirmed as the loan black intermediary node.

8.3.3 Fraudulent User Account Coloring Identification

Based on the loan black intermediaries identified in the previous step, we further propagate additional loan fraud user nodes through both active and passive first-degree relationships to enhance our capability to identify loan fraud users. The loan fraud user accounts propagated from the loan black intermediary nodes are shown in Figure 8.28.

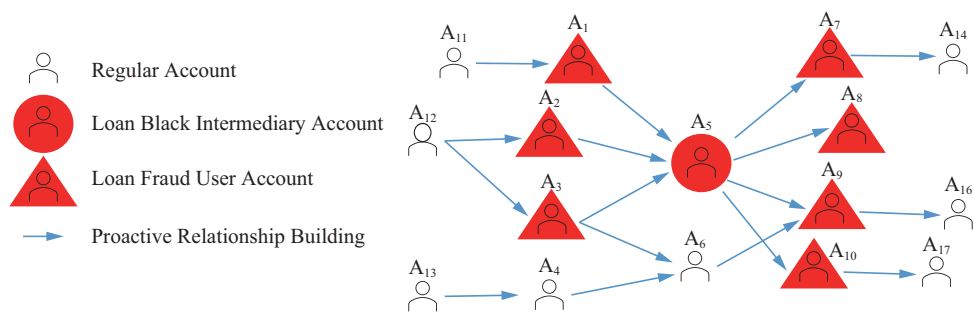


FIGURE 8.28 Loan fraud user accounts propagated from loan broker nodes.

This section primarily illustrates the approach using loan fraud black intermediaries as an example. Similar coloring identification schemes apply to backbone nodes of other intermediary types and fraudulent users, with the core distinction lying in the differing fraud patterns within their business models.

8.4 SUMMARY

This chapter constructs corresponding schemes to identify traffic fraud based on the typical complex network characteristics formed during each stage of the traffic lifecycle. First, leveraging the complex network of shared resources from the pre-traffic phase of the black/gray industry, algorithms such as community detection are used to identify groups aggregating these resources. Second, building upon the complex network of co-behavior paths and node attributes formed during the mid-traffic phase, heterogeneous graph algorithms detect structurally similar account groups. Finally, leveraging the complex network of shared relationships formed in the post-traffic phase, the bidirectional PageRank scheme detects intermediary nodes and identifies loan fraud user accounts through coloring. This establishes a comprehensive, full-lifecycle complex network countermeasure spanning the pre-traffic phase, mid-traffic phase, and post-traffic phase.

Multimodal Integrated Countermeasure Scheme

From advertising promotions to attracting user registrations, through user logins for promotional activities, to final marketing settlements by business partners—the black/gray industry permeates every stage of platform traffic flow. Yet every malicious act leaves traces: during account registration, the black/gray industry may use specialized mobile phone numbers for bulk registrations, revealing anomalies in the relationships between platform accounts, mobile phone numbers, device IDs, and network IPs. Similarly, on dating platforms where users post text and images, the black/gray industry may mass-distribute pre-generated content to attract customers, leaving traces in the content itself.

The traffic anti-fraud techniques discussed earlier focus on identifying malicious traces within single dimensions, such as device or mobile phone number dimensions, resulting in limited coverage against black/gray industry traffic fraud. To enhance coverage against such traffic fraud, the following section introduces a multimodal integrated countermeasure solution. A modality refers to a source or data format. In identifying traffic fraud from the black/gray industry, each trace of malicious activity constitutes information from a single modality. The countermeasures discussed in previous sections for different dimensions represent solutions for single-modality threats. The multimodal integrated countermeasure combines traces from diverse sources into a coherent multimodal framework to achieve superior detection of traffic fraud. A comparison between single-modal and multimodal integrated countermeasures is shown in Figure 9.1.

From a data dimension perspective, compared to single-modal adversarial schemes, multimodal integrated adversarial schemes fuse data from multiple modalities, resulting in higher data dimensionality. In high-dimensional spaces, black/gray industry samples and legitimate samples exhibit greater separability, thereby enhancing the precision of traffic fraud detection. From a business perspective, black/gray industry activities may not be conspicuous within a single modality. However, as long as black/gray industry fraudsters

Single-Modal Defense	Multi-Modal Defense
☐ Unidimensional Data Pattern	☐ Cross-Modal Feature Fusion
☐ Low Fraud Signal Saliency	☐ Residual Fraud Footprint
☐ Suboptimal Precision/Recall	☐ Enhanced Threat Coverage

FIGURE 9.1 Comparison of single-modal and multimodal integrated countermeasures.

leave traces in any modality, they can be identified based on these clues. Therefore, the multimodal integrated adversarial approach demonstrates higher accuracy and coverage in detecting traffic fraud.

9.1 SOURCES OF MULTIMODAL DATA

Given that black/gray industries may leave traces of malicious activity across various stages of a platform, modal data can typically be categorized into four types: complex network relationship information, textual information, image information, and other modal information.

9.1.1 Relationship Graph Information

Complex network relationship information describes the associative connections between different entities within traffic flows, such as the relationships among platform accounts, mobile phone numbers, device IDs, and network IP addresses. Here, a platform account refers to the user ID registered on a business platform; typically, a platform account requires binding to a mobile phone number. Users require devices to log into platform accounts, where the device ID serves as the device identifier. Devices need a network environment to access the platform, and the network IP acts as the logical address distinguishing each network and host. The complex network relationships within the account login environment are illustrated in Figure 9.2.

This relationship graph documents the correspondence between platform accounts, their associated mobile numbers, active device IDs, and corresponding network IP addresses. To extract insights from this graph, one can either construct rules and models by integrating multi-dimensional data or directly apply complex network models for identification. Chapter 8 of this book provides a detailed introduction to complex network adversarial solutions based on these models.

The application of complex network models for adversarial defense is further detailed in Chapter 8 of this book. The exploration of relationship graph information can be primarily categorized into the following three directions.

- Anomaly Node Detection: Identifying abnormal nodes within the complex network relationships of account login environments. Such anomalies typically manifest as unusually high measures like degree centrality or PageRank scores. Common

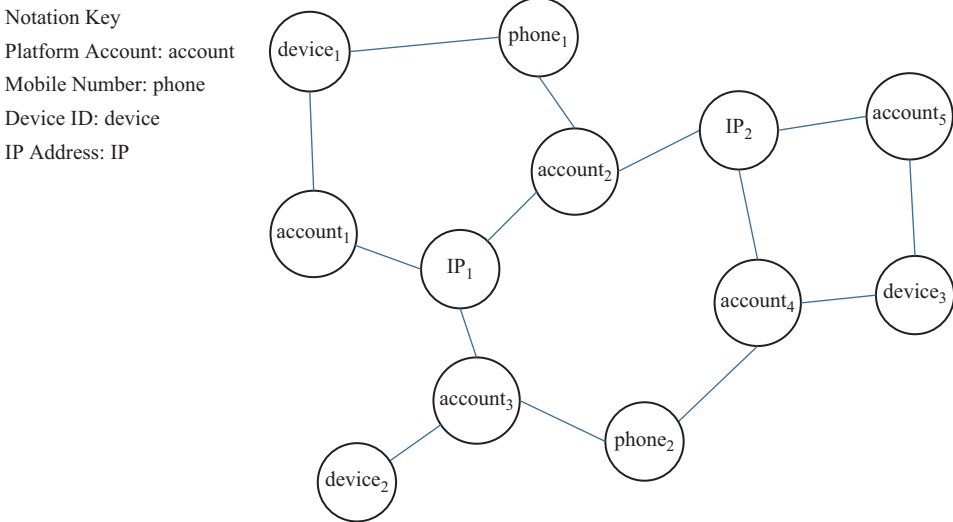


FIGURE 9.2 Complex network relationships in account login environments.

examples include mobile phone numbers controlled by the black/gray industry registering excessive accounts or devices logging into an unusually high number of accounts.

- **Anomalous Community Detection:** Utilizing label propagation algorithms and Louvain algorithms to discover communities within complex network relationships. Extracting and validating corresponding communities helps identify anomalous ones. Platform accounts, mobile phone numbers, and devices within anomalous communities can be recognized as collectively held by black/gray industry groups.
- **Vector Representation of Learning Nodes:** By leveraging structural information within complex network relationships, algorithms like DeepWalk and node2vec can generate node vector representations that capture structural information based on node-to-node relationships, thereby identifying anomalous nodes.

9.1.2 Textual Information

Another method the black/gray industry uses to acquire customers from platforms involves posting texts in articles or comments that redirect users to pornography and gambling sites, as well as fraudulent texts designed to induce users to click and share. In this process, malicious text information is left behind. For example, there are instances of spam comments left by black/gray industry groups in product comment sections, which use financial incentives or pornographic content to attract users for the purpose of traffic diversion fraud or traffic fraud.

To identify and combat these traffic diversion fraud and fraudulent texts, their content must be analyzed.

9.1.3 Image Information

Images are another common medium for content expression. Compared to text, visual information is richer and more effective at capturing user attention. The black/gray industry often publishes images containing enticing content to achieve traffic diversion and fraud objectives. For example, by using phrases such as “Earn money while lying at home” and “Make 100 a day,” combined with a QR code for downloading an app, users are lured into downloading the corresponding app. They are then redirected to crowdsourcing platforms, where they become victims of download fraud. Image algorithms can identify and combat traffic diversion images posted by members of the black/gray industry on various platforms. There are generally two approaches to addressing black/gray industry activities through images: first, extracting text from images and using text algorithms for recognition; second, directly applying image algorithms to detect the images. In practice, both approaches are often combined.

- **Image Text Extraction and Recognition:** Given the explicit purpose of images posted by members of the black/gray industry—often containing descriptive text—these texts can be accurately identified. Optical character recognition technology extracts text from images, which is then analyzed to determine if it originates from black/gray industry sources.
- **Image Content Recognition:** From the perspective of image content, detection models can be trained to recognize and intercept images published by black/gray industries.

9.1.4 Other Modal Information

Beyond the relationship graph data, textual information, and image data discussed above, the black/gray industry also disseminates messages via audio or video formats—such as voice messages in game chat channels or videos posted on platforms—to evade detection. Consequently, audio and video-based detection frameworks can also be applied in anti-fraud scenarios.

Figure 9.3 illustrates how multimodal information can identify harmful content produced by black/gray industries. By integrating information from four modalities—text, images,

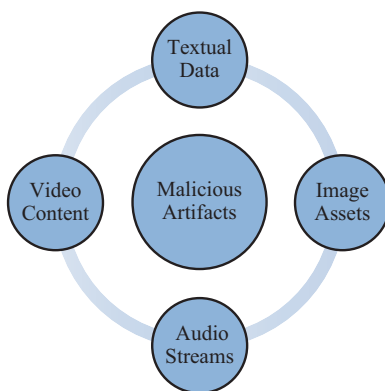


FIGURE 9.3 Multimodal information for identifying harmful content produced by black/gray industries.

audio, and video—it is possible to comprehensively identify harmful content produced by black/gray industries, enabling effective detection and interception. Subsequently, leveraging complex account network relationship information allows for the identification of criminal syndicates from an organizational perspective, achieving a holistic recognition of black/gray industries from local to systemic levels.

9.2 MULTIMODAL FUSION APPROACH

To apply multimodal information for identifying traffic related to the black/gray industry, multimodal data must first be fused. The relationship between multimodal information fusion and modeling is illustrated in Figure 9.4. Following the three stages of modeling, multimodal information fusion can be categorized into: data-layer fusion, feature-layer fusion, and decision-layer fusion.

- Data-layer fusion: During data collection, merge data of the same type for subsequent joint processing into features.
- Feature-layer fusion: Data is collected separately, processed individually into features, then fused together for subsequent modeling.
- Decision-layer fusion: Features are not fused; separate models are constructed, and decisions are made through the collective operation of multiple models.

Next, we will detail the multimodal fusion approaches across these three stages and their common application scenarios. However, in practical engineering, we do not adhere to a fixed standard fusion approach. Instead, we flexibly select different fusion approaches based on data characteristics, business scenarios, business objectives, and evaluation criteria.

9.2.1 Data-Layer Fusion

Data-layer fusion refers to collecting and processing data from different modalities together during the data acquisition and preprocessing stage. This approach facilitates the transformation of multimodal information into features. Its advantage lies in achieving thorough data integration at the foundational level, which enhances the effectiveness of subsequent model training. However, a disadvantage is that if the underlying data exhibits excessive variability, the correlation between modalities may weaken, potentially undermining the quality of subsequent modeling.

Therefore, when integrating multimodal information at the data layer, prioritize merging information of the same form. For example, black/gray industries leaves various textual traces on platforms to achieve traffic diversion, including comment texts, user profile

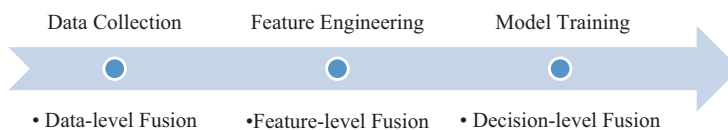


FIGURE 9.4 Relationship between multimodal information fusion and modeling.

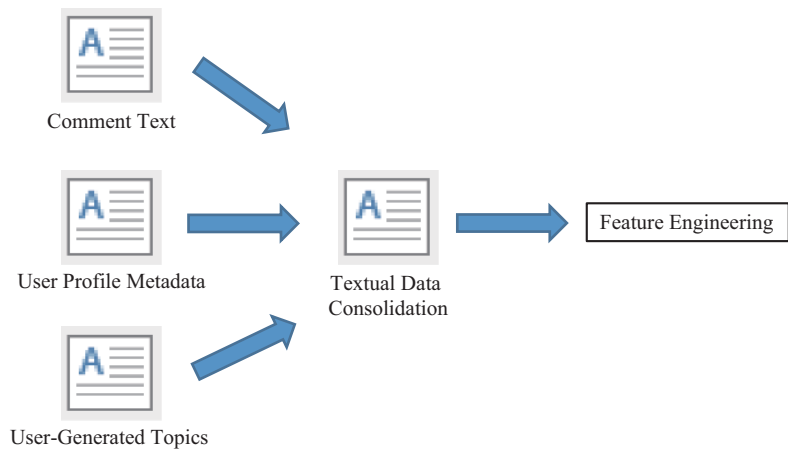


FIGURE 9.5 Data-layer integration of text from different sources.

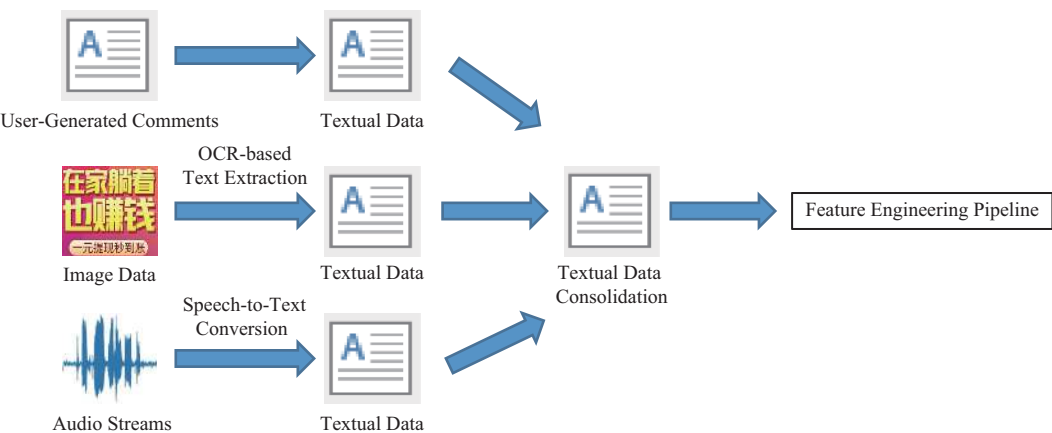


FIGURE 9.6 Data-layer fusion of different data modalities.

information, and user-posted topic content. Relying solely on text from any single source risks missed detections. However, integrating text from multiple sources at the data layer enables thorough fusion, significantly boosting the model’s recall rate in later stages. Figure 9.5 illustrates the data-layer fusion of text from diverse sources.

For text content sharing the same data format, direct merging at the data layer is feasible. However, merging data of different formats—such as text, images, and audio—proves challenging. In such cases, preprocessing is required to convert disparate formats into a uniform format before merging. For textual information, the content is extracted directly. For image information, text within the images must be extracted to obtain textual content. For audio information, speech must be converted into text to derive textual content. Finally, the textual information derived from converting image and audio information is merged with the original textual information. The data-layer fusion for different data formats is illustrated in Figure 9.6.

9.2.2 Feature-Layer Fusion

For multimodal information of the same form, or information that can be converted to the same form through data preprocessing, modal fusion can be achieved at the data layer. However, when data forms are difficult to unify or the information expressed by the data lacks correlation, fusion at the data layer alone is insufficient to achieve satisfactory results. By applying feature engineering to process information from different modalities, the original information can be mapped into a feature space, and the feature vectors from different modalities can then be concatenated.

Since complex network relationship information, textual information, and image information are all unstructured data that models cannot recognize, feature engineering is required to convert these unstructured data into features data recognizable by models. The schemes for converting different modal information into features data are shown in Table 9.1.

Fusing the aforementioned complex network relationship information, textual information, and image information at the feature layer involves a common approach of concatenating feature vectors from different modalities. Two prevalent methods exist for feature vector concatenation: one involves concatenation followed by compression via principal component analysis to obtain fused features; the other first elevates features to vectors of equal length before concatenation or aggregation processing (e.g., averaging, summation, and maxing).

9.2.2.1 Concatenation Followed by Compression

For features F_1 and F_2 , where F_1 has feature length m and F_2 has feature length n , concatenation yields F_{concat} with feature length $m + n$. Since redundant information exists between features of different modalities, feature compression is required after concatenation to eliminate redundancy. An example of feature fusion through concatenation of feature vectors is shown in Figure 9.7. After concatenation, principal component analysis is applied for feature compression, resulting in the final fused feature result F_{fusion} .

9.2.2.2 Dimension Expansion Followed by Aggregation

The process of feature dimensionality expansion followed by fusion is illustrated in Figure 9.8. First, image information undergoes feature engineering to produce F_1 , while text information undergoes feature engineering to produce F_2 . At this stage, F_1 and F_2 may have inconsistent dimensions. Then, F_1 and F_2 undergo dimensionality expansion,

TABLE 9.1 Schemes for Converting Different Modalities of Information into Feature Data

Modal Information	Feature Construction Approach	Feature Meaning
Complex network relationship information	Learning node embedding vectors	Representing structural information of nodes in complex networks
Textual information	Learning sentence vectors from text through temporal learning	Representing temporal information of text vocabulary
Image information	Feature construction via convolutional neural networks	Representing texture information in images

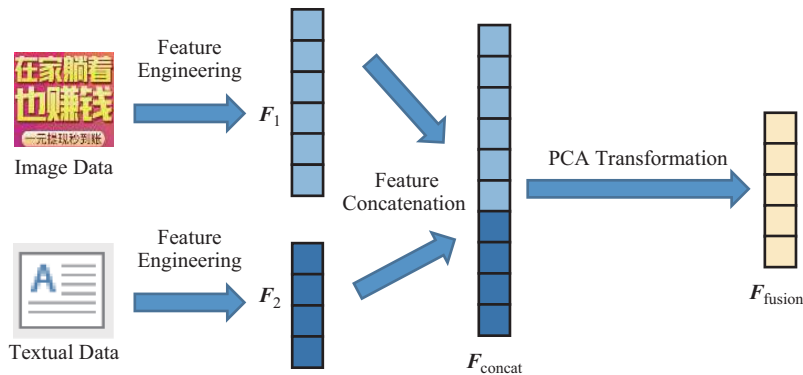


FIGURE 9.7 Example of feature fusion through front-to-back concatenation of feature vectors.

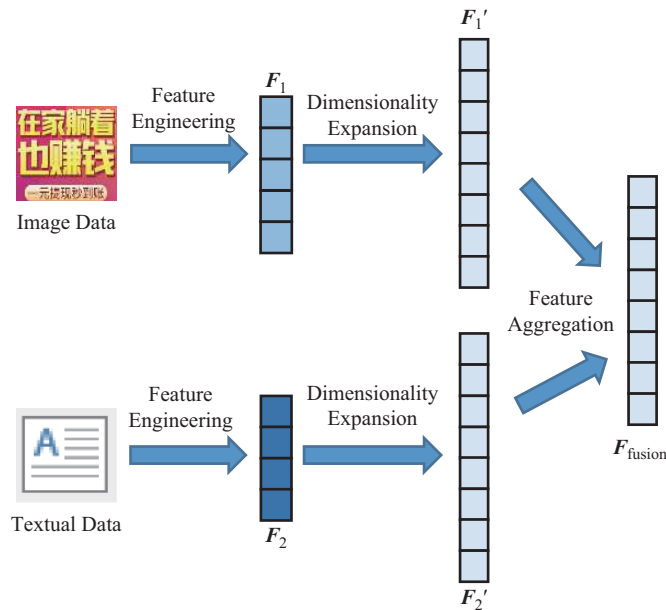


FIGURE 9.8 Process of dimensionality expansion followed by fusion for modal features.

generating features F_1' and F_2' that exist in the same feature space. Finally, the features F_1' and F_2' are added to obtain the fusion result F_{fusion} . Beyond addition, common operations include averaging, taking the maximum value, or direct concatenation.

9.2.3 Decision-Layer Fusion

The data-layer and feature-layer fusion mentioned above both involve merging data from different modalities before modeling. This approach simplifies subsequent modeling by generating unified features from the fused data.

However, in practical applications, when multimodal data is complex, feature dimensions often become excessively high after fusion. Attempting to fit this data with a single model may result in underfitting. An alternative approach involves building separate models for each data modality to generate distinct model scores, which are then fused

to determine whether traffic constitutes traffic fraud from the black/gray industry. This method of integrating multiple modalities is termed decision-layer fusion.

The entire process of decision-layer fusion of multimodal information primarily involves two steps: first, constructing model scores from each modality's information, where different model scores represent the suspected level of traffic fraud from various dimensions. Second, collecting the model scores obtained from each modality and making a comprehensive judgment. Based on the method of comprehensive judgment in the second step, decision-layer fusion is further classified into the following two categories.

9.2.3.1 The Types of Modal Information Are Relatively Limited

When the number of modal information types is relatively small, the number of models constructed in the first step for each modal information type is also limited. In this case, the final result can be determined through model score voting across all modal information types. The decision-layer fusion based on model score voting is illustrated in Figure 9.9. For comment text information, feature engineering yields feature F_1 , which is then used to train the corresponding text model. After the model predicts traffic, it produces model score S_1 . Image information undergoes feature engineering to derive feature F_2 , which is then used to train a corresponding image model. After predicting traffic, the model yields score S_2 . Audio information undergoes feature engineering to derive feature F_3 , which is then used to train a corresponding audio model. After predicting traffic, the model yields score S_3 . The final model judgment result S is obtained by combining the voting results of S_1 , S_2 , and S_3 . Voting methods typically include soft voting and hard voting. When a single modality's model score exhibits extremely high maliciousness, fusion is unnecessary, and a direct judgment can be made.

- **Hard Voting:** S_1 , S_2 , and S_3 represent the probabilities predicted by three models that the traffic is traffic fraud. First, convert the results of S_1 , S_2 , and S_3 to either 0 or 1, where 1 indicates a traffic fraud determination and 0 indicates a Normal Traffic determination. Then, vote using the three determinations, ultimately adopting the majority vote result.

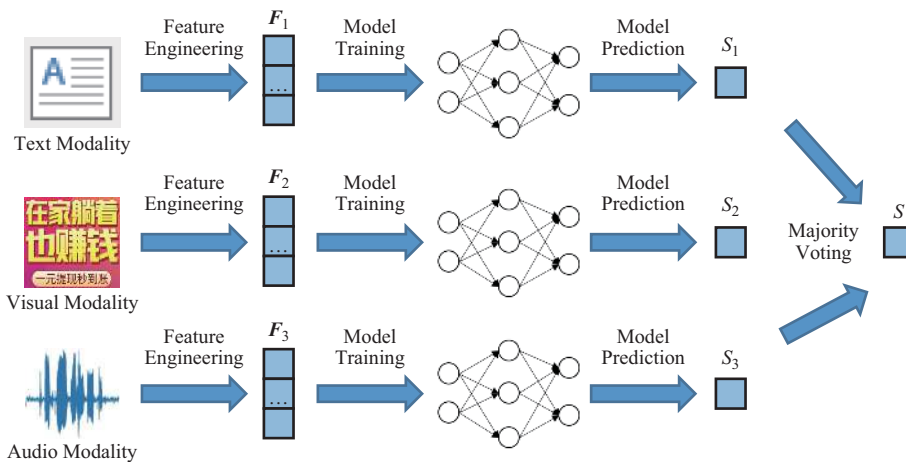


FIGURE 9.9 Decision layer fusion based on voting of model scores across modalities.

- Soft Voting: Directly average or calculate a weighted average of S_1 , S_2 , and S_3 . The resulting value S is used as the final predicted probability of traffic fraud.

9.2.3.2 When There Are Numerous Types of Modal Information

When dealing with numerous types of modal information, the initial step of constructing models for each type can become extensive. In such cases, the stacking approach from ensemble learning can be adopted. These models are treated as features to build a secondary model, with its prediction serving as the final probability for determining malicious traffic. The decision-layer fusion based on stacking is illustrated in Figure 9.10.

First, construct features from comment text information to train model M_1 , predicting the probability S_1 that the traffic is fraudulent. Similarly, train distinct models sequentially using information from different modalities, ultimately training n models and obtaining n model scores (M_1 to M_n). Then, use these n model scores as features to train another model M . Finally, predict the model score S for the traffic using model M , which represents the probability that this traffic is fraudulent.

When performing decision-layer fusion based on the stacking approach, note that the training samples for models M_1 through M_n may be identical. However, the training samples for model M must differ from those used for models M_1 through M_n . Otherwise, label leakage may cause model M 's training to fail.

Decision-layer fusion often requires maintaining multiple models. When a single modality has highly rich feature dimensions, this fusion approach fully exploits the information across all modalities, thereby enhancing the detection recall rate for traffic fraud.

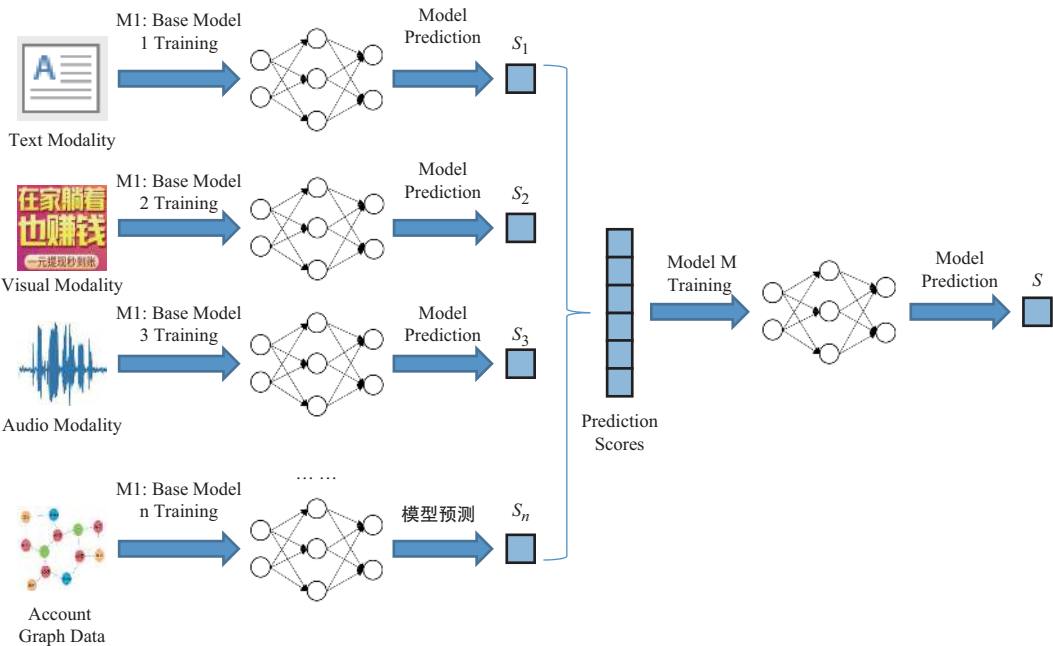


FIGURE 9.10 Decision layer fusion based on stacking approach.

9.2.4 Hybrid Fusion Approach

In practical traffic fraud detection, the core of multimodal hybrid solutions hinges on business scenarios, data sources, enforcement standards, and evaluation metrics. Generally, universal hybrid fusion approaches primarily consider the following four aspects.

- For same-format data: For instance, textual information from diverse sources—including user-posted comments, topic content, and personal profiles—shares a consistent format as text data. Combining such information facilitates determining whether an account is controlled by the black/gray industry. Therefore, these data types should be prioritized for integration at the foundational level, followed by unified feature engineering. This generates text-based features for subsequent modeling.
- For data of different modalities: Since textual information and complex account network relationship information differ fundamentally in form, they are typically not fused at the data layer. For data of different modalities, features specific to each modality can be generated and then fused at the feature layer.
- For scenarios with low feature dimensionality: Merging features at the feature layer allows for the integration of multimodal features. Subsequent unified modeling can then construct traffic fraud detection models.
- For scenarios with high feature dimensions: Fusion at the feature layer followed by unified modeling is feasible. However, when single-modal features are highly dimensional, modeling after feature-layer fusion may fail to extract the full value of the data in practical applications. It is generally recommended to fuse at the decision layer—constructing single-modal submodels to thoroughly analyze single-modal behaviors before unified ensemble modeling. Of course, in the case of high-dimensional features, both feature-layer and decision-layer fusion can be conducted in parallel. This is because both approaches aim to fully extract data value and enhance recall rates for traffic fraud.

In practical applications, considerations extend beyond the four points above. In real-world traffic fraud scenarios, hybrid solutions are rarely derived purely from theoretical frameworks. They require thoughtful integration with business challenges such as data sources, formats, and anomaly characteristics. The goal is to select the most cost-effective and efficient detection frameworks to achieve fraud detection objectives.

9.3 SUMMARY

This chapter introduces multimodal adversarial approaches to enhance traffic fraud recall rates based on common user data sources and formats within platforms. It first outlines prevalent multimodal data sources, then details different fusion methods across data collection, feature engineering, and model training workflows. Finally, it presents practical adversarial strategies—data-layer fusion, feature-layer fusion, decision-layer fusion, and hybrid fusion—drawn from real-world experience.

Novel Adversarial Approaches

Previous chapters systematically outlined practical methodologies for building traffic fraud countermeasure systems, covering device fingerprinting, CAPTCHA verification, rules engine, machine learning countermeasures, complex network countermeasures, and multimodal integrated countermeasures. Beyond these, countermeasure effectiveness or efficiency can be further enhanced through two additional dimensions:

- **Leveraging Third-Party Data Capabilities:** Current mainstream approaches directly utilize third-party SAAS models. These can be employed to construct rules or integrated as feature data into business models. While both methods enhance countermeasures to some extent, insufficient information fusion limits effectiveness gains. Under compliant data authorization, novel federated learning solutions can strengthen cross-platform data fusion, improving coordinated defense and control across platforms.
- **Boost computational efficiency for complex models:** As countermeasures against the black/gray industry evolve, models in practical scenarios often grow increasingly complex to fully extract data value and enhance coverage against traffic fraud. This complexity reduces prediction efficiency and increases resource costs. Knowledge distillation can migrate complex models to simpler ones, significantly boosting prediction efficiency without substantial performance degradation. This approach combines the efficiency of simple models with the approximate accuracy of complex ones.

Next, we address traffic fraud scenarios using two novel adversarial approaches to further enhance adversarial effectiveness and efficiency.

10.1 FEDERATED LEARNING

Federated learning, introduced by Google in 2016, is a novel approach designed to help Android device users update local models without compromising user privacy. In traffic fraud detection scenarios, when proprietary business data lacks distinct features,

third-party data becomes essential for early identification of suspicious traffic. Federated learning effectively leverages both business and third-party data for integrated modeling. The federated learning process is illustrated in Figure 10.1. While the private data of Company A, Company B, and Company C cannot be directly exchanged, they can train a federated model through encrypted training. This ensures data confidentiality while enabling comprehensive data fusion, thereby significantly enhancing traffic fraud detection capabilities.

10.1.1 Federated Learning Framework

From the perspectives of samples and features, federated learning can be categorized into horizontal federated learning, vertical federated learning, and federated transfer learning. Horizontal federated learning is suitable for scenarios where the features of each party's data are largely consistent but the samples differ significantly. Vertical federated learning is suitable for scenarios where the samples of each party's data are largely consistent but the features differ significantly. Federated transfer learning is suitable for scenarios where there is little overlap in both features and samples.

10.1.1.1 Horizontal Federated Learning

Horizontal federated learning, also known as sample-based federated learning, requires identical features across all parties while allowing for different samples. The principle of horizontal federated learning is illustrated in Figure 10.2. It primarily employs a master-slave training framework comprising an aggregator server and multiple participant servers. During model parameter training, participant servers handle model training on their

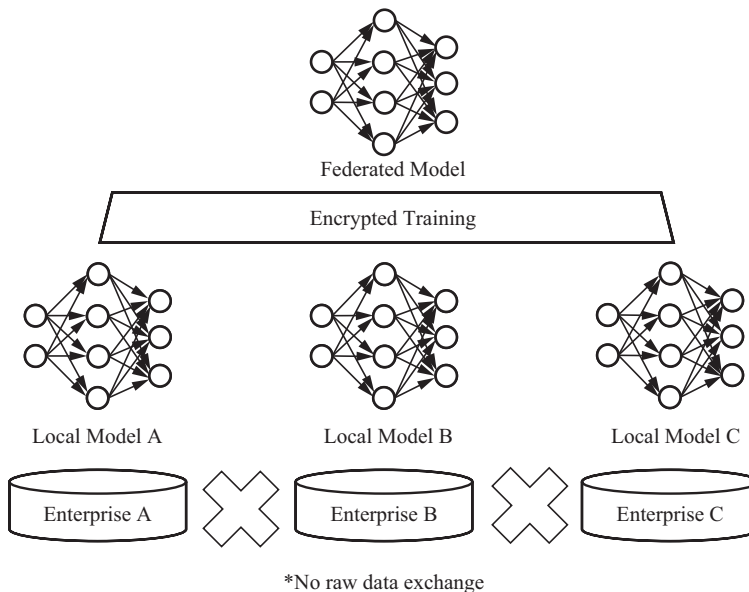


FIGURE 10.1 Process of the federated learning solution.

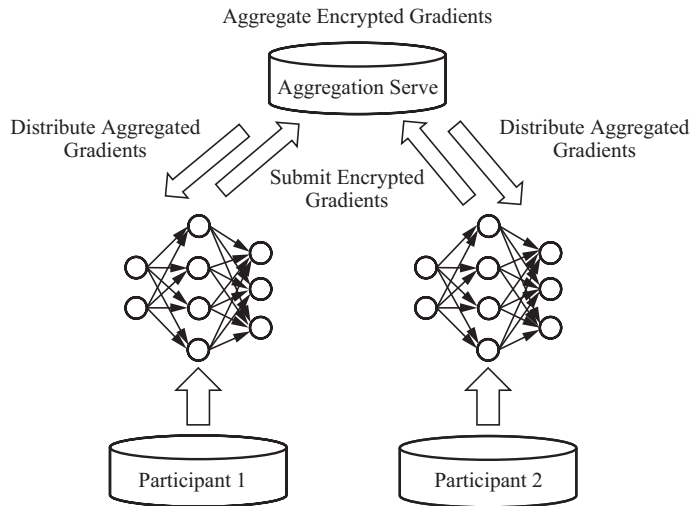


FIGURE 10.2 Principle of horizontal federated learning.

respective data, while the aggregator server collects and distributes training gradients. The core process involves the following four steps:

1. Participants train models using their own data, encrypt the resulting gradient information, and transmit the encrypted gradients to the aggregator server.
2. The aggregator server aggregates the encrypted gradients transmitted by each participant to generate an aggregated encrypted gradient.
3. The aggregation server transmits the generated aggregated encrypted gradient to each participant. At this point, all participants receive the identical aggregated encrypted gradient.
4. Participants decrypt the aggregated encrypted gradients, update their local models, and proceed to the next training iteration until model training is complete.

10.1.1.2 Vertical Federated Learning

Vertical federated learning, also known as feature-based federated learning, requires consistent data samples across parties while allowing for differing features. Its principle is illustrated in Figure 10.3. Vertical federated learning comprises two parts: first, individual encrypted alignment among parties, aligning individuals without revealing shared identities to select common samples; second, training corresponding models based on these common samples.

10.1.1.3 Federated Transfer Learning

Federated transfer learning is applicable when parties' data exhibit no overlap in either samples or features. In such cases, neither horizontal nor vertical federated learning can identify common samples across participants' data. If participants' data structures are similar, transfer learning approaches can address the challenges of insufficient samples and labels.

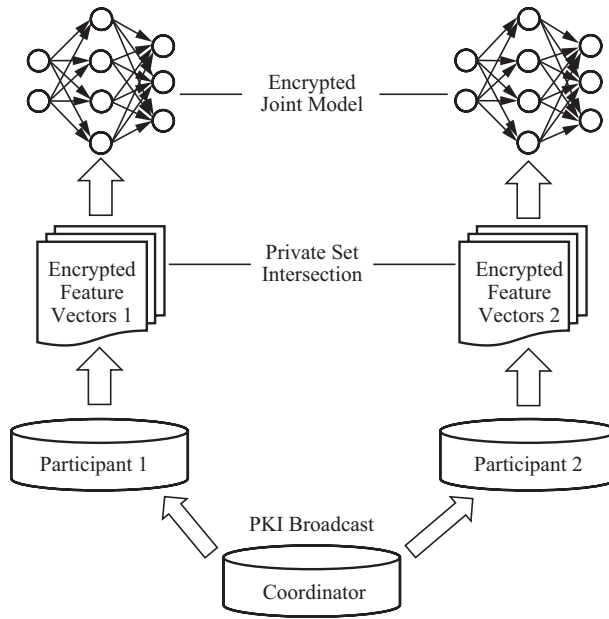


FIGURE 10.3 Principles of vertical federated learning.

10.1.2 Detection Effectiveness for Anomalous Traffic

In anomaly traffic detection scenarios, black/gray industry actors reuse resources like devices and accounts across multiple platforms. While sample overlap may exist across platforms, feature dimensions vary due to differing contextual environments and user profiling. This scenario requires feature-level information supplementation, making vertical federated learning suitable for integrating external data sources to enhance anomaly detection effectiveness.

Using the vertical federated learning solution within the FATE framework as an example, we validate the effectiveness of building anomalous traffic detection models by integrating third-party SAAS scores with proprietary features, as well as by using vertical federated learning to combine third-party features with proprietary features. Figure 10.4 compares the federated learning model-building approach with the SAAS score-incorporated model-building approach. Compared to the fusion of proprietary data and third-party SAAS scores, the fusion of proprietary data and third-party data yields over a 5% improvement in the AUC metric on the test set.

10.2 KNOWLEDGE DISTILLATION

Knowledge distillation was proposed by Hinton in his 2015 paper and is based on the teacher–student model concept. The principle of knowledge distillation is illustrated in Figure 10.5. First, a complex model is trained as the teacher model. Next, a simpler model is trained to fit the teacher model’s output as the student model. Finally, the less complex student model is deployed as the final model, achieving reduced computational resource consumption and easier model deployment.

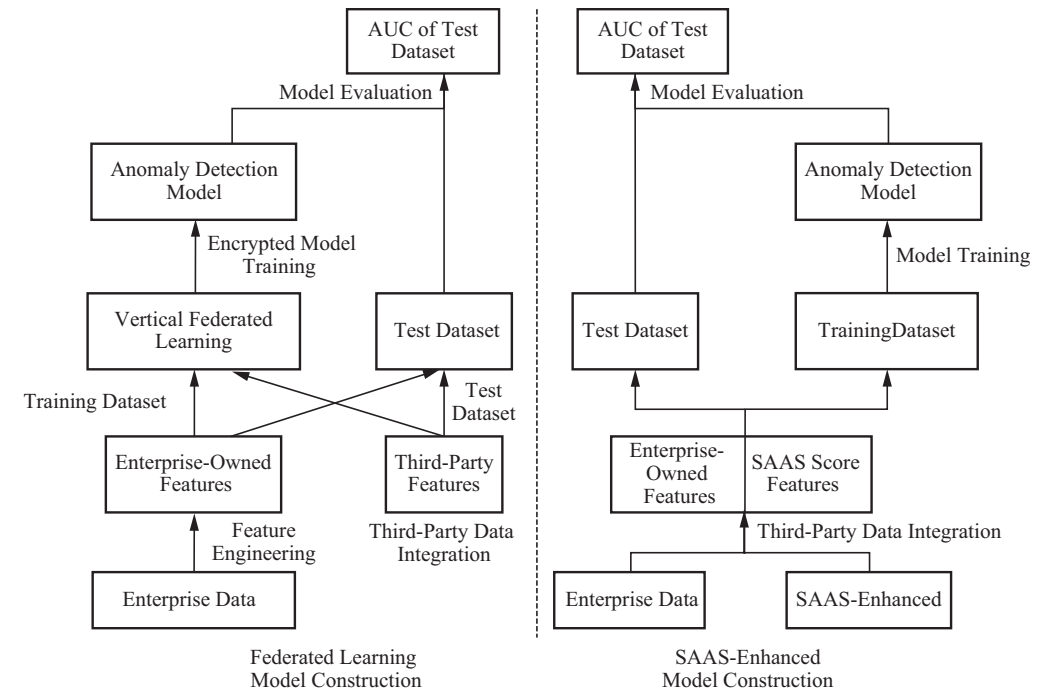


FIGURE 10.4 Comparison between federated learning model construction and SAAS score-integrated model construction.

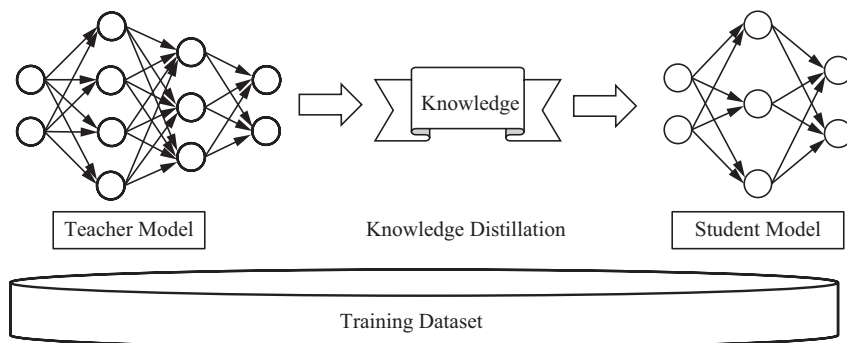


FIGURE 10.5 Principle of knowledge distillation.

Knowledge distillation simplifies complex models from traffic adversarial learning into simpler models suitable for real-time system deployment. It achieves high-performance outputs while maintaining comparable accuracy. The following sections focus on the knowledge distillation framework.

10.2.1 Knowledge Distillation Framework

The teacher–student model-based knowledge distillation framework comprises three main modules. The first two modules train the teacher model and student model, respectively,

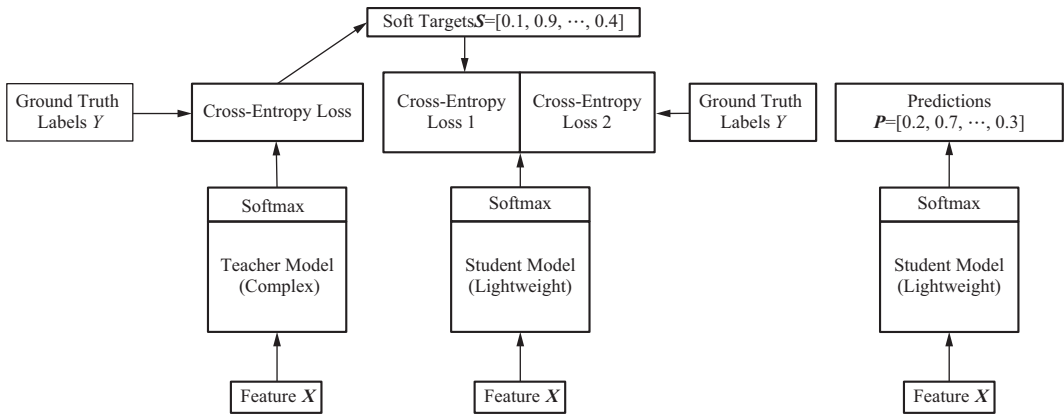


FIGURE 10.6 Knowledge distillation framework based on the teacher–student model approach.

while the final module is the prediction model, which aligns with the trained student model. The teacher–student model-based knowledge distillation framework is illustrated in Figure 10.6.

The first model in the framework is the teacher model, constructed based on a complex architecture. Its training objective is the sample label Y , where “0” denotes normal traffic and “1” denotes anomalous traffic. A cross-entropy loss function is then constructed based on the sample labels. By optimizing this cross-entropy loss function, optimal model parameters are obtained. Predictions are made on pre-segmented samples using the Softmax function to obtain the probability of each sample belonging to each class label, yielding the soft target S . The soft target S consists of continuous probability values between 0 and 1.

The second model in the framework is the student model, built upon a simpler architecture. Cross-entropy loss function 1 is based on the soft target S predicted by the teacher model, while cross-entropy loss function 2 is based on the sample labels Y . A new loss function L is then constructed by weighting and summing cross-entropy loss functions 1 and 2. By optimizing this combined loss function, the optimal model parameters are obtained. The loss function L is calculated as follows, where a is the weighting coefficient for the soft target-based loss function. A larger value of a indicates a greater proportion of the soft target-based loss function in the overall loss function, leading the model training to prioritize optimizing the soft target-based loss function. Thus, the core of knowledge distillation lies in constructing a novel loss function to optimize a simple student model. This enables the student model to achieve the predictive performance of the teacher model while maintaining its simplicity.

$$L = a \times L^{\text{soft target}} + (1 - a) \times L^{\text{sample label}}$$

The essence of the teacher model guiding the student model lies in introducing a new soft target S . In practical operations, black/gray industry traffic is characterized through user behavior. However, such activities exhibit varying degrees of severity—for instance, logging into two accounts versus five accounts on a device yields inconsistent assessments of

black/gray industry activity. Therefore, leveraging the probability value of soft target S to characterize the malicious attributes of target samples enables greater precision. By jointly optimizing the new loss function using the more precise soft target S for characterization of the black/gray industry and the original labels Y (offline 0/1 values) from the same data-set, we can obtain a highly effective student model.

The third module involves deploying the trained student model in production environments for predictions in offline systems or real-time online systems. Since student models typically involve simpler architectures with lower complexity, their prediction performance is exceptionally high, generally meeting the performance requirements of real-time online systems.

The differences between teacher and student models are summarized in Table 10.1.

10.2.2 Steps for Anomalous Traffic Detection

The implementation plan for knowledge distillation in anomalous traffic detection scenarios can be divided into five steps. The anomaly traffic detection process based on knowledge distillation is shown in Figure 10.7.

10.2.2.1 Training the Teacher Model

Construct the teacher model using collected profiling data and sample labels. When building complex anomalous traffic detection models earlier, we mentioned stacked models. These models are highly complex, requiring the construction of M primary XGBoost discriminative models and one secondary XGBoost discriminative model—meaning $M + 1$ models must operate simultaneously for discrimination.

TABLE 10.1 Differences Between Teacher and Student Models

Model Name	Teacher Model	Student Model
Model structure	Large model, more complex	Smaller model, simpler structure
Number of parameters	High	Fewer
Loss function composition	Based on sample labels	Sample labels and soft targets
Model response	Slow	Fast
System performance requirements for deployment	High	Low

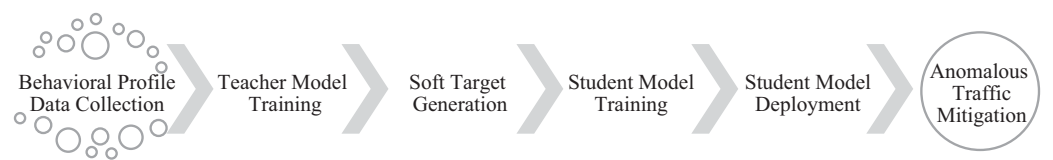


FIGURE 10.7 Knowledge distillation-based anomalous traffic detection process.

10.2.2.2 *Generating Soft Targets*

Using the trained stacked anomaly detection model, we classify another set of samples to obtain the predicted probability values for each sample belonging to anomalous traffic. These predicted probability values are then used as soft targets.

10.2.2.3 *Training the Student Model*

Combine the sample's original label with the newly predicted soft target to generate a fused loss function. Train the student model based on this fused loss function. Here, a single XGBoost discriminative model is used as the student model.

10.2.2.4 *Deploying the Trainee Model*

Deploy the trained single XGBoost discriminative model. The deployment and maintenance costs of a single XGBoost discriminative model are significantly lower than those of complex models, and its discrimination efficiency is higher.

10.2.2.5 *Blocking Anomalous Traffic*

Use the single XGBoost discriminative model to identify anomalous traffic in real time and intercept such traffic.

10.2.3 Effectiveness of Anomalous Traffic Detection

The effectiveness of anomalous traffic detection is evaluated based on two key metrics: the recognition effect on the test set, measured by the AUC metric, and traffic classification efficiency, which compares the number of traffic instances classified per minute under identical hardware performance.

We constructed anomalous traffic detection models using three approaches: direct XGBoost implementation, stacking-based model, and knowledge distillation-based model. The AUC of the knowledge distillation-based model closely matched that of the complex stacking model and performed similarly to a single XGBoost discriminative model.

10.3 SUMMARY

This chapter addresses two typical challenges in traffic fraud scenarios: maximizing the effectiveness of third-party data and improving the operational efficiency of complex models. It proposes mature, novel adversarial solutions developed in recent years—federated learning and knowledge distillation. The chapter also briefly compares the validation results of the federated learning framework and the knowledge distillation framework in specific traffic fraud scenarios.

Operational Framework

Previous chapters systematically covered the fundamentals of traffic security, insights into traffic risks, traffic data governance, and the evolution of traffic anti-fraud technologies. At this point, we can address the security challenges encountered in real-world scenarios. However, to ensure the healthy and stable operation of anti-fraud systems, a robust operational framework must be established to support system management. This chapter focuses specifically on building an operational system for traffic scenarios.

In traffic fraud identification scenarios, the four core modules of the operational system are illustrated in Figure 11.1. The operational system is primarily divided into stability operations, false-positive handling, user feedback processing, and alert handling.

- **Stability Operations:** Monitors traffic fluctuations and interception rates at the service layer. When traffic volumes undergo sudden changes, it promptly traces the root cause and alerts the responsible parties. It conducts comprehensive stability monitoring of foundational log data, features data, and models at the data layer to ensure data integrity.
- **False-Positive Prevention:** Implements protective mechanisms around model construction, adding a safeguard layer before model output to prevent high-quality users from being mistakenly flagged. Models with high misclassification rates receive immediate emergency handling to minimize the impact on business traffic.
- **User Feedback Handling:** User feedback is crucial for model iteration and improvement. This process primarily addresses user reports and appeals. Reports occur when users believe an issue exists but the model fails to flag it as malicious, while appeals arise when users believe no issue exists but the model incorrectly flags content as malicious. Generally, low model coverage tends to trigger user reports, while low model accuracy often leads to user appeals.
- **Alert Handling:** For the aforementioned stability monitoring, false-positive prevention, and user feedback processes, anomalies detected require incident classification.

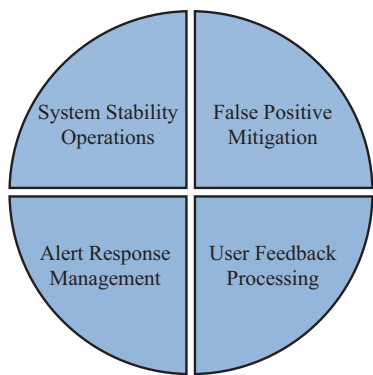


FIGURE 11.1 Four core modules of the operational system.

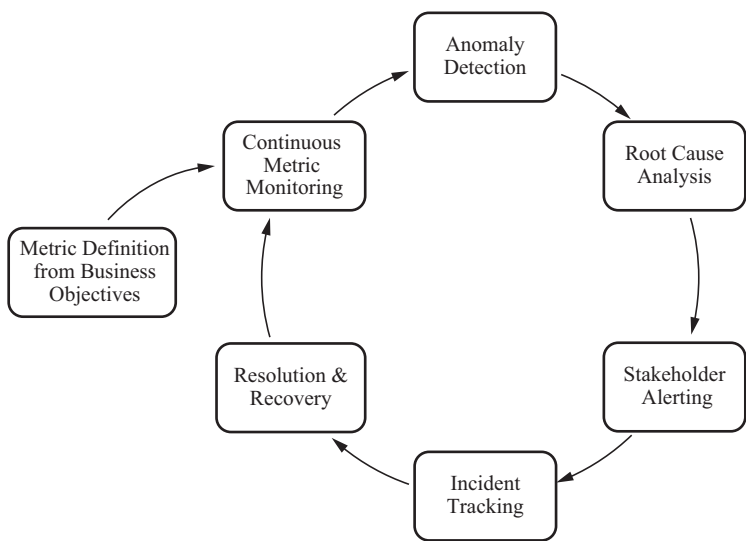


FIGURE 11.2 Operational system process for anomaly handling.

Alerts are triggered at varying severity levels based on the incident’s impact, with ongoing tracking of resolution efforts. If an incident remains unresolved within the predefined timeframe, it is escalated until fully resolved.

Typically, the operations system maintains a comprehensive anomaly handling workflow. This process begins by abstracting business metrics for monitored operations. These metrics are then continuously monitored, triggering immediate alerts upon detecting issues. Alerts are promptly traced to their source and escalated to responsible parties. The alert cycle only concludes once the problem is resolved and metrics return to normal. The operational system’s anomaly handling process is illustrated in Figure 11.2. The system abstracts business focus points into metrics for daily monitoring, capturing anomalies, tracing their origins, alerting responsible parties, and tracking alerts until the issue is resolved and metrics return to normal.

Next, we will focus on introducing the methodology for building the four core modules of the operations system and explain common core operational metrics through practical examples. In actual business scenarios, additional monitoring metrics can be derived by applying the same principles to specific issues.

11.1 STABILITY OPERATIONS

Stability is the minimum requirement for business teams using model services. If a model service is unstable, even with excellent performance, business teams will hesitate to adopt it long term. Stability monitoring typically encompasses two aspects: first, service stability monitoring tracks business traffic trends to ensure the model consistently and effectively predicts traffic. Second, data stability monitoring focuses on the data used for model classification and scoring, tracking changes in foundational data, features, and model scores to guarantee the model's validity.

Next, using a marketing promotion traffic fraud detection service as an example, we will introduce how to construct stability operation metrics.

The use case for this marketing promotion traffic fraud detection service involves a marketing campaign for a specific app product. First, the campaign attracts new users to download the app, register, and log in. Users then earn coins by checking in within the app, posting topics, and participating in discussions. Finally, they redeem these coins for physical prizes or virtual rewards like phone credit in the app's marketplace. However, such campaigns are prime targets for the black/gray industry. These groups mass-register fake accounts to simulate activity, accumulate large amounts of coins, redeem prizes for cash, and exit the system, ultimately causing losses for the campaign organizer. The fraud traffic detection service effectively identifies and blocks fake traffic generated by these syndicates throughout this process. The fraud traffic detection service comprises several subservices: fake registration detection targeting "account registration"; fake login detection targeting "login"; traffic manipulation detection targeting "check-ins, comments, and likes"; and "freeloader" detection targeting "gold coin collection and prize redemption."

To ensure the stable and effective operation of the traffic fraud detection service, a supporting operational system must be established to guarantee stable operations. Stable operations monitor the traffic fraud detection service from two perspectives: service stability and data stability. Upon detecting anomalies in monitoring, the system rapidly pinpoints the fault location, alerts relevant personnel, and oversees their resolution until monitoring metrics return to normal. The relationship between stability operations, fraud detection services, and marketing campaigns is illustrated in Figure 11.3. The stability operations system safeguards the normal functioning of fraud traffic monitoring services, which in turn ensures the smooth execution of marketing campaigns. The following sections detail the service stability and data stability metrics within stability operations.

11.1.1 Service Stability

As a critical component of stability operations, service stability primarily monitors whether each subservice within the fraud detection service is functioning correctly. First, it requires comprehensive traffic awareness to understand traffic trends and the actual interception

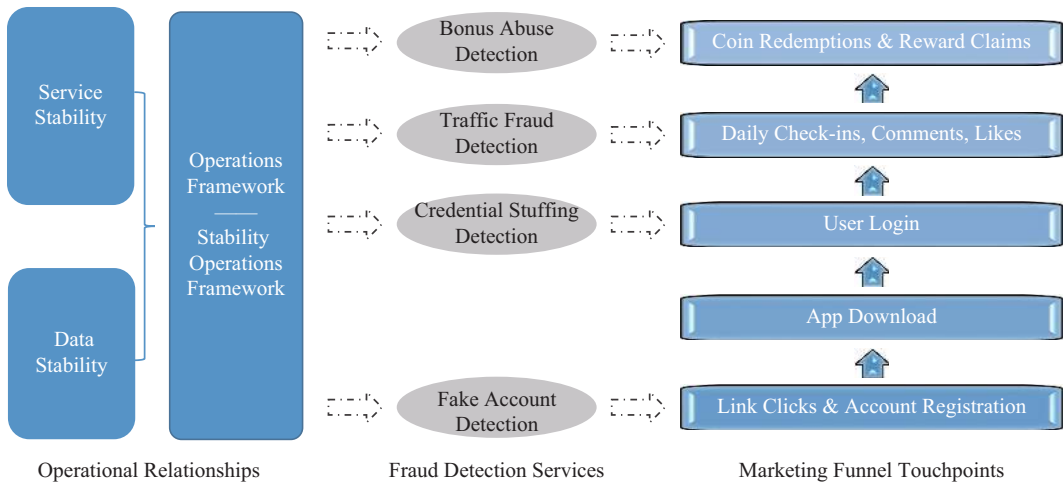


FIGURE 11.3 Relationship between stability operations, fraud detection services, and marketing campaigns.

status at each point. Second, monitoring whether servers provide stable services necessitates tracking the latency of returned traffic results. Finally, the foundation for stable server service delivery is a stable hardware environment, which requires monitoring of the hardware infrastructure. Below, we elaborate on how to ensure service stability from three dimensions: traffic trend monitoring, service latency monitoring, and hardware environment monitoring.

11.1.1.1 Traffic Trend Monitoring

For each stage of marketing campaigns, aggregate business traffic volumes by time period and conduct year-over-year (YoY) and month-over-month (MoM) comparisons. When significant traffic fluctuations occur, promptly alert relevant personnel.

First, aggregate business traffic from both input and output dimensions, then break it down by specific stages. For marketing campaigns, segment traffic into account registration, login, topic posting/commenting, and prize redemption flows, tracking business input and service output traffic over time. Regarding the granularity of business monitoring, time windows can be set at various intervals, such as 5 minutes, 10 minutes, 30 minutes, 1 hour, 4 hours, and 1 day. The subservice items, traffic classifications, and time window specifications for traffic monitoring are illustrated in Figure 11.4.

Second, for each stage of marketing campaigns, monitor traffic interception rates and present them via reports to model owners and business stakeholders. This enables real-time awareness and visualization of traffic interception. When interception rates experience sudden changes, alerts can be triggered for relevant personnel. For example, monitor the interception rate of registration traffic during user account creation. The interception rate for registration traffic from 00:00 to 24:00 on a certain day is shown in Figure 11.5. Around 4:00 AM, the interception rate exceeded 0.1%, representing a significant increase compared to the overall rate for that time period, thus triggering an alert. Subsequent post-incident

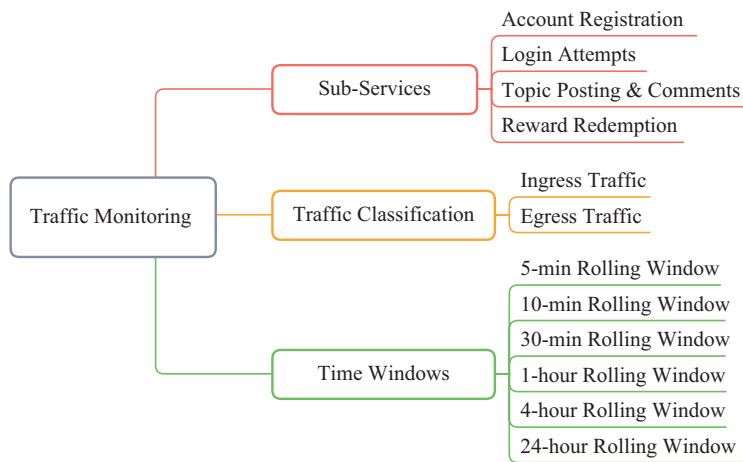


FIGURE 11.4 Explanation of traffic monitoring subservices, traffic categories, and time windows.

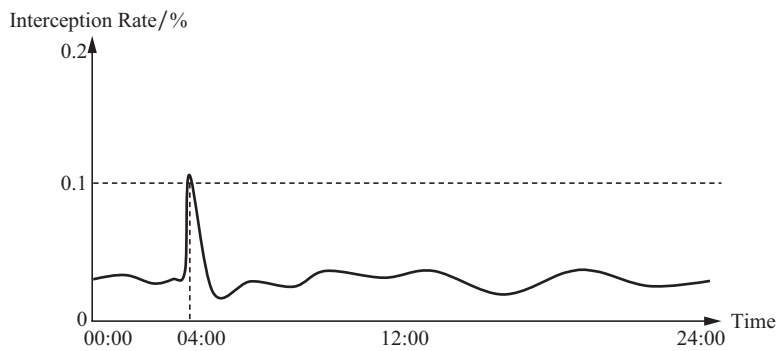


FIGURE 11.5 Interception rate of registration traffic from 0:00 to 24:00 on a given day.

analysis revealed that at 4:00 AM, a group from the black/gray industry was conducting bulk registrations of new users, causing the sudden spike in the interception rate.

11.1.1.2 Service Latency Monitoring

Service processing latency is a critical indicator of service stability, as it significantly impacts business requests and frontend responses. For instance, when a user performs a CAPTCHA verification on a frontend page, the backend service must promptly return whether the user is a genuine human. High latency in CAPTCHA recognition services severely degrades user experience. Therefore, it is essential to monitor the latency of fraud detection services across all stages of marketing campaigns.

Different services have varying latency requirements, with time-sensitive services demanding stricter limits. Consequently, latency thresholds differ across scenarios in traffic fraud identification for marketing campaigns. In ad fraud prevention, detecting fake clicks requires high latency sensitivity, typically with a maximum tolerance of 20 ms. In “freeloader” detection scenarios, identifying malicious group activities involved

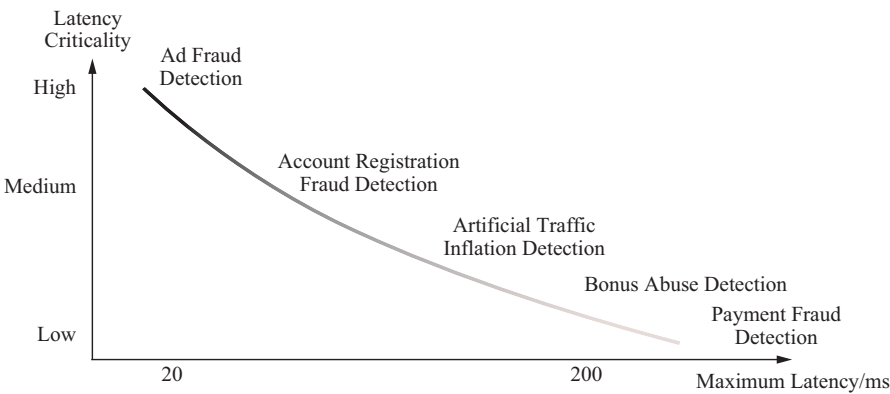


FIGURE 11.6 Latency requirements for different traffic fraud detection scenarios.

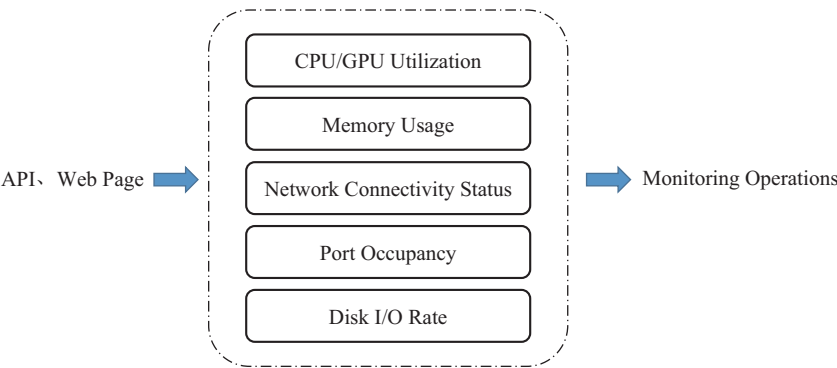


FIGURE 11.7 Server hardware environment monitoring operations.

in cash-out fraud has relatively lower latency requirements, typically with a maximum tolerance of 200 ms. The latency requirements for various traffic fraud detection scenarios are illustrated in Figure 11.6.

For services with high latency requirements, to meet service latency targets, measures such as enhancing database query performance, implementing caching, or upgrading server hardware and network bandwidth configurations are typically employed to accelerate data retrieval and transmission speeds.

11.1.1.3 Hardware Environment Monitoring

Stable hardware environments are fundamental to service stability. When hardware failures occur, services must have alerting and automatic migration capabilities. Hardware environment monitoring primarily covers five aspects: CPU/GPU utilization, memory usage, network communication status, port occupancy, and disk read/write rates. Issues in any of these areas pose a risk of server downtime. Monitoring server hardware performance can be achieved by directly accessing monitoring data via API interfaces or web pages. This data is then converted into metrics and presented in the monitoring operations module, as illustrated in Figure 11.7.

Abnormal hardware metrics can directly cause widespread service disruptions. For instance, excessively high CPU utilization may indicate a heavy CPU load, potentially leading to significant request delays or even server crashes. Therefore, when hardware metrics deviate from normal ranges, alerts are triggered to the operations platform for handling. The operations platform also performs preliminary assessments based on traffic trend indicators. If rapid traffic growth is detected, it automatically initiates scaling. If traffic remains stable, it preliminarily identifies a server failure and automatically activates backup service nodes while continuing to monitor changes in hardware environment metrics. If these automated operations fail to resolve the issue, the O&M platform escalates the alert for the operations and maintenance personnel to manually analyze and process. The alert process concludes and normal monitoring resumes only after hardware environment metrics return to normal.

11.1.2 Data Stability

Service stability monitoring primarily focuses on whether the traffic fraud service can reliably predict traffic patterns. However, the stability of prediction results requires analysis from the perspective of data stability. Here, “data” refers to all information used throughout the entire data modeling process: from collecting log data via tracking points, to generating high-quality data through data governance, to producing features data via feature engineering, to generating model scores by applying trained models to traffic data, and finally using model score data for traffic fraud detection. The entire data processing workflow is illustrated in Figure 11.8.

Monitoring the stability of log data, high-quality data, features data, and model score data primarily involves two aspects: data volume stability and data distribution stability. For data, both sudden changes in volume and shifts in distribution can cause significant fluctuations in service prediction results.

- **Data Volume Stability:** Monitoring data volume stability requires tracking the scale of log data, high-quality data post-governance, features data after feature engineering, and model scoring data across different time windows. Calculate YoY and MoM changes in data scale. Any YoY or MoM deviation triggers alerts to data owners. From log data collection via tracking points to model scoring for traffic fraud detection services, data flows through a serial processing pipeline. For instance, log data undergoes cleaning to remove anomalies before yielding high-quality data for subsequent feature engineering. Consequently, data volume progressively

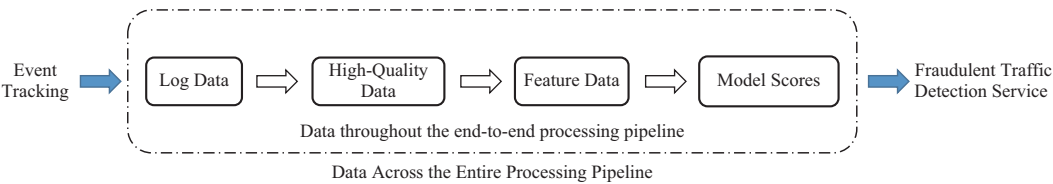


FIGURE 11.8 Full data processing workflow.

decreases during processing, with each stage incurring a data loss rate—defined as the percentage of processed data relative to the preceding stage’s input. Therefore, monitoring changes in overall data volume must be complemented by monitoring data loss rates at each processing stage. Excessively high loss rates should trigger alerts to the data owner.

- **Stability of Data Distribution:** To monitor the stability of data distribution, data must first be divided into buckets. The proportion of data in each bucket is then calculated, and the changes in the distribution of each bucket across two distinct periods are evaluated. The Population Stability Index (PSI) is used to assess and monitor distribution changes. A smaller PSI value indicates less variation in distribution between the two periods, signifying greater stability in the data distribution. Generally, a PSI value below 0.1 indicates a stable distribution. When PSI exceeds 0.25, it suggests a significant shift in the data distribution, requiring the data owner to verify if any data processing issues have occurred. For PSI values between 0.1 and 0.25, decisions should be made based on the specific business context. In marketing promotion anti-fraud scenarios, fake registration detection services are periodically impacted by black/gray industry groups registering accounts. Consequently, related log data and final model scores may exhibit relatively larger fluctuations. For instance, PSI values between 0.1 and 0.25 may still be deemed relatively stable. However, payment fraud prevention demands higher data stability standards; therefore, PSI values between 0.1 and 0.25 are classified as unstable. When data stability issues arise and root cause analysis is required, problems are typically traced backward through the data processing workflow based on traffic data. For instance, if the volume of model-scored data decreases, verify whether the preceding features data volume is normal. If the features data volume is normal, the issue lies in the model prediction scoring stage. Otherwise, continue tracing upstream until the root cause is identified, then implement targeted fixes.

11.2 FALSE-POSITIVE PREVENTION

In marketing promotion anti-fraud scenarios, model detection accuracy is critical. Substandard accuracy may falsely flag normal traffic, compromising user experience. Therefore, a defense system against model false positives must be established. During the development and deployment of traffic fraud detection models, the three-tier defense system against false positives is illustrated in Figure 11.9.

- The first line of defense is the offline validation phase: Before model deployment, offline validation assesses effectiveness by verifying the intersection between traffic flagged as fraudulent offline and suspected whitelisted traffic. For example, if the intersection ratio of flagged traffic is less than 0.1%, the model is deemed effective. Only after meeting this criterion is the model deployed. As shown in Figure 11.9, Models 2 and 3 failed offline testing and were returned to their respective owners for adjustments.

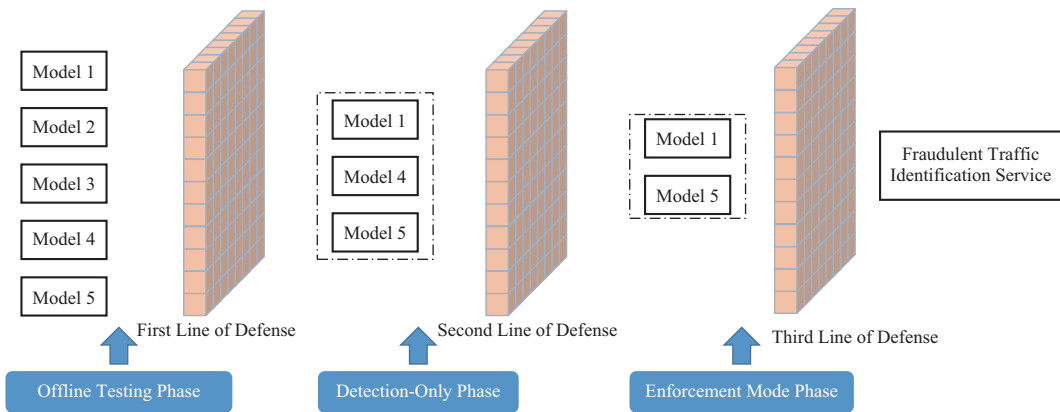


FIGURE 11.9 Three lines of defense against false positives.

- The second line of defense is online silent observation: After passing the first line of defense, the model is deployed to identify online traffic in real time without imposing penalties. At this stage, the model only performs classification without subsequent actions like blocking based on the results. Performance is then validated through real-time traffic monitoring, specifically by verifying the hit rate of traffic classified as black by the model against suspected whitelist entries. For instance, if the hit rate exceeds 0.1%, the model's performance is deemed unsatisfactory. Subsequently, models that fail to meet performance standards are temporarily taken offline for reanalysis. Once performance standards are met, the model is deployed for formal online penalties. As shown in Figure 11.9, Model 4 was temporarily taken offline for retraining due to its failure to meet online testing performance standards.
- The third line of defense is online suspected whitelist protection: Once the model goes live, it evaluates online traffic and processes it based on the evaluation results. Although the model has passed online performance validation, its accuracy may decline due to factors like significant changes in user behavior caused by product upgrades. Therefore, a third line of defense is needed to safeguard model accuracy and minimize the impact of misclassifications. During live operation, if Models 1 and 5 flag traffic as fraudulent but this traffic falls under the third line of defense's suspected whitelist, no immediate action is taken. Instead, a manual review process is triggered to verify whether it constitutes actual traffic fraud. When models frequently flag suspected whitelist cases, they should be proactively muted for silent operation, followed by manual analysis.

These three lines of defense protect the model and help prevent false positives to a certain extent. However, the effectiveness of these lines in preventing false positives depends on the validity of the suspected whitelist. The following sections will elaborate on the construction and automated operation of the suspected whitelist.

11.2.1 Suspected Whitelist

The purpose of anti-fraud traffic detection services is to identify fraudulent traffic and reduce losses for merchants engaged in marketing promotions. However, if such services mistakenly flag high-quality users, it can damage the platform's reputation and lead to user complaints. Therefore, constructing a suspected whitelist is crucial for establishing the three lines of defense against false positives in the model. The goal of building this whitelist is to protect the business's high-quality users.

This article focuses on constructing the suspected whitelist from three dimensions: IP addresses, devices, and accounts. IP addresses refer to network protocol addresses; devices denote various terminals in the network, such as mobile phones, personal computers, and servers; accounts refer to the unique identifiers each person uses across different projects, such as mobile phone numbers, WeChat IDs, QQ numbers, and user accounts set up within other apps.

- **IP Address Suspected Whitelist:** IP addresses belonging to well-known enterprises, public venues, and base stations are designated as entries for the suspected whitelist for protection.
- **Device Suspected Whitelist:** Categorized by device model, such as newly released high-end models or high-quality.
- **VIP Active Devices:** Carry a lower risk of exploitation by the black/gray industry.
- **Account Suspected Whitelist:** High-activity accounts, long-registered accounts, and VIP accounts provided by clients are protected as entries on the suspected whitelist.

11.2.2 Automated Operations

After establishing the suspected whitelist, whitelisted users can be deployed across three defense lines. This process can be automated through three modules: manual review of false-positive traffic, model deactivation during high false-positive rates, and suspected whitelist updates.

- **Manual Review of False-Positive Traffic:** Filter traffic flagged as malicious by online models using the third-line suspected whitelist. Block traffic not on the suspected whitelist, while traffic matching the suspected whitelist undergoes manual review. Final processing follows the outcome of the manual review.
- **Model Deactivation for High False Positives:** Track the rate of traffic confirmed as false positives after manual review. If this rate exceeds a set threshold (e.g., 0.1%), temporarily deactivate the model. The model owner must then reanalyze and iterate the model until accuracy meets standards before reactivating it.
- **Suspected Whitelist Updates:** Based on manual review outcomes, if traffic is confirmed as traffic fraud, corresponding updates to the suspected whitelist are required. Alternatively, suspected whitelist entries may be periodically updated offline through system analysis, identifying new high-quality VIP users or other channels.

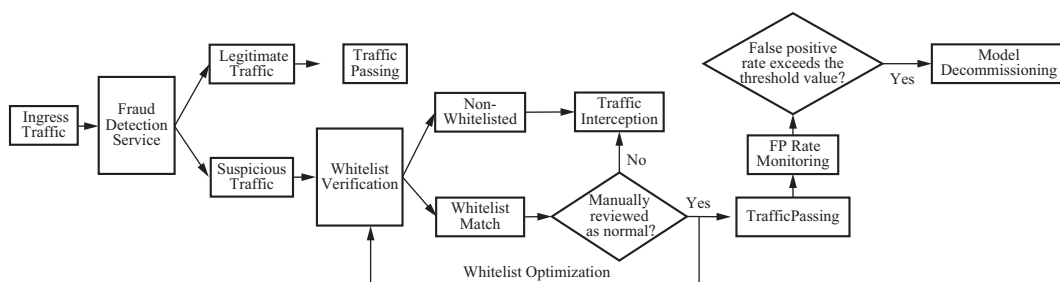


FIGURE 11.10 Automated false-positive prevention framework.

The automated false-positive prevention framework, as shown in Figure 11.10, primarily includes protecting traffic fraud from false positives, monitoring the misclassification rate of detection models and promptly decommissioning models with high misclassification rates, and adjusting and updating the suspected whitelist based on manual review results.

11.3 USER FEEDBACK HANDLING

To address misclassifications and false negatives in fraud models, it is essential to follow user appeals and reports. These feedback mechanisms enable further model performance monitoring and iterative refinement. Typically, an increase in user appeals indicates rising misclassification rates in live traffic fraud detection models, while a rise in user reports suggests potential coverage gaps in the model. The following sections detail the processing workflows for user complaints and user reports.

11.3.1 User Appeal Handling

When participating in marketing promotions, users may be flagged as suspicious traffic fraud due to questionable behavior and subsequently blocked from continuing the activity. In such cases, the operations system provides an appeal channel to protect legitimate users' interests. The appeal backend aggregates appeal data, links it to model information and relevant evidence, and forwards it to reviewers for assessment. Upon approval, reviewers either lift traffic blocks or uphold the original determination based on their findings. Simultaneously, normal traffic is treated as a false-positive sample. These samples are compiled and fed back to the model owner, who then adjusts the model based on the false-positive patterns.

However, appeal channels are also susceptible to exploitation by the black/gray industry. After traffic fraud is blocked, the black/gray industry may submit bulk, repetitive appeals in an attempt to “clean” their accounts. Such bulk and repetitive actions place significant pressure on reviewers. Therefore, it is essential to filter out malicious appeals from the black/gray industry before processing appeals.

The user appeal processing flow is illustrated in Figure 11.11, primarily comprising three modules: malicious appeal filtering, evidence correlation and manual review, and misjudgment monitoring.

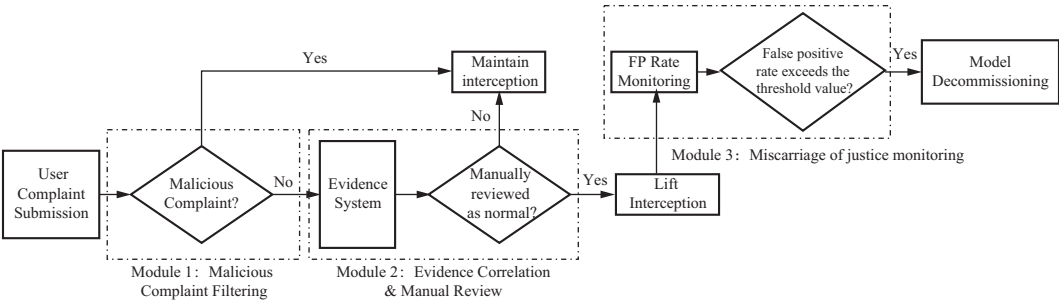


FIGURE 11.11 User appeal processing workflow.

- **Malicious Appeal Filtering:** Utilizing data submitted by the appellant and the account under appeal, this process employs a malicious appeal model to identify and filter out fraudulent appeals from the black/gray industry, thereby reducing the volume of subsequent reviews.
- **Evidence Correlation and Manual Review:** This module correlates the details of the blacklisting strategy applied to the reported traffic, the corresponding scores, and some basic dimensional information. This information is then provided to reviewers as evidence for blacklisting decisions, further improving review efficiency.
- **False-Positive Monitoring:** Normal traffic that is manually reviewed as legitimate is tracked as false positives. When model false-positive rates become excessive, the corresponding model is promptly decommissioned.

11.3.2 User Report Handling

Fraudulent activities by members of the black/gray industry not only cause financial losses to the platform but also disrupt normal users, potentially harming their interests. Consequently, affected users file targeted reports after experiencing fraud. The backend aggregates report data and forwards it to reviewers for assessment. Based on review outcomes, traffic is either blocked or released. Simultaneously, anomalous traffic is compiled as uncovered samples and reported to model owners. Model owners then enhance the model to identify and block such traffic based on the uncovered patterns.

However, reporting mechanisms can be exploited maliciously. Black/gray industry actors or certain users may fabricate false reports to maximize their own gains. For instance, they might create fake reports to disrupt the system's timely processing of legitimate reports. Therefore, malicious reports must be filtered out before processing to reduce the burden on subsequent reviews.

The user report processing flow is illustrated in Figure 11.12, comprising three main modules: malicious report filtering, manual review, and model iteration.

- **Malicious Report Filtering:** This module filters out malicious reporting behavior to reduce the volume of subsequent reviews.

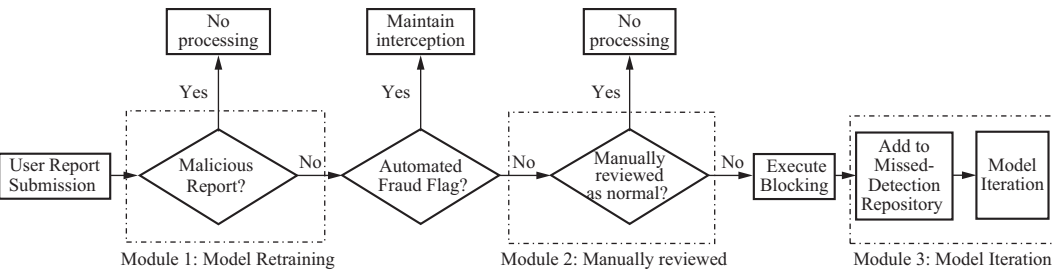


FIGURE 11.12 User report processing workflow.

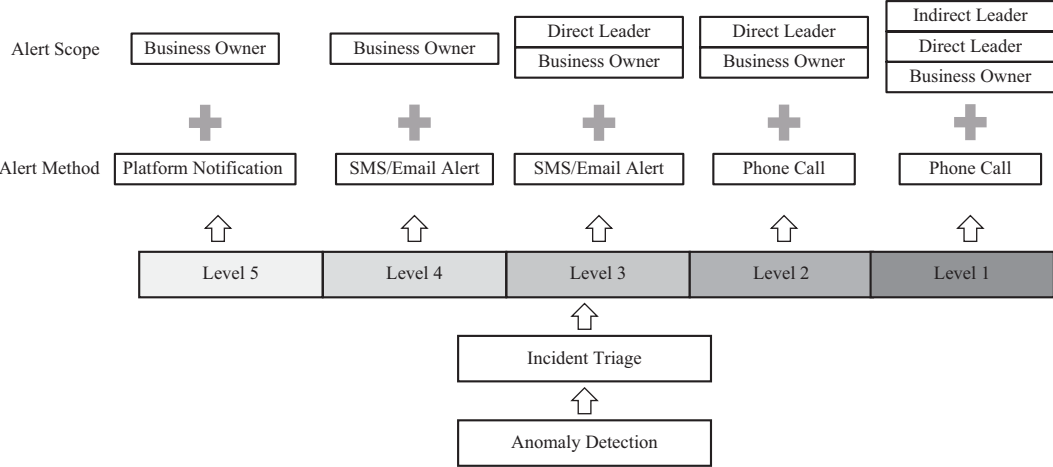


FIGURE 11.13 Fault classification and handling process.

- **Manual Review:** Before human intervention, the system can automatically identify and intercept highly suspicious accounts. Accounts with lower suspicion levels proceed to manual review, where traffic fraud identified during the review is blocked.
- **Model Iteration:** Add samples missed by manual review to the missed sample repository and utilize them for subsequent model iterations to enhance detection coverage.

11.4 ALERT HANDLING SYSTEM

Section 11.3 covers stability operations, false-positive prevention, and user feedback handling—all three modules aim to detect issues in the anti-fraud traffic detection service early. To ensure efficient resolution once problems arise, a comprehensive alert handling system must be established. This system first classifies issues by severity and then tracks their resolution throughout the entire process until they are properly addressed.

When an issue arises, the system assigns it a severity level. Different alerting methods and notification scopes apply based on these levels. More critical incidents trigger more direct alerts with broader coverage. This tiered approach ensures high-priority alerts receive prompt attention while minimizing unnecessary disruption to responsible personnel. The incident escalation process is illustrated in Figure 11.13. Upon detecting metric anomalies,

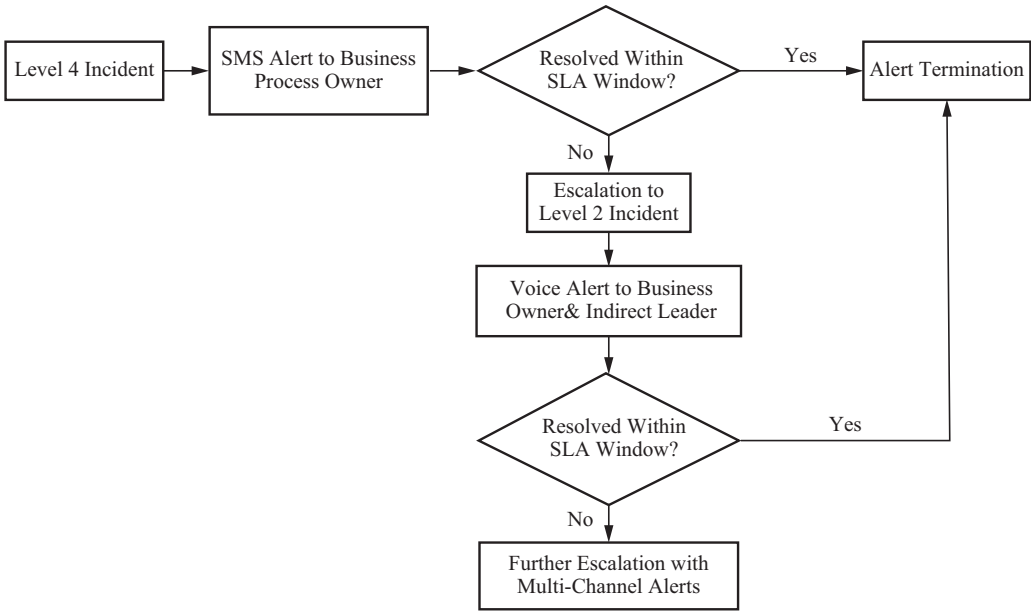


FIGURE 11.14 Fault tracking and alert escalation system.

incidents are first categorized into five severity levels (Level 5 to Level 1, with Level 1 being the most critical). Different alerting methods and scopes are then applied based on the assigned severity level. Alarm methods escalate from least to most urgent: platform notifications, SMS/email alerts, and phone calls. Alarm scope expands from smallest to largest: business owner, direct supervisor, and indirect supervisor.

The alert handling system continuously tracks issue resolution progress. If an issue remains unresolved within the preset timeframe, the alert escalates until the problem is fully resolved, at which point the alert cycle concludes. The fault tracking and alert escalation system is illustrated in Figure 11.14. Initially classified as a Level 5 incident, it triggers an SMS alert to the business owner. The system then assesses whether the incident is resolved within the stipulated timeframe. If resolved, the alert concludes; otherwise, it escalates to a Level 2 incident. Specific escalation rules are determined based on actual business requirements. After escalating to a Level 2 incident, a telephone alert is sent to the business owner and indirect leader. The system then determines whether the incident was resolved within the specified timeframe. If resolved, the alert is closed; otherwise, the incident continues to escalate and trigger alerts until it is properly resolved.

11.5 SUMMARY

This chapter outlines the process for constructing the operational framework for traffic fraud scenarios. The operational framework comprises four core components: stability operations, false-positive handling, user feedback management, and alert processing systems. Establishing this framework ensures the effective operation of the entire traffic fraud system, enabling rapid and stable fraud detection to safeguard system health.

Knowledge Intelligence Mining and Application

Previous sections covered the background of traffic fraud, technical details of countermeasures including device fingerprinting, CAPTCHA verification, rules engine, machine learning-based countermeasures, complex network-based countermeasures, and multimodal integrated countermeasures, and how the operational framework ensures refined management of these technical solutions. However, to achieve earlier fraud detection and prevention, timely in-process response, and post-event analysis and summarization, a complementary knowledge intelligence system must be established to inform countermeasure solutions and operational frameworks.

This chapter focuses on knowledge intelligence relevant to traffic fraud scenarios, illustrating the knowledge intelligence mining process through practical case studies.

The primary sources of intelligence data for black/gray industries are based on third-party data, open-source data, and operational data. For operational data, a comprehensive knowledge intelligence system can be built from two dimensions: criminal syndicates and individual points. The framework for traffic fraud knowledge intelligence mining, as illustrated in Figure 12.1, encompasses three dimensions of intelligence development: first, the upstream perspective of the black/gray industry composition, specifically intelligence mining on criminal syndicates. Second, the individual behavior perspective of black/gray industry actors, focusing on anomalous traffic. Third, the mining and monitoring perspective of upstream resources, involving surveillance of the black/gray industry's industrial chain resources. The framework ultimately outputs multiple types of fraud intelligence, including threat event intelligence, countermeasure techniques intelligence, profiling/pattern intelligence, fraudulent data intelligence, and black market pricing intelligence.

12.1 BLACK/GRAY INDUSTRY GANG INTELLIGENCE MINING

To maximize profits from traffic, black/gray industries often operate in organized groups, forming specific relationships across multiple hidden dimensions—such as multi-account

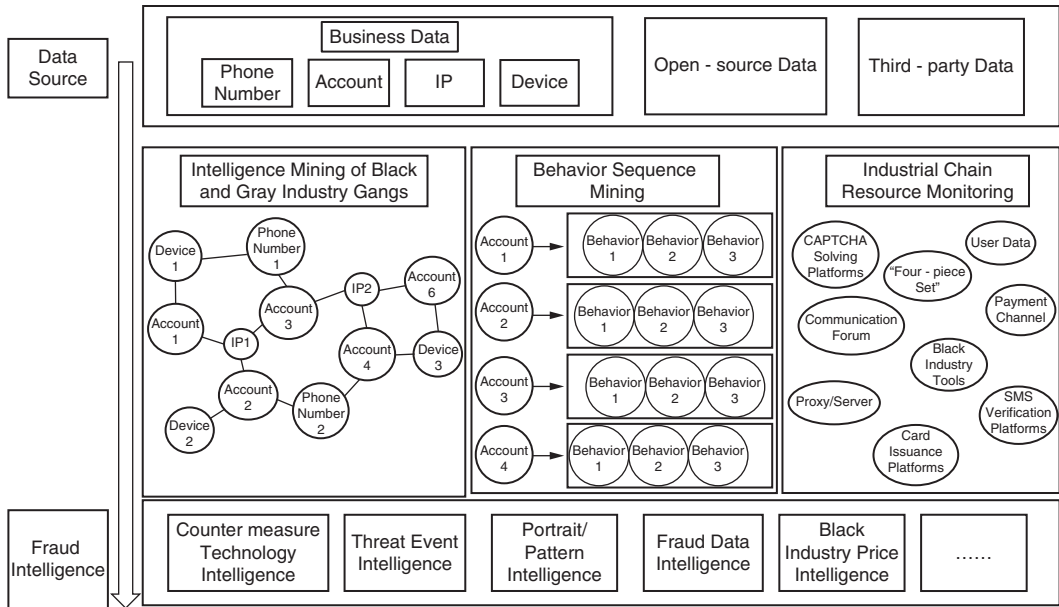


FIGURE 12.1 Framework of the traffic fraud knowledge intelligence mining solution.

behavior on a single device. Analyzing these groups enables earlier detection of criminal activities, enhancing the coverage of detection models. By controlling these accounts before they bulk-claim coupons for resale, business losses can be significantly reduced.

The analytical approach for mining black/gray industry gang intelligence typically involves identifying clusters of abnormal accounts. Clustering manifests in two dimensions: environmental clustering and attribute clustering. The clustering patterns for gang intelligence mining are illustrated in Figure 12.2.

Furthermore, intelligence mining approaches vary across different business fraud scenarios: first, they depend on available data sources—whether derived from proprietary business data or third-party data. Second, they hinge on big data platform infrastructure, as illustrated in the diagram of platform facilities analysis. Next, from a technical perspective, we will elaborate on intelligence mining methods through two prevalent fraud scenarios.

12.1.1 “Freeloading” Intelligence

Background of a Common “Freeloading” Case: To drive massive downloads of its e-commerce app and stimulate user spending, a platform launched a promotion for premium liquor using a “flash sale” strategy. Immediately after launch, the discounted liquor sold out instantly. However, subsequent user growth metrics fell far short of expectations, and external feedback indicated that scalpers had snapped up the promotional stock.

The following analysis examines actual cases to dissect activities of the black/gray industry, distilling patterns in business rules, cheating methods, and data behavior. This establishes intelligence methodologies for early anomaly detection and guides the continuous

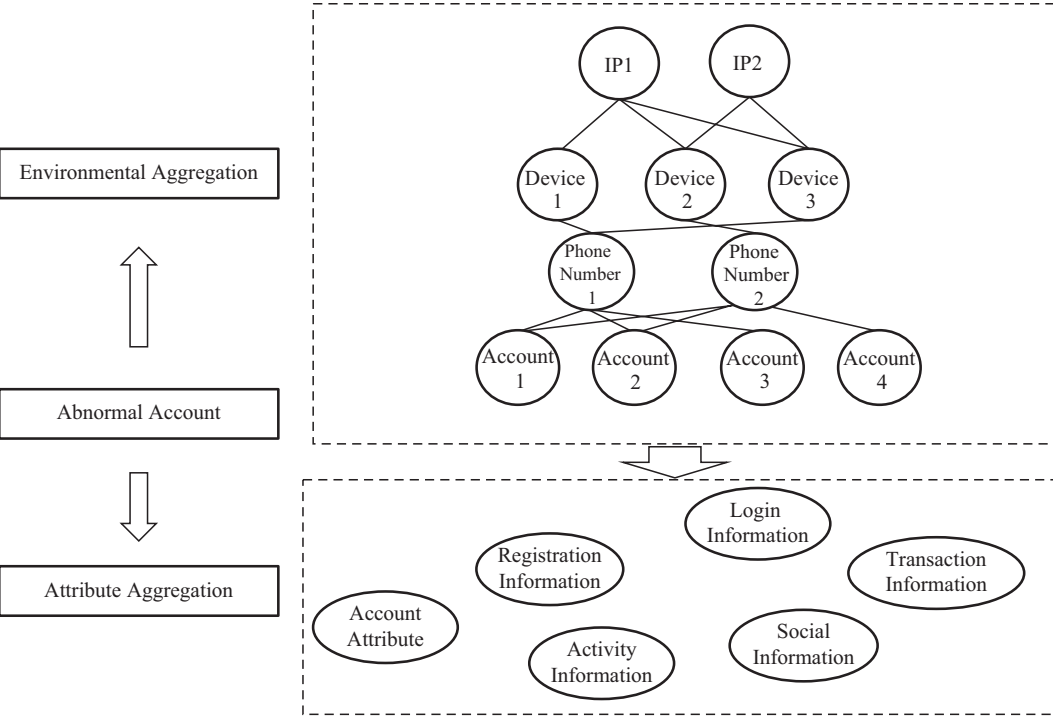


FIGURE 12.2 Clustering patterns in gang intelligence mining.

refinement of monitoring systems. While anomaly monitoring methods and steps may vary slightly across different business scenarios and datasets, they generally encompass the following phases.

12.1.1.1 Identify Anomalies in Current Campaigns

Filter accounts that successfully participated in the flash sale at the designated time and track their multiple participation timestamps. Identify suspicious accounts based on minimal time differences between actions such as clicking the product, entering the CAPTCHA, filling out the order form, and completing payment, as illustrated in Figure 12.3.

12.1.1.2 Expanding Suspicious Information

By examining the login IP records of these suspected abnormal accounts, we find that the accounts cluster around a fixed set of IP addresses, which also log in to other suspicious accounts. Furthermore, the device parameters used to initiate requests by these accounts exhibit clustering patterns. While the mobile phone numbers used to register the accounts are geographically dispersed, the addresses provided during order completion cluster in a few identical regions. Through these dimensional anomalies, we can further expand the list of suspicious accounts, devices, IPs, mobile phone numbers, and addresses. The propagation process for devices, mobile phone numbers, accounts, IPs, and addresses is illustrated in Figure 12.4.

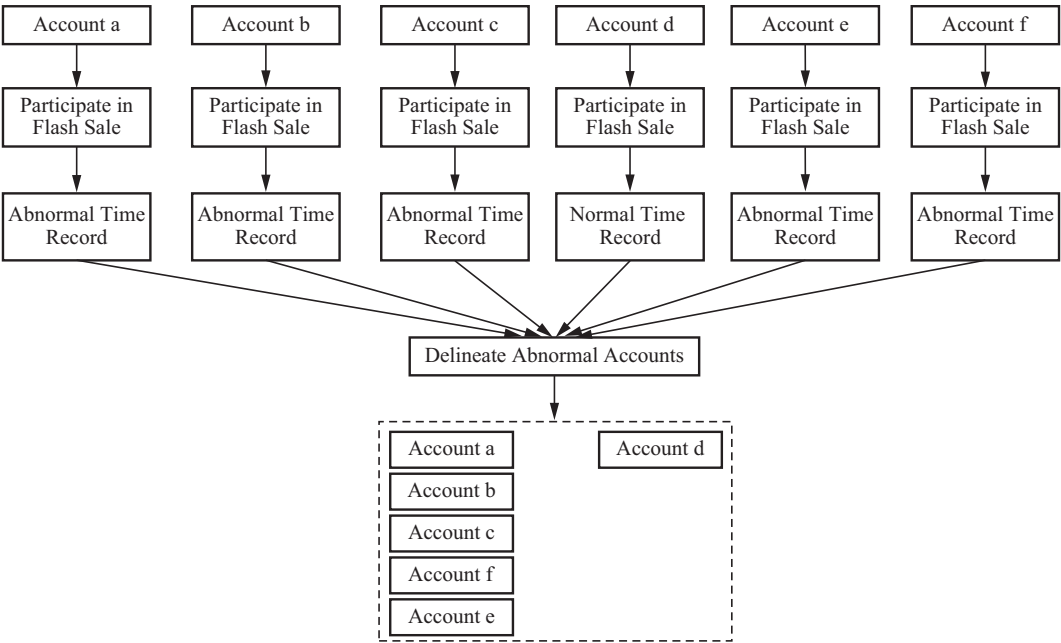


FIGURE 12.3 Identifying a group of suspicious abnormal accounts.

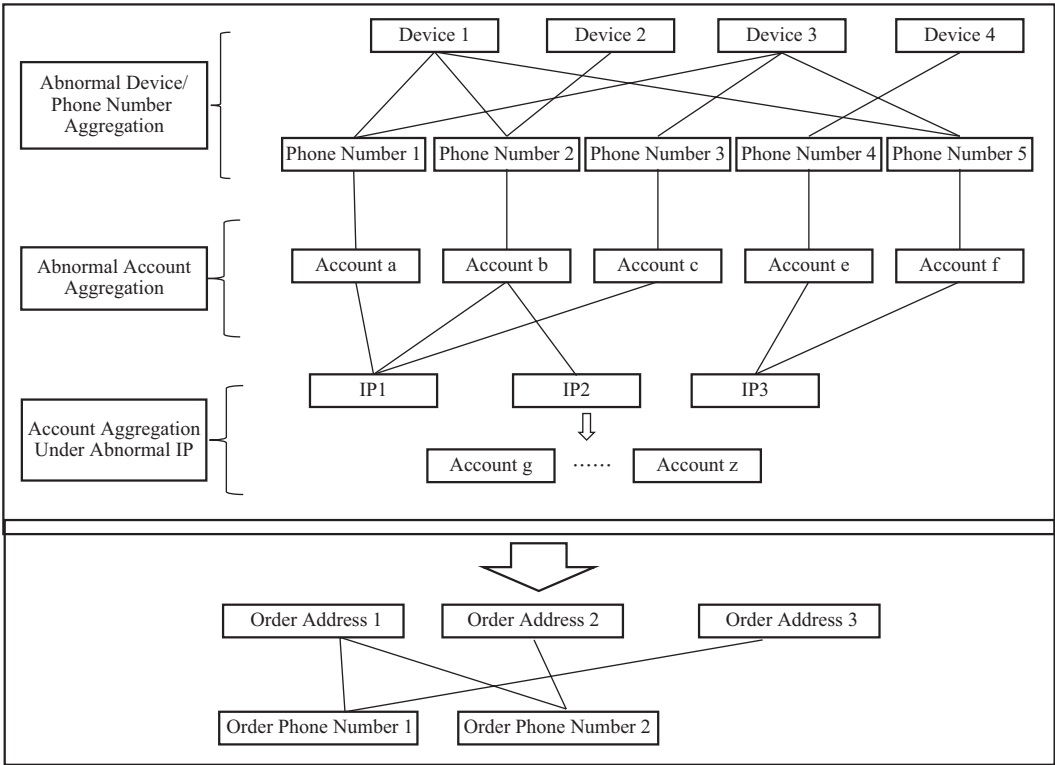


FIGURE 12.4 Diffusion process for devices, mobile phone numbers, accounts, IPs, and addresses.

12.1.1.3 Retracing Historical Data for Evidence Collection and Expansion

Further analysis of the additional suspicious accounts identified in the second step revealed highly similar behavioral patterns across all stages from registration to usage. Historical data analysis of the correspondence between order addresses and mobile phone numbers for this group of accounts showed that these addresses and mobile phone numbers appeared multiple times in the platform’s early activity records. Additionally, the login information for these accounts exhibited significant overlap, such as closely aligned login times. Historical activity associations of accounts belonging to the “wool-pulling” fraud ring are shown in Figure 12.5.

12.1.1.4 Evidence of Anomalies through Multi-Dimensional Data Analysis

From the perspective of analyzing request data at a more fundamental level, during the flash sale preparation phase, these accounts utilized the same API to construct a large number of highly similar parameter request links for packet transmission.

12.1.1.5 Methodology for Summarizing the Analysis Process

By analyzing foundational data such as platform accounts, mobile phone numbers, devices, IP addresses, and locations involved in fraudulent activities, and by tracing historical behaviors from account registration, login, and account maintenance activities, a methodology for generating knowledge intelligence can be established. The types of intelligence generated through intelligence mining are illustrated in Figure 12.6.

12.1.2 Comment Farming Intelligence

Comment manipulation is a common form of traffic inflation in platform UGC, where paid commenters flood content platforms with fake comments to boost visibility for promoters, thereby directing attention to their resources. The interference from large numbers of black/gray industry marketing accounts significantly degrades the quality of content platforms and their direction, posing substantial reputational risks. The following section illustrates the intelligence mining process for comment manipulation groups using actual cases.

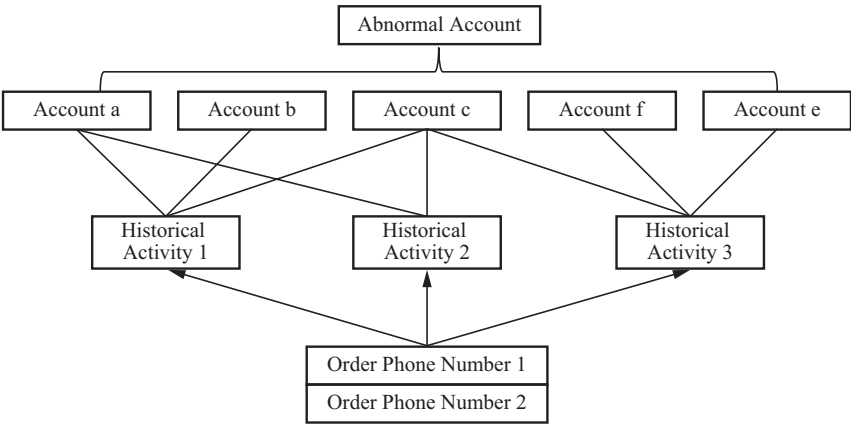


FIGURE 12.5 Historical activity correlations among accounts of “freeloading” fraud syndicates.

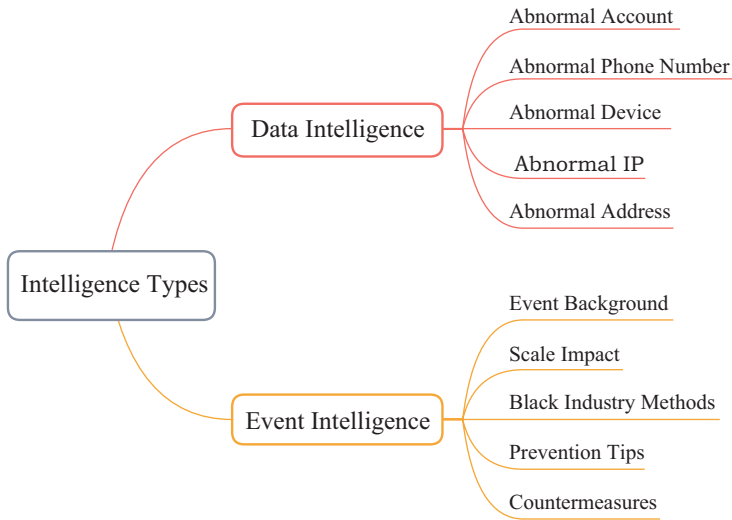
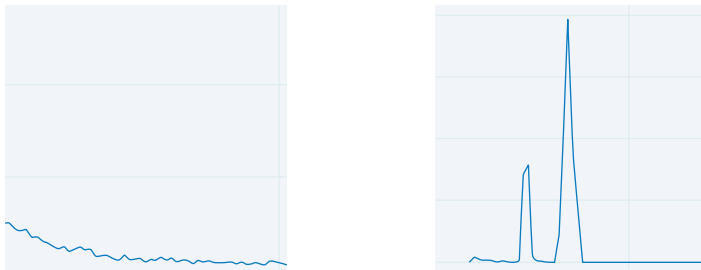


FIGURE 12.6 Types of intelligence generated through intelligence mining.



(a) the trend of normal Comment heat

(b) the trend of heat when abnormal data is monitored

FIGURE 12.7 Trend of heat metrics for normal and abnormal comments.

12.1.2.1 Identifying Products or Articles with Abnormal Comments

Platform operations monitoring detected abnormal data patterns for a content provider: a peak in comments occurring immediately after content updates, followed by a period of stability, then a sudden surge of comments within a short time window. The trend changes in heat metrics for normal and abnormal comments are shown in Figure 12.7. Figure 12.7a displays the heat trend for normal comments, while Figure 12.7b shows the heat trend when abnormal data was detected.

12.1.2.2 Accounts Spreading Content Further

Through monitoring, we identified content generating abnormal comments. After text analysis, we pinpointed accounts suspected of mindlessly repeating identical scripts. Analysis of these commenting accounts revealed concentrated commenting times and anomalies in corresponding IP addresses, device information, and geographic data. This indicates bot-generated comments from black/gray industry bot farms. The relationship

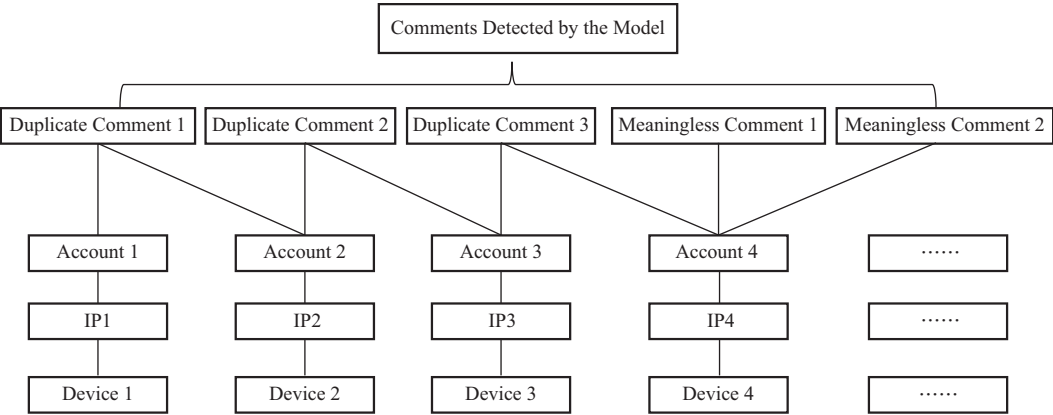


FIGURE 12.8 Account–IP–device relationship for abnormal comments.

between abnormal comments and their associated accounts, IP addresses, and devices is illustrated in Figure 12.8.

12.1.2.3 Further Analysis through Suspicious Account Propagation

For content platforms with social attributes, analysis of account clustering reveals that suspicious accounts interact with certain other accounts, exhibiting hierarchical interactions and IP clustering patterns, along with characteristics indicative of organized groups. Analysis of these spreading accounts reveals distinct organizations from the black/gray industry engaged in volume boosting. By examining interactions between these organizations and external entities, we uncover their division of labor and simultaneously map the volume-boosting supply chain. The comment-boosting supply chain is illustrated in Figure 12.9.

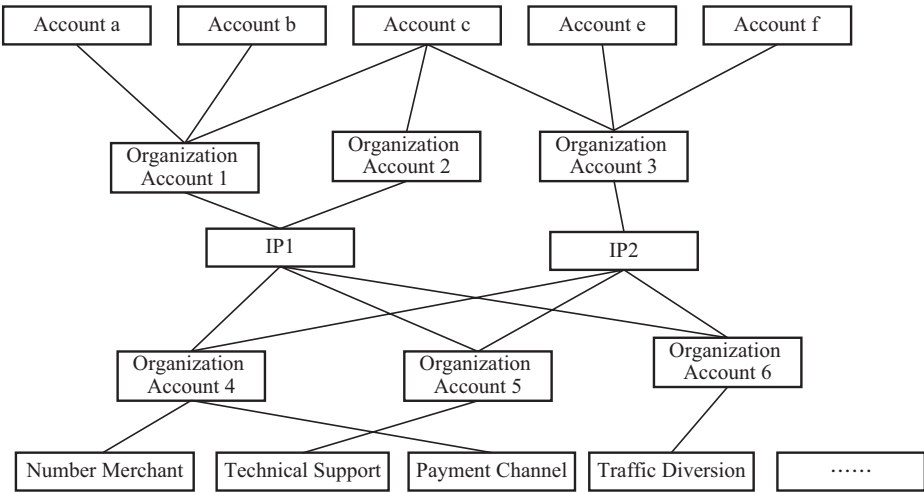


FIGURE 12.9 Comment manipulation industry chain.

12.1.2.4 Methodology for Analyzing Operations of the Black/Gray Industry

Regarding intelligence on comment-boosting activities, from a traffic analysis perspective, the focus ultimately returns to IP addresses, devices, accounts, and behaviors. Using a general methodology for gang intelligence mining, preliminary intelligence monitoring can be conducted first, followed by enhancing perception through partial comment content during the process, and finally optimizing countermeasures.

12.2 INTELLIGENCE MINING OF BLACK/GRAY INDUSTRY BEHAVIOR PATTERNS

Section 12.1 introduced intelligence mining for black/gray industry groups, primarily focusing on the pattern of aggregation-to-diffusion-to-re-aggregation. This approach aims to detect large-scale traffic fraud earlier across broader dimensions. As black/gray industry operations adopt high-cost countermeasures—such as limiting device logins to one or two accounts—making device and account clustering less obvious, behavior-based analysis of accounts can then be employed to uncover aggregated black/gray industry activities.

Chapter 7 detailed the principles of behavioral sequence models. This section presents case studies demonstrating the application of behavioral sequence models in intelligence mining.

12.2.1 Short Video Platform Account Farming

The purpose of black/gray industry account farming is to artificially inflate followers, likes, and traffic for one's own or others' accounts through improper means. This exploits short video platforms' account weighting rules to boost target account rankings or to mass-produce marketing accounts for resale. Ultimately, it undermines the health of short video ecosystems and content quality, severely degrading user experience.

A black/gray industry bulk account farming tool batch-generates fake devices, switches accounts, and mimics normal user behavior by performing clicks, shares, and saves. While bulk farming evades detection based on IP, device, or account clustering attributes, it cannot escape detection based on farming behavior attributes—specifically, the sequence of farming operations. Therefore, the following section employs a behavioral sequence model to identify accounts exhibiting farming activities.

Within short-video platform systems, tracking points enable reporting of various user operation IDs, including login, profile updates, content sharing, content posting, and interactions such as likes, comments, and follows. By generating time series, distinct differences in temporal patterns are revealed. While normal user actions exhibit significant randomness, bulk operations by members of the black/gray industry often show similar behavioral patterns across accounts over time. Comparing these time series clearly highlights discrepancies. Figure 12.10 illustrates the behavioral sequence contrast between normal user accounts and those from the black/gray industry.

Furthermore, as countermeasures against the black/gray industry evolve, operators employ multiple distinct behavioral sequences across different accounts to mimic genuine user behavior. This renders the previously identifiable patterns in account cultivation time series less pronounced. However, the bulk account farming objectives of the black/gray

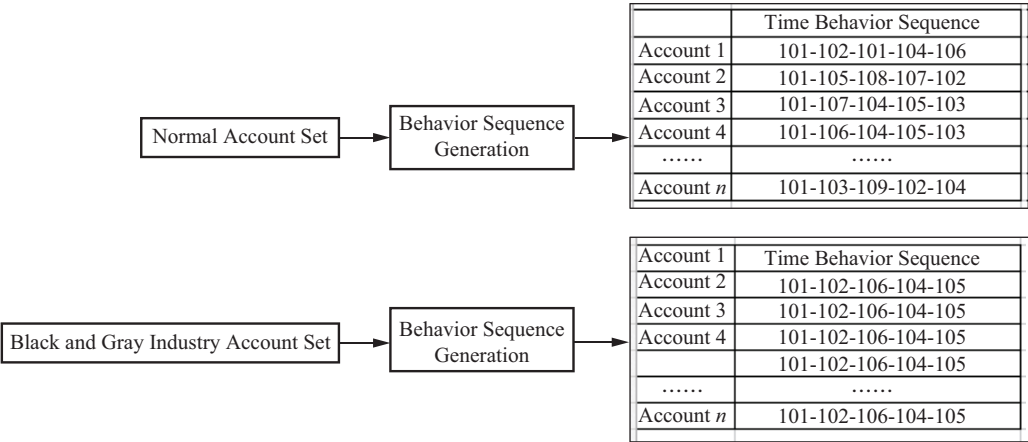


FIGURE 12.10 Comparison of behavioral sequences between normal user accounts and black/gray industry accounts.

industry remain consistent—for example, achieving M1 likes on N posts and M2 shares on N posts. Therefore, behavioral sequences can be further analyzed by transforming from the time domain to the frequency domain. Detailed procedures are outlined in Chapter 7 of this book. The following section uses the ad click fraud case to illustrate how frequency-based behavioral models uncover new fraud intelligence.

12.2.2 Ad Click Fraud

Ad click fraud involves using improper technical means to generate massive amounts of clicks, views, and interactions on advertisements. These actions are typically fraudulent and do not reflect the genuine intent of real users. An ad impression leading to a user click signifies successful matching through the pre-defined targeting process. Upon reaching the ad page, users may exhibit various behaviors. Only core interactions—such as clicking avatars, nicknames, images, or performing actions like like/comment—constitute genuine engagements and are counted toward the cost-per-click.

A platform detected that click spending for a particular ad exceeded expectations, and the spending timeline deviated from the anticipated window. Data analysis revealed that the overall click operation distribution diverged from normal patterns. Simultaneously, tracking the exposed user set via heatmaps showed scattered distributions in user clicks, views, and dwell times on the ad. Based on the ad click anomaly monitoring, the identified account set is shown in Figure 12.11. This indicates a preliminary suspicion of ad click fraud. Further statistical analysis can then pinpoint a group of suspicious ad-clicking accounts.

The differences in click behavior between normal and abnormal users are illustrated in Figure 12.12. Analysis of the identified suspicious accounts reveals the following five anomalous and systematic patterns across multiple dimensions:

- Their ad page click coordinates are highly concentrated.
- Clicks occur primarily during non-standard time periods.

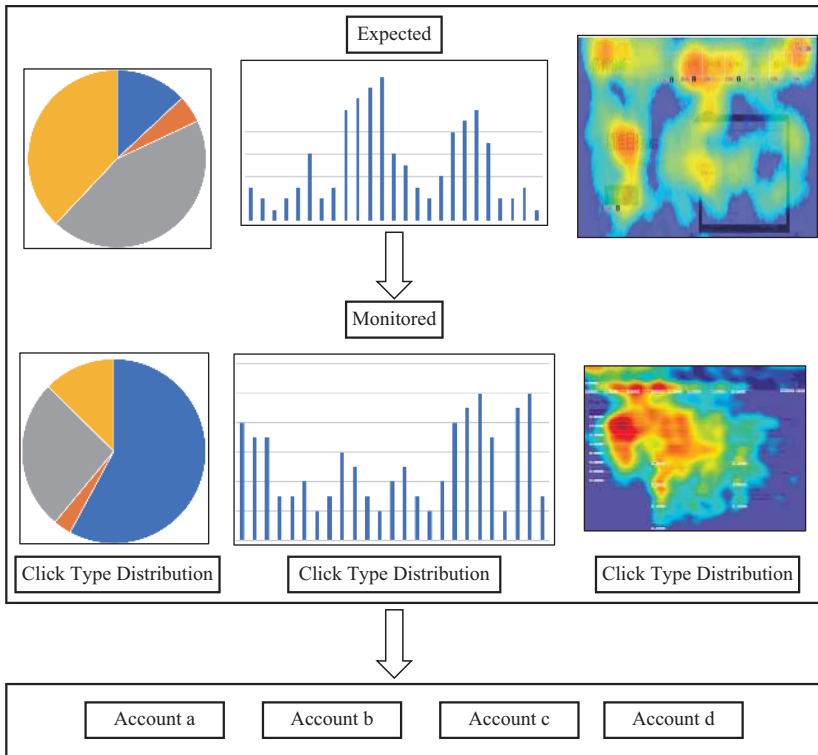


FIGURE 12.11 Account set identified based on ad click anomaly monitoring.

- Most accounts interact with relatively few page elements, and their click speeds are very similar.
- Their click sequences deviate from typical user behavior.
- The frequency of clicks is excessively high.

By further analyzing other dimensions of these accounts—such as whether IP addresses and devices are clustered or appear suspicious—the deductive process described above can be validated. This ultimately leads to the creation of rules suitable for monitoring and deployment.

12.3 BLACK/GRAY INDUSTRY PRICE INTELLIGENCE MINING

Section 12.2 primarily focuses on mining anomalous traffic from criminal groups and individuals based on actual data dimensions of the black/gray industry, with the analysis results serving as intelligence outputs. The greatest disadvantage of intelligence mining for criminal behavior patterns is the lack of more objective data for evaluation, such as feedback on enforcement effectiveness from external public sentiment, trends in account trading prices, and the latest cheating tools used by the black/gray industry. Intelligence from these dimensions provides more persuasive evidence for evaluating enforcement effectiveness and offers more accurate insights into the latest cheating trends. Therefore,

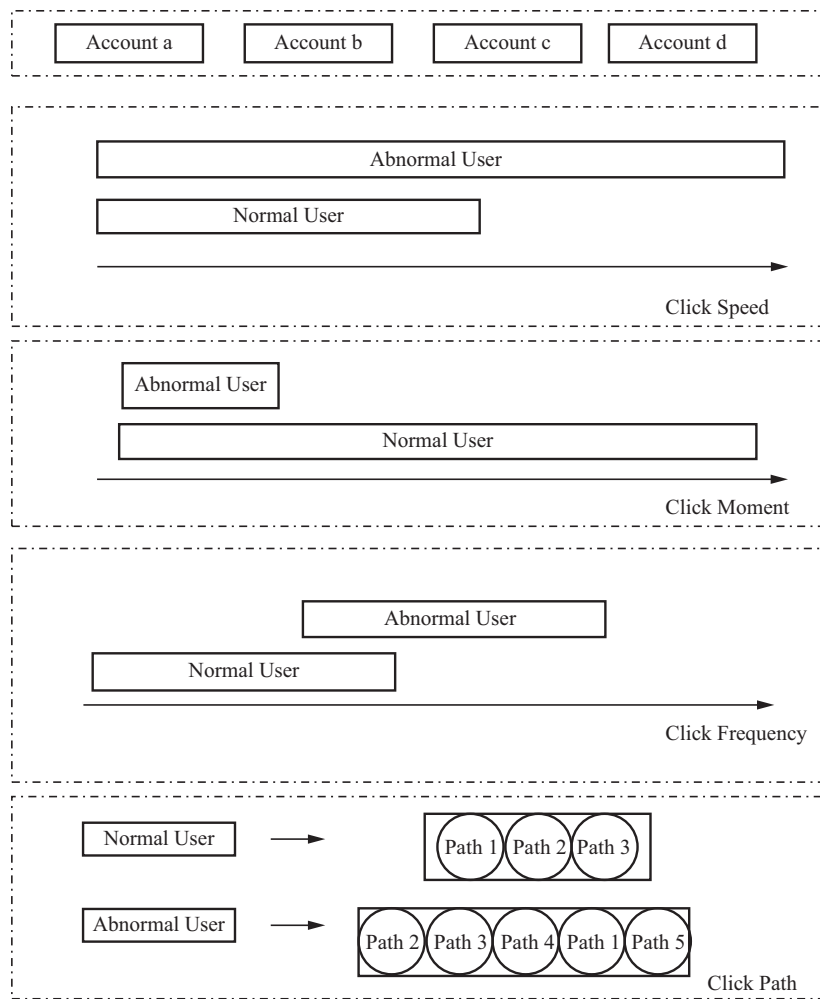


FIGURE 12.12 Differences in click patterns between normal and abnormal users.

monitoring public sentiment, account trading prices, and tool pricing within the black/gray industry supply chain is equally critical. The following example illustrates the value of such intelligence using account pricing as a case study.

In the traffic business scenario of a certain video platform’s paid membership, targeted monitoring of external web resources can yield intelligence on account trading and rental/sale prices.

By continuously monitoring price information for rented accounts in the black/gray industry, compiling and summarizing account and pricing data, and generating intelligence on price fluctuations, the pricing trends for rented accounts in these markets are illustrated in Figure 12.13. External feedback and the upward trend in rental prices clearly indicate that platforms have recently tightened their restriction rules and crackdown strategies, demonstrating significant effectiveness. If rental prices remained unchanged or declined, it would indirectly suggest that the platform’s restrictions and crackdown



FIGURE 12.13 Price trends for account rentals in the black/gray industry.

strategies may not be effective. Combining external data feedback with comprehensive assessments of proprietary mining data yields more persuasive results.

12.4 SUMMARY

This chapter primarily explores the mining and application of intelligence in anomalous traffic detection scenarios. It begins by illustrating the process of mining intelligence on black/gray industry groups through case studies of “scraping for benefits” and paid comment flooding. It then explains the process of mining behavioral pattern intelligence through short video platform account farming and ad click fraud. Finally, it demonstrates more objective and authentic intelligence data by monitoring cases of black/gray industry sentiment and pricing systems. Beyond these intelligence categories, additional dimensions can be mined and applied based on specific business challenges and adversarial scenarios. For the entire traffic fraud defense system, the intelligence system serves not only as a critical tool for detecting security incidents and uncovering leads but also as a vital metric for evaluating the effectiveness of countermeasures.