Security and Communication Networks

Communication Security in Socialnet-Oriented Cyber Spaces

Lead Guest Editor: Ilsun You

Guest Editors: Karl Andersson, Zheli Liu, and Hao Peng

Communication Security in Socialnet-Oriented Cyber Spaces

Security and Communication Networks

Communication Security in Socialnet-Oriented Cyber Spaces

Lead Guest Editor: Ilsun You
Guest Editors: Karl Andersson, Zheli Liu, and Hao Peng

Chief Editor

Associate Editors

Academic Editors

Contents

Contents

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6669429, 19 pages
https://doi.org/10.1155/2021/6669429

Research Article

Design and Implementation of Continuous Authentication Mechanism Based on Multimodal Fusion Mechanism

Jianfeng Guan,[1,2] **Xuetao Li**,[1,2] **and Ying Zhang**[1]

[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Jianfeng Guan; jfguan@bupt.edu.cn

Received 16 October 2020; Revised 8 July 2021; Accepted 3 September 2021; Published 15 October 2021

Academic Editor: Karl Andersson

Most of the current authentication mechanisms adopt the "one-time authentication," which authenticate users for initial access. Once users have been authenticated, they can access network services without further verifications. In this case, after an illegal user completes authentication through identity forgery or a malicious user completes authentication by hijacking a legitimate user, his or her behaviour will become uncontrollable and may result in unknown risks to the network. These kinds of insider attacks have been increasingly threatening lots of organizations, and have boosted the emergence of zero trust architecture. In this paper, we propose a Multimodal Fusion-based Continuous Authentication (MFCA) scheme, which collects multidimensional behaviour characteristics during the online process, verifies their identities continuously, and locks out the users once abnormal behaviours are detected to protect data privacy and prevent the risk of potential attack. More specifically, MFCA integrates the behaviours of keystroke, mouse movement, and application usage and presents a multimodal fusion mechanism and trust model to effectively figure out user behaviours. To evaluate the performance of the MFCA, we designed and implemented the MFCA system and the experimental results show that the MFCA can detect illegal users in quick time with high accuracy.

1. Introduction

With the vigorous development of 5G, IoT (Internet of Things), and AI (Artificial Intelligence), the Internet has penetrated into various traditional industries, which brings in greater data privacy disclosure and more serious information security risks due to the endogenous security issues of the Internet. As the first line of network defense, authentication mechanism becomes a crucial way to ensure information security [1]. The current authentication schemes can be classified into four kinds: (1) authentications based on passwords or PINs (Personal Identification Numbers), (2) portable smart card or token-based authentications, (3) biometric-based authentications, such as face, fingerprint, and iris recognition, which are also called hard biometric-based authentications, (4) behaviour-based authentications such as gait and keystroke, which are called soft biometrics [2, 3]. More specifically, the hard and soft biometric-based authentications overcome the problems that password authentication is long and hard to remember and the problems related to smart card, which is easy to be stolen. On the contrary, biometric-based authentications do not require the authentication entity to be carried along at all times, which is inconvenient and also easy to be lost. Biometric authentications are based on human physiological behaviour characteristics, and have the advantages of natural nonreplication, which greatly improves user experience and reduces the risk of privacy disclosure [4]. However, physiological feature recognition generally relies on specific feature recognition devices such as face recognition device and fingerprint collector, which depend on expensive equipment and even poses the risk of forgery when they lack effective supervision. Besides, due to the limitations of user devices, computation and storage of authentication procedure are generally offloaded to edge or remote cloud, which may increase the attack surface and security risks.

On the other hand, the first three kinds of authentications belong to one-time authentication from the perspective of the identification mode, which only verifies the users' identity when the devices are unlocked for the first time. Once the users have passed the authentication, which is just like getting the device's pass card, they can use the system resources continuously without receiving the verification again [5]. For example, when a legitimate user temporarily leaves for tea or has a short conversation with others, the device that is not immediately locked may be at risk of being used to steal information by an adversary. These kinds of attacks can be classified as internal attacks, which are difficult to defend. The recent report from Cybersecurity Insiders [6] shows that 68% of organizations feel moderately to extremely vulnerable to insider attacks. To prevent insider attacks, the concept of "Zero Trust" has been proposed, which follows the principle of "Never Trust, Always Verify" [7]. The authentication, especially the continuous authentication, plays an import role in zero trust architecture.

This paper designs and implements a continuous authentication system that continuously monitors the user's operations after the device is unlocked. Once the system finds that user identity is abnormal, the device will be automatically locked to prevent the risk of "one-time authentication" and guarantee the user information security. The main contributions of this paper are as follows:

(1) We propose a multimodal fusion mechanism for multidimensional behaviour characteristics. Considering that the single mode recognition is not enough to effectively depict the user's behaviours, we design a multimodal fusion mechanism based on multidimensional features, and construct a trust model for continuous identity authentication, to improve the authentication accuracy and recognition rate.

(2) We select three users' behaviours to realize the multimodal identification, which include keystroke behaviours, mouse movement, and application usage characteristics. The keystroke and mouse movement are time-sensitive and convenient, and the application usage based on logs is more stable and efficient. More important, all of them do not rely on additional hardware devices, and therefore have the advantages of being low cost and user friendly.

(3) We design a Multimodal Fusion-based Continuous Authentication (MFCA) system based on multimodal fusion mechanism. The MFCA system mainly consists of three parts: First, the multidimensional behaviour models (keystroke model, mouse model, and application model) are obtained by training on multidimensional behaviours data; second, the multidimensional model classification results are fused; third, the multidimensional behaviour models are evaluated based on the trust model algorithm, and the real identity of users is evaluated based on the trust evaluation.

The structure of this paper is organized as follows. In Section 2, the related work has been discussed. Section 3 describes Multimodal Fusion mechanism and recognition models used in the MFCA. The design and procedure of the MFCA system have been discussed in Section 4. The performance of the proposed MFCA system is analysed comprehensively in Section 5. Finally, Section 6 concludes the work of this paper.

2. Related Work

The related researches in terms of authentications based on keystroke, mouse movement, or swipe and application usage are shown in Figure 1.

Keyboard and mouse, as the most commonly used input devices, have their own advantages to depict users' behaviour characteristics. Keyboard dominates text input while mouse is more commonly used in GUI. The identification based on single input device will affect the immediacy and accuracy of user identity verification. Moreover, as the inherent hardware equipment, the keyboard and mouse are transparent to users during identity verification, which can avoid targeted destruction or forgery by identity counterfeiters in advance. When users interact with the operating system using the keyboard and mouse, they will trigger the iterative update procedure of the application state. Different users' preferences for application reflect users' using habits, which are irreplaceable and can be used as an important way for identity verification.

Keystroke dynamics refers to the physiological neural control mechanism of humans, which reflects the unique characteristics by analysing users' habits, patterns, or rhythms through the keystroke such as different time intervals between keystrokes and keystroke strength. As early as the 1980s, research studies [8–10] had proved the utility of keystrokes in terms of identity verification.

Mouse dynamics refers to the track and the click of the mouse during user interaction with the system, and the most commonly used features include mouse keystroke speed, habits, frequency, and direction of the mouse moving distance. Everitt and Mcowan [11] found and proved the feasibility of knowing a user's identity by analysing the user's mouse operation habit and behaviour characteristics. With the development of computer GUI, the mouse has superseded the keyboard and become the dominant I/O device [12].

User application usage refers to the characteristic of terminal device in terms of resources scheduling, consuming, and even interacting with other equipment, which can be obtained through the system interface or process information since they are independent of special hardware. More recently, the behaviour-based continuous authentication technologies have been widely active in many fields. For example, user identity recognition can be based on smartphone applications [13], the information from sensors such as gyroscope and magnetometer [14], the users' arm movement records on smart watch [15], and gait recognition

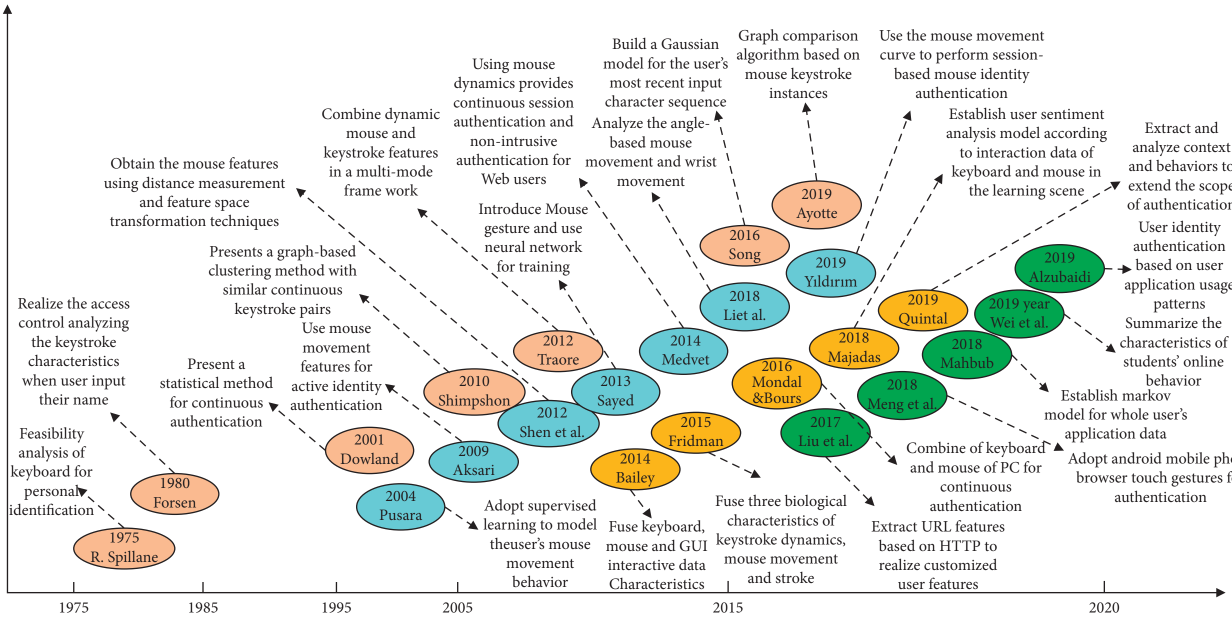

Feasibility analysis of keyboard for personal identification
1975 R. Spillane
Realize the access control analyzing the keystroke characteristics when user input their name
1980 Forsen
Present a statistical method for continuous authentication
2001 Dowland
Use mouse movement features for active identity authentication
2009 Aksari
2004 Pusara
Adopt supervised learning to model theuser's mouse movement behavior
Presents a graph-based clustering method with similar continuous keystroke pairs
2010 Shimpshon
Obtain the mouse features using distance measurement and feature space transformation techniques
2012 Shen et al.
Combine dynamic mouse and keystroke features in a multi-mode frame work
2012 Traore
Introduce Mouse gesture and use neural network for training
2013 Sayed
Using mouse dynamics provides continuous session authentication and non-intrusive authentication for Web users
2014 Medvet
2014 Bailey
Fuse keyboard, mouse and GUI interactive data Characteristics
Analyze the angle-based mouse movement and wrist movement
2018 Liet al.
Build a Gaussian model for the user's most recent input character sequence
2016 Song
2015 Fridman
Fuse three biological characteristics of keystroke dynamics, mouse movement and stroke
2016 Mondal &Bours
Combine of keyboard and mouse of PC for continuous authentication
Graph comparison algorithm based on mouse keystroke instances
2019 Ayotte
Use the mouse movement curve to perform session-based mouse identity authentication
2019 Yıldırım
2017 Liu et al.
Extract URL features based on HTTP to realize customized user features
2018 Majadas
Establish user sentiment analysis model according to interaction data of keyboard and mouse in the learning scene
2019 Quintal
Extract and analyze context and behaviors to extend the scope of authentication
2018 Meng et al.
Adopt android mobile phone browser touch gestures for authentication
2018 Mahbub
Establish markov model for whole user's application data
2019 year Wei et al.
Summarize the characteristics of students' online behavior
2019 Alzubaidi
User identity authentication based on user application usage patterns
1975
1985
1995
2005
2015
2020

Figure 1: The related research of keystroke, mouse/swipe, and App usage. The different shapes represent the different authentication methods.

based on wristband [16]. While in traditional PC, the most commonly used biometric authentication is still based on keystroke and mouse [17].

2.1. Keystroke-Based Authentication Schemes. As early as 1975, Spillane [8] discussed the feasibility of keystroke in user identification, and this suggestion was also verified by Forsen and Gaines in 1977 [9] and 1980 [10], respectively. Forsen et al. [9] realized access control by analysing the keystroke characteristics of users when they input names that are similar to human signatures, while Gaines et al. [10] recorded the keystroke interval when the typist input the specified text, and analysed the time probability of consecutive typing characters, and verified the uniqueness of individual keystroke characteristics. This is the starting point of keystroke recognition, and belongs to static authentication technology based on fixed text. Fixed text refers to the predefined text or phrases to register a user, and it requires the user to type exactly the same text to perform identity verification with the objective of reducing uncertainty by controlling variables and observing the performance of a single keystroke feature in identification. This kind of experiment is generally applicable to scientific researches [18, 19], which has great limitations and is currently only applicable to the identity verification of fixed user names and passwords. Therefore, it is also called static authentication.

On the contrary, in continuous authentication scenarios, users are free to type texts that are not limited by the predefined contents. Free-text-based keystroke recognition is more difficult than fixed one in terms of data preprocessing, feature selection, and keystroke authentication [20]. Dowland et al. [21] proposed a statistical method for continuous certification in 2001, whose accuracy was less than 60%. However, after nearly ten years of development, Shimpshon et al. [22] proposed a clustering method based on graph in 2010 which added a similar continuous keystroke to form a fixed length of the session, and the experimental results in 21 real users and 165 counterfeiters showed that it has a False Accept Rate (FAR) of 3.47% just by using 250 keystrokes. More specifically, Rybnik et al. [23] explored keystrokes in different lengths of nonfixed text in 2013, providing a reliable basis for the authentication of free text. After that, Song et al. [24] constructed a Gaussian model for the user's recent input characters sequence based on the Gaussian probability density function in 2016, which shortened the authentication cycle and reached FAR of 5.3% under 30 characters. Huang et al. [25] updated the keystroke samples using the sliding window method and achieved FAR of 1% and False Reject Rate (FRR) of 11.5% in a 1-minute sliding window in 2017. Furthermore, they evaluated the ability of the proposed algorithm to resist short quick insider attacks and detected insider attacks that lasted 2.5 minutes or longer with a probability of 98.4%. More recently, Ayotte et al. [26] proposed an instance-based graph comparison algorithm, which achieved an EER (Equal Error Rate) of 7.9%, 5.7%, 3.4%, and 2.7%, respectively, under the samples of 50, 100, 200, and 500 keystrokes, realizing faster and more accurate free-text keystroke identification.

2.2. Mouse-Movements-Based Authentication Schemes. Mouse dynamics has also received attention in terms of authentication. In 2003, Everitt and Mcowan [11] proved the feasibility of mouse behaviour characteristics in user identity authentication for the first time, which set a solid foundation for the subsequent extensive researches in the academic community. In 2004, Pusara and Bradley [27] used supervised learning to model the mouse movement behaviour of 11 users and obtained FAR of 0.43% and FRR of 1.75%. However, due to the small number of user samples and single mouse features, the authors also pointed out that the analysis of that study was not enough to achieve independent user identity authentication. In 2009, Aksari and Artuner [28] used mouse movement characteristics for active identity authentication, and obtained FAR of 5.9% and FRR of 5.9% by analysing the mouse trajectory when the user clicked 10 squares in a row. In 2013, Sayed et al. [29] introduced mouse gestures into the user registration system and performed training through the neural network to realize identity verification when the user logins, and finally reached FAR of 5.26% and FRR of 4.59% within 26.9 s in a dataset of 39 users. The above researches are also called static mouse authentication, which mainly explores the diversity of mouse features and the wide application field of mouse recognition through specifying user behaviour or limiting mouse operation scope and trajectory.

On this basis, mouse movement-based continuous authentication schemes have attracted more and more attention. In 2012, Chao Shen et al. [30] evaluated 5550 mouse operation samples of 37 users, and obtained mouse features based on distance measurement and feature space transformation technology, which reached FRR of 8.74% and FAR of 7.69% within 11.8 s. Besides, they established the first public mouse-behaviour dataset, and their research results revealed the potential of mouse dynamics in user authentication. In the same year [31], the pattern-based growth method was used to mine frequent mouse behaviour fragments, and obtained more stable mouse features and reached FAR of 0.37% and FRR of 1.12%. In 2014, Medvet et al. [32] used mouse dynamics to provide continuous session authentication and nonintrusive authentication for web users, and achieved the accuracy of 97% for 24 users. Their work extended the potential scope of mouse dynamics as a continuous authentication tool to web applications hosted in the cloud rather than just in local devices. In 2018, Li et al. [33] used the random forest and sequential sampling analysis to analyse the angle-based mouse movement and wrist movement, and reached FAR of 1.46% on the dataset of 26 users, and the verification time could be determined within 9–12 mouse clicks. Their approach is more effective in timely authentication compared with methods based on the mouse geometry and locomotion features. In 2019, Yildirim and Anarim [34] verified on Balabit dataset [35] that mouse movement curves alone and session-based mouse identity could be used, achieving Area Under Curve (AUC) of 93% and EER of 13%.

2.3. Application-Usage-Based Authentication Schemes. Different from keyboard and mouse-based authentication schemes, application-usage-based authentication is not a biometric technology but a behavioural analysis based on user activity records with the objectives to mine user activity records, extract user multi-attribute behaviour characteristics, and then build user behaviour model to represent user identity and complete continuous authentication. Lots of current research efforts have proved the uniqueness of user behaviour, and thus derived a lot of user behaviour analytical methods based on big data.

In 2017, Liu [36] extracted URL characteristics and identified user's consumption level by taking users' surfing time as the sample. In 2018, Mahbub et al. [13] built a Markov model based on the complete application data of users, carried out continuous authentication by evaluating the changes of hidden Markov model (HMMs), and finally realized the capture of abnormal users within 2.5 minutes on the experimental dataset. They solved the active authentication problem by using application usage formulaically and systematically. Furthermore, they suggested that unknown application and unforeseen events had more important impacts on the authentication performance than the most common ones. In 2018, Meng et al. [37] conducted user authentication based on the touch gestures of Android mobile phone browser, achieving an average EER of 2.4% among 48 participants, and their system can reduce the touch behavioural deviation than others. In 2019, Wei [38] analysed the DNS logs of campus network by categorizing the domain names to obtain users' online behaviour habits and access preferences, and summarized the characteristics of students' online behaviours. In terms of user analysis based on application records, it can be divided into single application based, top *n* applications based, and all applications based behaviour identification by considering potential unique user behaviour pattern when using applications.

Besides, Alzubaidi et al. [39] presented an active authentication based on the smartphone usage data under different machine learning models, and achieved a lower EER of 8.2% for authenticating users within short periods of time with a small number of features on the MIT dataset [40]. Their scheme was effective in reducing the classification error rate compared with other authentication methods. For mobile devices, they are easy to deploy authentication based on the App using record, phone usage record, and even web browser history. However, due to the different operating systems, it is difficult to achieve the intersystem authentication.

2.4. Multimodal-Fusion-Based Authentication Schemes. With the development of various authentication technologies, some researchers try to combine different authentication technologies to increase the accuracy and timeliness. Although no single authentication technology is perfect, it is very difficult to fool multiple authentication methods at the same time. Therefore, multimodal fusion authentication can overcome the problems of partial feature loss. The recent work from Modak and Jha [41] summarized the multi-biometric fusion strategy and its different applications in terms of multi-modal, multi-algorithm, multi-sample, multi-sensor, and multi-instance, and proved the performance upgrade of combining two or more individual biometric traits.

As early as in 2012, Traore et al. [42] proposed an online authentication system under web environment, which combined dynamic mouse and keystroke features in a multimodal framework to conduct real-time monitoring of 24 user operations, and the final system EER was 8.21%. However, their results had a low Average Number of Genuine Actions (ANGA) value which made the system not practical for real users. In 2014, Bailey et al. [43] proposed a user authentication system based on multimodal behavioural biometrics by fusing user data from keyboard, mouse, and GUI interactions, and adopted ensemble classification method to get FAR of 2.1% and FRR of 2.24% over the dataset with 31 users, which supports the idea of multimodal fusion to gain better consequence. In 2015, Fridman et al. [44] presented a multimodal fusion for continuous authentication by collecting the behavioural biometrics of keystroke dynamics, mouse movement, and a high-level modality of stylometry, and developed a sensor for each modality and organized these sensors as parallel binary decision fusion architecture. Their experimental results based on database of 67 users who work individually for a week show that FRR and FAR are less than 1% within 30 s. In 2016, Mondal and Bours [45] proposed a continuous identity authentication for PC users by combining keystroke and mouse dynamics, and the recognition rate reached 62.6% and 58.9% in closed and open environments, respectively. The average operation times were 471 and 333, respectively. Besides, they first introduced the issue of Continuous Identification (CI) and discussed the concept of Continuous Authentication and Identification that provided the combination of security and forensics. In the same year, Beserra et al. [46] applied the dynamic identity recognition application by combining keyboard and mouse for the first time in online games, and carried out real-time identification of player operations to realize anti-cheating function. In 2018, Sergio et al. [47] established a user emotion model based on the interaction data of the keyboard and mouse in the learning scenario, so as to predict the affective state of the learner. In 2019, Quintal et al. [48] analysed the mobile user continuous authentications in IoT, and classified these authentication factors into event capture types such as password, fingerprint, applications start and end, network connection and disconnection, continuous sequence of events, such as gestures, and derived behavioural features, such as application choice, and demonstrated that all factors are correlated with the actual user identity. Currently, lots of multimodal continuous authentications are proposed in smartphone, IoT [49–51].

The key points of multimodal fusion continuous authentication are the association, unified representation, and coordination of multimodal information, and the main issues are: (1) multimodal characteristic expression, that is, how to design single-modal characteristics under the framework of multimodal architecture; (2) how to unify the model of multimodal characteristics. In our preliminary work, we have studied continuous authentication based on users' keystroke and mouse behaviour [19], and developed a prototype system. Among them, a static authentication algorithm based on convolutional neural network is proposed for user keystroke behaviour. The average accuracy on CMU dataset is 96.8%, the average FAR is 0.04%, and the average FRR is 6.5%. At the same time, a continuous authentication algorithm based on the weighted reward and punishment mechanism was proposed. When the effective double key pairs of each user are 100, the EER is 8.5% and the AUC is 93.94%.

For this purpose, this paper designs a multimodal fusion continuous authentication mechanism based on users' multidimensional behaviour characteristics in terms of keystroke, mouse, and application usage to effectively prevent the illegal user identity phishing, avoid data privacy, improve authentication efficiency, and ensure the safety of user information.

3. Multimodal-Fusion-Based Continuous Authentication

The MFCA system consists of multimodal fusion mechanism, trust model, and multidimensional behaviour recognition models. In this section, we will introduce the multimodal fusion mechanism and three recognition models that are used in the MFCA.

3.1. Multimodal Fusion Mechanism.

The multidimensional behaviours of network users mainly include keystrokes, mouse, screen swipe, and application usage. This paper designs the multimodal fusion mechanism to collect user behaviour data, combines these multidimensional features effectively, fuses the multiple classifier to avoid the limitation of the single classification and improve the classification accuracy and generalized capability, and finally realizes the continuous authentication.

3.1.1. Multi Classifier Fusion Mechanism.

Considering the diversity, complexity, and fusibility of the features, this paper adopts the Multi-Classifier Fusion (MCF) mechanism to improve the accuracy and generalization capability of the final classification results by integrating the output classification results of base classifiers. At the same time, MCF can simplify classify design, balance classification time and performance, and improve time and space efficiency. The typical structure of the MCF includes cascade combination, parallel combination, and mixed combination. Parallel combination does not have the error accumulation problem of cascade combination. Furthermore, the system can achieve the best performance of real-time classification by designing an appropriate decision process. So, this paper adopts parallel combination to perform parallel processing on the user's multidimensional behaviours including keystroke dynamics, mouse movement, and application usage data.

The results of the MCF algorithm depend on the output type of the base classifier. When the base classifier output is an interval value or probability value, we can adopt the mean value method (simple average or weighted average), maximum-minimum value method, product method, etc. When the output is a predefined class label, we adopt the voting method such as weighted voting, supermajority voting, or relative majority voting.

3.1.2. Trust Model Design. The trust model is the base of the MFCA and its time-variant characteristic is the key to realize the continuous authentication. The basic idea of the trust model is that the degree of credibility of the current operating user depends on the deviation between the user's behaviour characteristics and the expected characteristics of the model over a period of time. The system predefines trust score and trust threshold at the outset, and then increases or decreases the trust score along with their operations. When the user's behaviour characteristics conform to the model, the trust score will increase (no more than maximum score). Otherwise, the trust score will decrease. Once the score falls below the predefined trust threshold, an exception alarm will be triggered.

Figure 2 shows the schematic diagram of the fluctuating trust score. Over a period of continuous operations, the legitimate user behavioural characteristic is the most trusted attribute even though it may not be stable most of the time. The corresponding trust score will be slightly up and down in a certain period of time, but it is always higher than the trust threshold. The legitimate users will almost imperceptibly perceive the authentication system in order to ensure transparency. On the contrary, the abnormal operations of the illegal user will inevitably lead to the continuous decline of the trust score, which will eventually make the trust score lower than the trust threshold and trigger the abnormal alarm. Therefore, without relaxing the timely detection of illegal users, the design of the trust model increases the tolerance of legitimate users' misoperations to improve the accuracy and user-friendliness of the authentication system.

Table 1 shows the related parameters used in the trust model. Each user has an initial trust score of T_0. The model verifies the current user's identity status in real time according to the user's behaviour characteristic F_i. When the user's identity is judged to be legitimate, the trust model gives rewards to increase the trust score until the highest threshold $T_{\max}$. Otherwise, it will reduce the trust score until the minimum threshold $T_{\min}$. When the score is lower than the trust threshold T_{alert}, the system alarm will be triggered to lock the device. The increase or decrease of the trust score is limited by the maximum reward score R and maximum punishment score P, and the increase or decrease range depends on the reward and punishment weight W_i of the current characteristic.

According to the above definitions, we can deduce equations (1) and (2), from which the trust score T_i is obtained after the initial authentication.

$$\Delta_T(F_i) = \begin{cases} F_i * W_i * R, & F_i = 1, \\ (F_i - 1) * W_i * R, & F_i = 0, \end{cases} \qquad (1)$$

$$T_i = \min\{\max\{T_{i-1} + \Delta_T(F_i), T_{\min}\}, T_{\max}\}. \qquad (2)$$

The basic component of the trust model is the user's single behaviour characteristic, and the reward and punishment range of the trust score depend on the reward and punishment weight of the given characteristic. The specific weight is introduced for that the system involves three types of classification models.

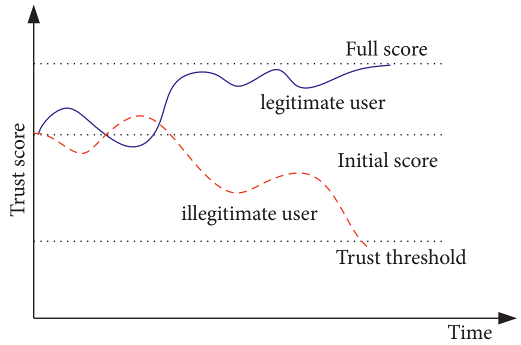


FIGURE 2: Schematic diagram of trust score fluctuation curve.

TABLE 1: The parameters of trust model.

Parameters	Meaning	Value

F_i	The classification result of i^{th} feature	{1, 0}
		Legal = 1,
		illegal = 0

T_i	The trust score of i^{th} authentication	$[T_{\min}, T_{\max}]$
$T_{\min}$	The minimum trust score	$T_{\min}$
$T_{\max}$	The maximum trust score	$T_{\max}$
T_{alert}	The alert threshold	$[T_{\min}, T_{\max}]$
W_i	The punishment weight of feature i	[0–1.00]
R	The maximum punishment score	$[0, T_{\max}]$
P	The minimum punishment score	$[0, T_{\max}]$

Different behavioural operations will generate different characteristics, but user behaviours have a certain pattern. Therefore, the characteristic with high frequency will be considered more stable and identifiable. In contrast, the characteristics with low frequency generally have lower credibility in the trust model. Take mouse keystroke events as an example; mouse keystroke events are divided into left click, left double click, right click, and right double click. When the occurrence probability of left-click events is much greater than that of right-click events, the stability of left-click behaviour is stronger, and the reward and punishment weight obtained are also higher. For example, left-click occurs 67 times, double-click occurs 20 times, right-click occurs 10 times, and double right-click occurs 3 times. When the user is judged as a legitimate user in the left-click feature, the trust score should be rewarded with 67 $R/100$. Otherwise, when the user is judged as an illegal user, the trust score will be punished with 67 $P/100$. Therefore, the trust score value is mainly affected by two major factors in the weight design: the weight ratio of characteristic model in model fusion and the frequency ratio of the feature in the feature set.

3.2. Keystroke Recognition Model

3.2.1. Keystroke Dataset Capture Module. In this section, we give the keystroke capture procedure of Windows as an example. The user interacts with the computer through the keyboard to finish the input, so the keystroke data capture range is global events. Therefore, we adopt the keyboard hook to collect the keystroke data and encapsulate hook into the Dynamic Link Library (DLL) to ensure the automatic loading and real-time collection of keystroke data. The implementation of keyboard hook is divided into three parts: the installation of keyboard hook, the monitoring and processing of keyboard message, and the uninstalling of keyboard hook. Figure 3 shows the procedure of keystroke data capture.

First, the keystroke capture module adds the keyboard hook to the list and binds the keystroke event to the keyboard hook via the *SetWindowsHookEx* () function that mainly consists of four parameters. The first parameter *idHook* represents the installed hook type which has two kinds. This module selects a global keyboard hook called *WH_KEYBOARD_LL*, which contains lots of keyboard information such as virtual keyboard key value *vKCode*, keystroke state *WM_KEYUP* and *WM_KEYDOWN*, and so on. The second parameter LPFN points to the hook subroutine for further processing of the hooked message, which is also called the call back function. In this module, we rename this function as *KeyboardProc*. The third parameter *hMod* is the current instance handle which is also known as DLL module handle. The fourth parameter *dwThreadId* is the thread identifier associated with the keyboard subroutine.

Second, the *KeyboardProc* function is used to monitor keyboard messages, and Table 2 shows the related field information to be collected. When a user clicks a key, the keyboard hook captures the event and begins to record the keystroke value, keystroke timestamp, keystroke event type, and so on.

As for the conversion of key values and codes, the commonly used ASCII codes distinguish the key values of upper and lower case letters "*a-z*" from "A-Z", with 65–90 representing uppercase letters and 97–122 representing lowercase letters. *VkCode*, on the other hand, is treated as the same keyboard key without distinction, and only records the A-Z key value with 65–90. Therefore, when collecting records, the system needs to further determine whether the Shift key is being pressed through *GetAsyncKeyState* (), and obtain the state of CapsLock key through *GetKeyState* (). When either of them is pressed, the letter key is defined as uppercase state, and vice versa.

After that, the specific type of keystroke event is obtained through *wParam*. The system aims to intercept *WM_KEYUP* (key press down) and *WM_KEYDOWN* (key release), and records the keystroke timestamp through *GetLocalTime* function. In this case, the time can be accurate to milliseconds. Finally, the *CallNextHookEx* () function is used to complete the delivery to the next hook in the list, and the keyboard hook is destroyed once the data collection is completed.

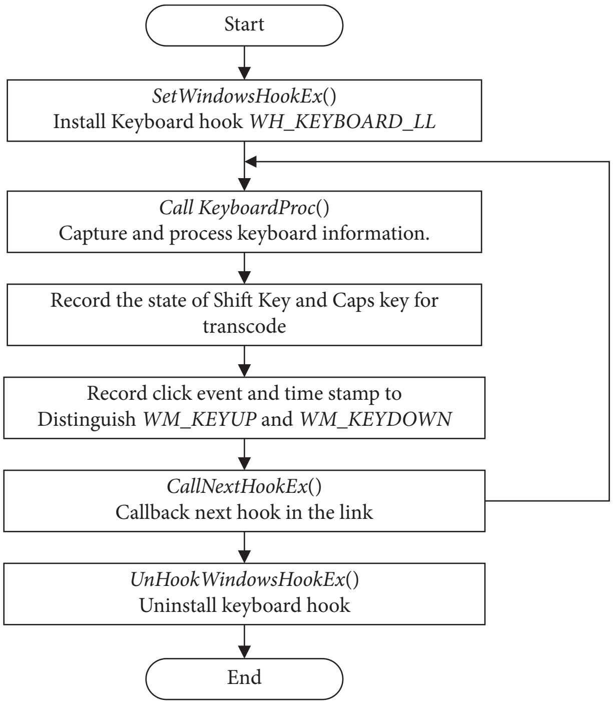

Start
SetWindowsHookEx()
Install Keyboard hook WH_KEYBOARD_LL
Call KeyboardProc()
Capture and process keyboard information.
Record the state of Shift Key and Caps key for transcode
Record click event and time stamp to Distinguish WM_KEYUP and WM_KEYDOWN
CallNextHookEx()
Callback next hook in the link
UnHookWindowsHookEx()
Uninstall keyboard hook
End

Figure 3: The procedure of keystroke capture.

Table 2: Keystroke data information.

Field name	Type	Description
keyCode	Int	Keystroke key code
keyValue	Char	Keystroke key values
keyEvent	Int	Event type
keyStamp	Long	Keystroke timestamp
isShiftOn	Bool	Shift key on/off
isCapsOn	Bool	CapsLock key on/off

3.2.2. Keystroke Data Preprocessing and Feature Selection.

The original keystroke record obtained through data acquisition is a combination of key code, key value, event type, and timestamp, such as 87, W, WM_KEYDOWN, and 59108278, respectively. Due to different event types, the keystroke behaviour of the same key value distinguishes two records of press and release. Therefore, it should be merged and converted into key code, key value, press timestamp, and release timestamp at first, and then deletes the record with the missing value. Finally, the raw data are transformed into feature data.

In free-text environment, user keystroke is affected by the language, profession, and even emotional stress. Therefore, the behaviour habit is random and diverse. On one hand, the keyboard layout is complicated, which includes typical QWERTY keyboard with 87, 104, and 109 keys. On the other hand, the use of the function keys is adventitious, and its characteristics need long-term observation. Therefore, timeliness is insufficient when it is used in continuous authentication. Our system extracts the characteristics of user's inputted characters. 26 character keys will randomly form different character sequences which are affected by language grammar and common words, and the typical character combinations have a wide range of universality. When different users hit the same character, they will show different time characteristics and keystroke frequency. Besides, the length of character sequence determines the order and the magnitude of the combined sequence and the space-time loss complexity of feature processing. Therefore, we select the user double key combination comprehensively, that is, the character sequence of length 2 is used as the feature sample of the keystroke recognition model. To select the most popular double key pairs, we conduct statistical experiments on the statistical frequency of double key combinations, and record the frequency of the top 20 double key combinations among hundreds of thousands of valid character keystrokes [19], as shown in Table 3. Finally, the top 7 double keys are selected as the double key feature samples.

The seven double bond characteristics (“AN,” “NG,” “IN,” “SH,” “EN,” “IA,” and “CH”) are extracted uniformly for three types of time characteristics: Hold time, Down-Down time, and DownUp time. As shown in Figure 4 taking the double keys “WO” as an example, its characteristics are described as follows:

(1) Hold [W]: The duration of key “W” from press to release, likewise Hold [O];

(2) DD [W] [O]: The interval between press “W” (down) key to press the “O” (down);

(3) UD [W] [O]: The interval between bounce “W” key (up) to press “O” (up) key.

3.2.3. Train and Test of the Keystroke Model. For the double key characteristics in the keystroke process, the system adopts the decision tree algorithm for model training, as shown in Algorithm 1. First, Shannon entropy and information gain are selected as the criteria for feature selection of the decision tree. Second, the 7 double keys features are calculated one by one to obtain the current information gain, so as to constantly update the maximum information gain and the best features. After that, the current subtree is created according to the best feature data, and the current best feature is continuously removed to complete the recursive creation of the entire subtree. Finally, after the entire decision tree is built, the decision tree generated by training is returned. In addition, we adopt Pessimistic Error Pruning (PEP), and the penalty factor is set to 0.5 to prevent overfitting.

The adoption of the decision tree is due to the fact that once the training is completed, the distinguishing and classifying of the existing features are very fast in the testing stage. Therefore, in the process of user’s continuous keystroke in the authentication stage, keystroke data within a short period will contain 7 predefined double key characteristics with a high probability. During this time, user identity determination will be quickly completed and authentication results will be calculated through the decision tree model immediately.

3.3. Mouse Movement Recognition Model

TABLE 3: The statistical table of double keys.

No (#)	Double keys	Frequency	No (#)	Double keys	Frequency
1	AN	9619	11	WO	32
2	NG	7580	12	AO	3220
3	IN	7338	13	NA	3015
4	SH	6605	14	EI	2900
5	EN	6049	15	HE	2653
6	IA	5932	16	HS	2622
7	CH	4926	17	XI	2554
8	ZH	4145	18	ON	2466
9	AI	3869	19	HI	2337
10	JI	3798	20	IE	1906

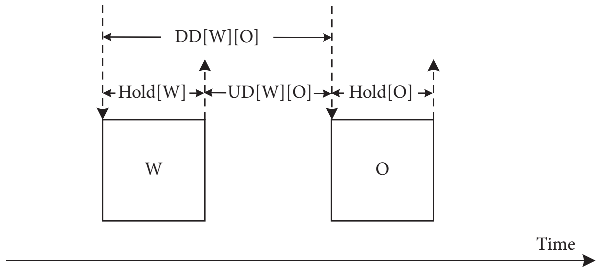


FIGURE 4: The time characteristics of “WO”.

3.3.1. Mouse Movement Data Capture. Similar to keystroke data collection, mouse movement data capture also applies hook technology, which belongs to the global mouse hook WH_MOUSE_LL in the system hook. The overall capture process includes the establishment of mouse hook, the interception and processing of mouse message, and the uninstallation of the mouse hook. The establishment and uninstallation of the mouse hook are similar to that of the keystroke hook. As for the monitoring and processing of mouse messages, it defines the unique mouse data as shown in Table 4, which is different from the keystroke data. When the user manipulates the mouse to trigger the mouse event, the mouse hook captures these messages, triggers the call back function *MouseProc*, and starts to record the mouse event type, mouse cursor coordinates (x, y), and event occurrence timestamp.

3.3.2. Mouse Movement Data Preprocessing. The original captured mouse data format is mouse event type, X coordinate, Y coordinate, and timestamp, which is relatively simple. However, mouse events have natural complexity, which can be mainly divided into four types of events: mouse idle, mouse moves, mouse drags, and mouse clicks. Among them, mouse clicks can be further divided into left click and right click, left double click, and right double click. Besides, the click events can be further divided into press (down) and release (up). Therefore, it is important to preprocess the mouse data, and transform the scattered data records into effective mouse events, and further divide them into mouse features that can be used for identity authentication.

As shown in Figure 5, the mouse data preprocessing procedure is as follows.

Step 1: mouse click events are divided into left mouse click, left mouse double click, right mouse click, and right mouse double click. The mouse hook further

```
    Input:
        The keystroke dataset matrix $X$;
        The keystroke feature vector $F$: = $F^1$, $F^2$, . . ., $F^n$;
    Output:
        The Decision Tree Model, Tree;
        initialize: do preprocess and split, data = process (X, F);
        initialize: init bestInfoGain = 0.0, bestFeature = -1
        calculate Shannon entropy; shang = calculateshang (data)
        for curFeature = 0 to n do
            calculate newEntropy and curInfoGain
            bestInfoGain = max (curInfoGain, bestInfoGain)
            bestFeature = curFeature
        end loop
        for value = 0 to data[bestFeature]. size () do
            Tree[bestFeature][value] = createTree (X, F - bestFeature)
    end loop
    return Tree
```

Algorithm 1: KeyStorke Model's Train [createTree].

Table 4: Mouse movement data capture.

Field	Type	Description
mouseEvent	Int	Mouse event type
mouseX	Int	Cursor x coordinate
mouseY	Int	Cursor Y coordinate
mouseStamp	Long	Mouse event timestamp

divides the left click and right click into left/right press and left/right release events, which are recorded, respectively. Therefore, the mouse click records are summarized and reformatted into the format of mouse left/right click, X coordinates, Y coordinates, press timestamp, and release timestamp.

Step 2: delete the null values. After clicking the event summary, the entire row of records with blank values in all mouse data will be deleted.

Step 3: classify the mouse data because it is difficult to extract effective features from the complicated mouse data. According to the time stamp record and pixel distance, mouse events are limited and the corresponding noncompliant data are eliminated as follows.

(1) The coordinates of the mouse cursor remain unchanged for 1 s, and the mouse is deemed to be stationary

(2) If the mouse cursor moves more than 30 pixels, it will be regarded as mouse movement

(3) If the mouse cursor changes for more than 1 s and the moving distance is greater than 30 pixels, while the left mouse button is not released when pressed, the mouse will be deemed as a drag

Step 4: define sessions to partition mouse behaviour events. The number of mouse behaviour events within a session is called session length X, and the average mouse operation time reaching session length X is called a time slice T. When the time slice is fixed, the more effective the mouse events in a session, the more stable the user behaviour characteristics and the higher the identification accuracy. However, the length of the time slice is proportional to the number of effective mouse events. The longer the time slice, the more effective the mouse events must be. However, excessively long time slice cannot guarantee the timely detection of abnormal users, which violates the original intention of the system design.

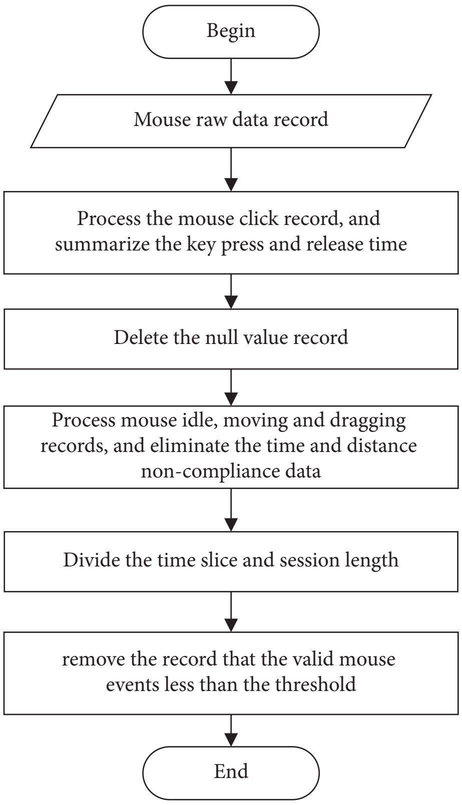

Begin
Mouse raw data record
Process the mouse click record, and summarize the key press and release time
Delete the null value record
Process mouse idle, moving and dragging records, and eliminate the time and distance non-compliance data
Divide the time slice and session length
remove the record that the valid mouse events less than the threshold
End

Figure 5: Flow of mouse data preprocessing.

Step 5: further simplify the mouse record according to the selected time slice $T$and session length X. When the number of valid mouse events in time slice $T$is less than X, this session event is discarded without further feature extraction.

3.3.3. Selection of Mouse Features. In the mouse recognition model, the system extracts the mouse features according to the user's mouse behaviour for verification. The mouse features are complex and diverse, and the user identity can be effectively measured by using the features of time, position, frequency, and mouse trajectory. In order to ensure the timeliness and accuracy of the model in the continuous authentication, the system chooses mouse movement with more obvious characteristics in a short period.

When the mouse cursor moves from the point P_1 (x_1, y_1) to P_2 (x_2, y_2), it shows the following five characteristics during the movement:

(1) The proportion of mouse movement events in 8 different movement directions.

(2) The moving distance (Euclidian distance) of the mouse in 8 directions including average moving distance and extreme moving distance. The calculation of the average moving distance is shown in Equation 3. The calculation of the extreme moving distance is the maximum moving distance in a single time slice T.

$$d = \frac{1}{M} \sum_{0}^{M} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{3}$$

(3) The moving speed of the mouse in 8 directions, including average moving speed and ultimate moving speed. The calculation of the average moving speed is shown in Equation 4. The calculation of the extreme moving speed is the maximum moving speed in a single time slice T.

$$v = \frac{\Delta d}{\Delta t} = \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{t_2 - t_1}. \tag{4}$$

(4) The moving acceleration of the mouse in 8 directions is shown in Equation 5.

$$a = \frac{\Delta v}{\Delta t} = \frac{v_2 - v_1}{(t_2 - t_1)^2}. \tag{5}$$

(5) The proportion of mouse movement events in all mouse operation events.

3.3.4. Training and Testing of the Mouse Model. In the fixed time slice, the distribution of mouse behaviour events does not have regularity, so the mouse movement characteristics extracted by this system also have a small sample size and do not conform to the normal distribution. Therefore, it is difficult to obtain a good recognition effect on the classification model based on statistics or neural network. Many studies have proved that SVM performs well in small sample data and nonlinear high-dimensional mode. After comprehensively considering the number of mouse features and the samples, this paper chooses linear Support Vector Machine (SVM) [52] as the classifier of mouse recognition model, and uses the open source *libsvm* 3.0 [53] to build the classification model. Furthermore, since the gamma parameter in the Gaussian kernel will affect the width of the Gaussian function, the larger the gamma, the easier it is for the SVM to overfit. So our system sets gamma to 0.5.

Our system first uses Principal Component Analysis (PCA) for dimensionality reduction processing of the early collection of multidimensional mouse motion features, and retains the correlation of each feature to avoid the occurrence of dimensional disasters. After feature selection is completed by PCA, the original 45-dimension mouse features are reduced to 16-dimension features.

The mouse recognition model based on *libsvm* is divided into training stage and testing stage. The algorithm description is shown in Algorithm 2. Since the feature dimension reduction is required at both stages, the training and testing of the mouse model are summarized in the same function description and distinguished by option O.

X is the characteristic matrix of the mouse, which is divided into training set and test set according to different O values. L is the mouse label matrix, and the size of the matrix depends on the number of samples n. This system is a binary classification model [54], so the label is defined as (0, 1), where the legal user is 1 and the illegal user is 0. *Op* for *libsvm* training custom parameters, including kernel function and other values, in this system is mainly selected by using the exhaustive method.

In the training stage, the system performs dimensionality reduction on the features of the training set, conducts training according to the *libsvm* options of the feature data set, and finally exports the mouse recognition model MM after the training. In the testing stage, the system predicts according to the existing model MM and test set data, and exports the user identification result, which is legal or illegal, and calculates the classification accuracy ACC.

After that, according to the mouse data and mouse recognition model, the session length X and time slice Twere tested, in which X was 50,100, and 200 valid mouse events. The experimental data set was user mouse operations collected within 48 hours, including about 15,000 effective mouse operation events.

Figure 6 shows the ROC of session length $X = 100$. FAR is negatively correlated with FRR, and when FAR = FRR, its value is ERR. In addition, when X are 50 and 200, the ROC trend is the same as the whole, but the error rate ERR and the average time slice $T$are greatly different. As shown in Table 5, with the increase of the session length X, the ERR is reduced. This is because, the more effective the mouse events in the session cycle, the more stable the mouse features displayed by users. However, at the same time, the longer the session length is, the greater the corresponding average session time T will be, which will lead to the longer user behaviour detection time and therefore greatly affect the system's timely

```
Input:
    The mouse dataset matrix $X$;
    The option of train or test, $O$;
    The mouse label vector $L := L^1, L^2, \ldots, L^n$;
    The libsvm options $op$;
    The number of principal components, $c$;
Output:
    The Mouse Model, $MM$;
    The Predicted Answer, $Ans$;
    The Predicted Accuracy, $Acc$;
initialize: do preprocess, $pca = \mathrm{PCA}\ (c)$;
initialize: $pca.$ fit $(X)$
if $O == 0$ then
    $MM = \mathrm{libsvmTrain}\ (L, X, op)$
    return Mouse Model, $MM$
else if $O == 1$ And $MM$ is exist then
    $[Ans, Acc] = \mathrm{libsvmPredict}\ (L, X, MM, op)$
    return $[Ans, Acc]$
else
    logging illegal options
end if
```

Algorithm 2: Mouse Model's Train And Test.

authentication and interception of illegal users. In conclusion, our system has made the balance among the above factors and selected the time slice length as 5 min and the session length as 100 effective mouse events.

3.4. Application Usage Recognition Model. Different from the biological behaviour feature recognition based on keystroke and mouse, the application feature recognition is based on the statistical analysis of the user's application records, and mines the user's behaviour features. When the system is deployed, it analyses the user application records in the current time window in the form of sliding window, extracts the features and standard model library for verification, and completes the authentication. Compared to the behavioural feature model, the data acquisition cycle of the application recognition model is longer, but the number of users' core applications is relatively fixed. So, the application features are more stable, which remedies the shortcoming of strong real-time but insufficient stability of keystroke and mouse recognition in multimodal recognition.

3.4.1. Application Data Collection. Application data collection mainly captures the process information, and our system adopts the Windows API PSAPI library to finish this work. When the user starts the system, the current process data are initialized through loading dynamic DLL.

First, *EnumProcesses* () enumerates the ongoing processes, counts the total number of processes, and obtains detailed data of each process (time, process ID, process name, process path) as shown in Table 6.

Second, when recording application processes data, our system adds process state, construct times, destroy times, and total running time fields according to the current process information, and initializes the value as 1, 1, 0, $t + \Delta t$.

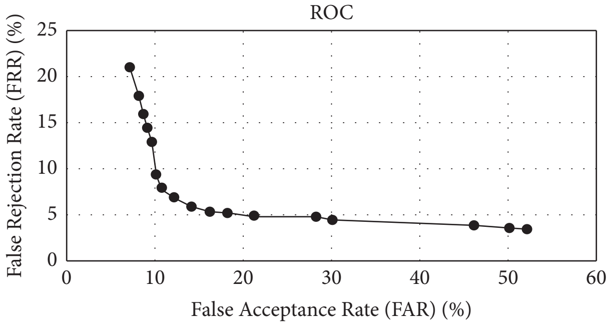


Figure 6: ROC curve graph for session length $X = 100$.

Table 5: Mouse session length and classification error rate.

Session length	Session duration (min)	ERR (%)
50	2.7	17.93
100	5.3	9.27
200	12.9	7.11

After that, our system cyclically monitors the process status, records the process construct, destroys the events, and updates the times and the total running time of the process. The specific process collection information is shown in Table 7.

3.4.2. Application Data Preprocessing and Feature Selection. The application-based identity recognition model is a statistics-based classification model, so it does not involve multidimensional features, and does not require complex dimensionality reduction and feature selection. After users log into the system, they are allowed to make basic system settings and manually select the list of applications to be monitored. Therefore, our system only carries out statistical processing for monitoring applications defined by users. First, the nontarget application information in the collected dataset will be eliminated, and then the event update frequency, running time, and total proportion of each target application are calculated. When the user does not define a monitoring application, the entire application process is handled by default.

3.4.3. Application Data Training and Test. For the training and prediction of the application recognition model, our system uses the Naive Bayes algorithm based on *sklearn naive_Bayes* library. The idea is to conduct model training according to the existing user characteristics and classification results. After the training is completed, the probabilities of each feature belonging to a different category are calculated in the testing stage as the final classification results. Therefore, it is also known as the classification algorithm based on statistics, and the related processing procedure is shown as follows.

(1) Assume that $X = \{x_1, x_2, ..., x_n\}$ is a user to be classified, and each user contains n application feature x_i

(2) The result of user identity classification is $Y = \{0, 1\}$, in which 0 means illegal user and 1 means legal user

(3) Calculate the probability $P\ (Y_i \mid x)$ that x belongs to the classification result Y, and $P\ (Y \mid x) = \text{Max}\ \{P\ (Y_1| x), P\ (Y_2 \mid x))$

Algorithm 3 shows the training procedure. First, the data of the training set is normalized and transformed. Second, the Gaussian Bayesian algorithm in naive Bayes [55] is selected for model training. After the training is completed, the fitting process is carried out, and the recognition model PM is finally output. Algorithm 4 describes the testing procedure when applying the recognition model. In the testing stage, our system normalizes the test data and computes the classification result *Ans* according to the existing training model PM and the test data set X, and calculates the output confusion matrix *Acc* according to the predefined indicators.

4. MFCA System Design and Procedure

In this section, we will introduce the design methods and procedure of the MFCA system. The MFCA system introduces multimodal fusion to analyse the collected multidimensional user behaviour characteristics, performs model training according to the characteristics, and generates the trust model to achieve continuous authentication.

The MFCA system mainly consists of three parts: first, the keystroke model, mouse model, and application model obtained from training based on keystroke data, mouse data, and application record, respectively; second, the multimodal fusion technique used to merge the classification results of the three models; third, the trust model algorithm used for continuous identity authentication.

Figure 7 shows the overall design of the MFCA system, and it can be divided into training stage and testing stage, in which the training stage mainly completes the training and fusion of multidimensional behaviour models and finally generates the trust model. In the testing stage, the authentication mechanism verifies the real-time behaviour characteristics of users through the trust model and exports the current trust score. When the trust score is lower than the predefined trust threshold, the current user is judged to be an illegal user and the MFCA system will lock the device, generate alarm, and prevent the user from using the devices. When the trust score is higher than the trust threshold, the MFCA system determines that the current user's identity is legitimate, and the user can continue to use without interference and any processing.

The following parts describe the MFCA from three aspects of model training and testing, multimodal fusion mechanism, and trust algorithm design.

4.1. Model Training and Prediction. The multidimensional behaviours of network users mainly consist of keystrokes, mouse, and application usage. Our system designs the multimodal fusion mechanism to collect user behaviour data and combines them effectively, adopts the multiple classifier fusion to avoid the limitation of the single classification and improve the accuracy of the classification results and generalized capability, and finally realizes continuous authentication.

The multidimensional behaviour model (keystroke model, mouse model, and application model) mainly consists of three stages: model establishment, training, and prediction, as shown in Figure 8. When a user registers for the first time, the system will default this user as legitimate and collect the data to establish the initial model. After that, the model will constantly update and evolve with the increase of the multidimensional behaviour data. Therefore, the authentication system will continuously collect users' multidimensional characteristic data, update the model to fit the current user behaviour characteristics while verifying the identity, and improve the identity recognition accuracy.

In the training stage, once either of the models is updated, the authentication system will trigger the iterative updating of the trust model to make the model learn the characteristics of the current users and ensure the timeliness of the model. In the test phase, once a behaviour such as keystroke has enough data for feature extraction, the MFCA system will depend on these characteristics through the trust model to generate the corresponding result. The trust model will convert multiple results as the latest trust score based on their predefined proportion, and compare it with the trust threshold to determine the legitimacy of user identity, and decide whether or not to trigger alarms.

4.2. Multimodal Fusion Mechanism. Multimodal fusion (known as multi-classifier fusion) is designed to effectively combine multidimensional features for decisions, avoids the

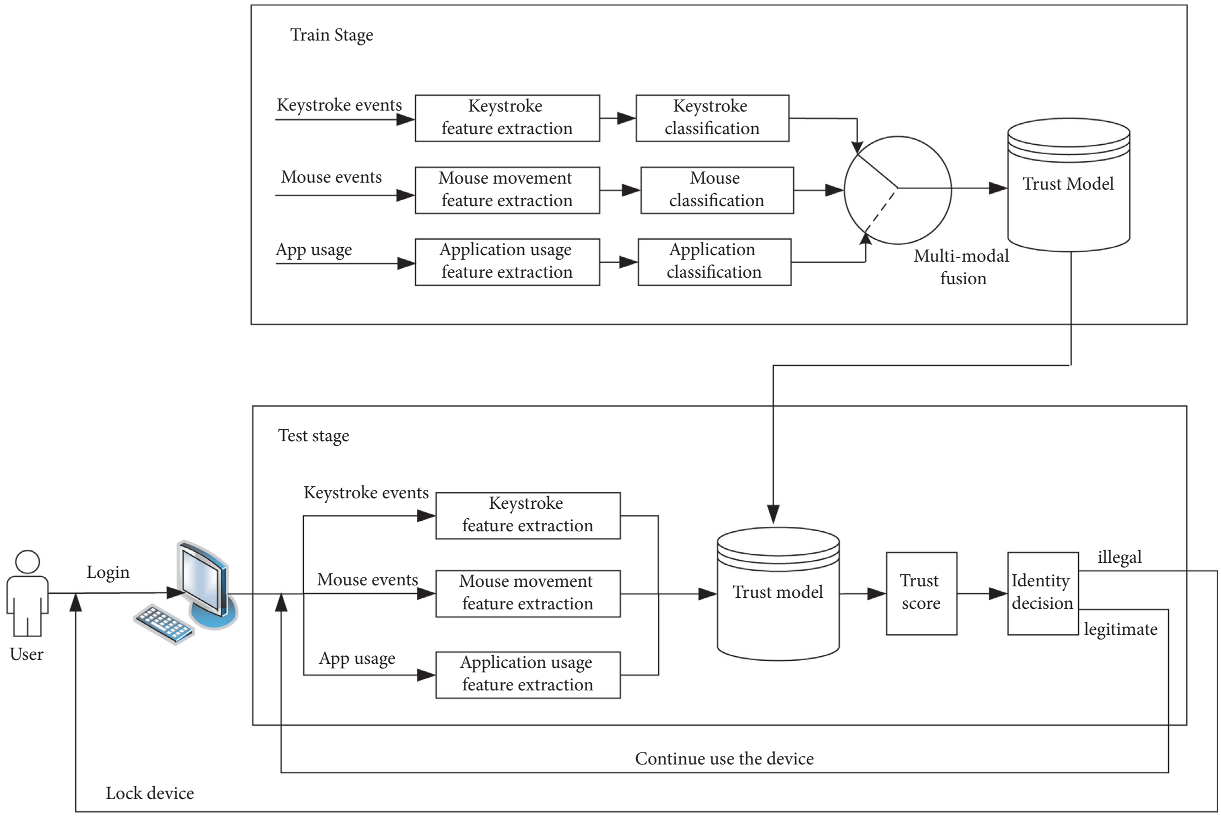


Figure 7: The overall design of the MFCA system.

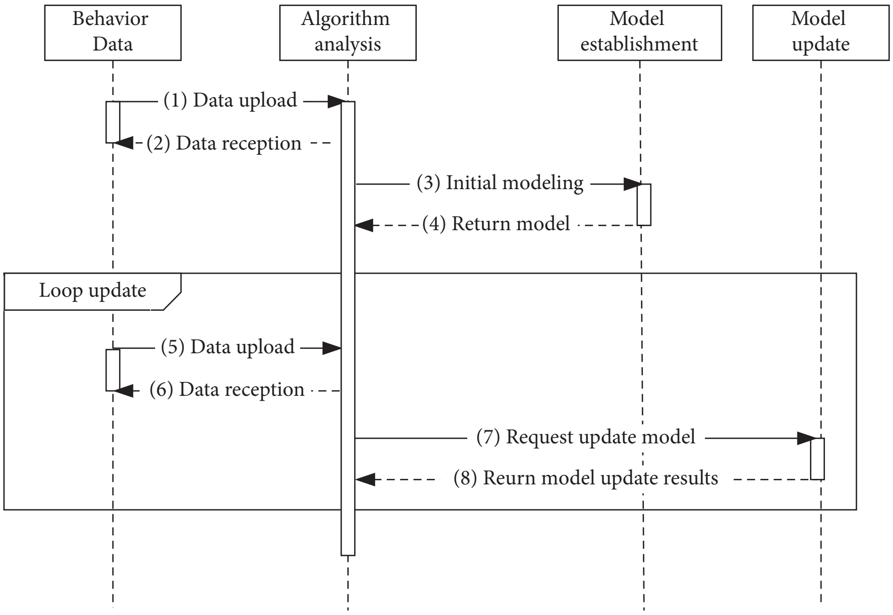


Figure 8: Model update sequence diagram.

limitations of single classification, and improves the accuracy and generalization of classification results by fusing multiple models finally. When performing continuous authentication, the complementarities among multidimensional behaviours need to be considered. In our work, three types of data, namely, keystroke, mouse movement, and application record, are collected as they have natural complementarity when users interact with the

Table 6: Initial application process information.

Time	Pid	ProcessName	ProcessPath
2019/12/11 18 : 52 : 01	10924	WeChat. exe	D:\SoftWare\WeChat\WeChat.exe
2019/12/11 18 : 52 : 01	36172	firefox. exe	D:\SoftWare\Firefox\firefox.exe
2019/12/11 18 : 52 : 01	27064	VISIO. EXE	C:\SoftWare\Visio\Office16\VISIO.EXE
......	......	......	......
2019/12/11 18 : 52 : 01	7740	KuGou. exe	D:\Download \KuGou\KuGou.exe
2019/12/11 18 : 52 : 01	6772	Microsoft. Photos. exe	C:\Program Files\WindowsApps

Table 7: Application data collect information.

Field	Type	Description
pName	String	Process name
Pid	Int	Process ID
pEvent	Int	Process event type
pStamp	Long	Process event timestamp
pPath	String	Process path
pStatus	Int	Process status
newTimes	Int	The number of process construct
delTimes	Int	The number of process destroy
totalAliveTime	Int	The duration of process

computer. Three kinds of models have different abilities to recognize users. Therefore, the MFCA can cover the using habit of different users based on multiple classifier fusion to improve the accuracy.

4.3. Trust Model Algorithm. Take behaviours of keystroke, mouse, and application as examples to describe the trust score algorithm, as shown in Algorithm 5, where the MCF adopts parallel combination, the outputs of the three base classifiers are all predefined binary values (illegal = 0 or legal = 1), and the exported results are labelled rather than probability values. The weighted voting method is selected to complete the multi-classifier fusion.

Taking keystroke identification model of double key pair "an" for example, "an" appears in the double characteristics of the weight for the W^f = Count (an)/Count (keyFeature); when a user types "an" and is identified as a legitimate user, the model exports classification results $FC = 1$. At this point, the system will reward the user with $W^f * W^K * R$, and update the trust score $\text{Trust}_i = \text{Trust}_{i-1} + W^f * W^K * R$. It should be noted that the new trust score will be no more than the maximum threshold $T_{\max}$. On the contrary, the system will punish the user with $W^f * W^K * P$ and update the trust score $\text{Trust}_i = \text{Trust}_{i-1} - W^f * W^K * P$. However, the new trust score should be no less than the minimum threshold $T_{\min}$. After obtaining the trust score Trust_i, the system will determine whether the trust score is lower than the alarm threshold T_{alert}. If the trust score is lower than the threshold T_{alert}, the system will set the warning sign Alert = 1 and trigger the alarm.

5. Performance Analysis of MFCA System

In the above, we have introduced the MFCA system and its sub-modules in detail. In this section, we will describe the experiment procedure and analyse the performance of the MFCA system in detail.

5.1. Experiment Dataset. The whole experiment scenario is free environment without static authentication, and the system does not require the user to type the specified statement to unlock the device or sign the gesture through the mouse. From data collection to authentication, the user maintains normal operations without additional restriction requirement, so that he/she can almost ignore the existence of our system except the alarm. In order to facilitate the experiment, 22 participants have been recruited to operate on computer in their daily life which can insure the continuity and integrity to reduce the impact of uncertain factors. The data are collected over three weeks after the installation of our system.

In addition, the system is applied to the general scenario rather than the strict laboratory environment. The system design considers the function and universality with the objective of balancing the application condition and the application effect, and ensures the high reliability of the characteristics selection and model training. As shown in Table 8, the computer and hardware are slightly different, but they all run on the basic Windows environment. In the keyboard and mouse equipment, the user selects the general qwerty keyboard and double key mouse. The equipment manufacturers are different, but the impact of the key feature collection of the system can be ignored.

5.2. Evaluation Metrics. After the system was deployed, 22 participants were tested to verify the performance of the MFCA system. Most previous research work adopts FAR and FRR to evaluate performance. However, it is not important to know whether an imposter or illegal user is detected, but when the illegal user is detected. In fact, FAR and FRR are more suitable for one-time authentication scenarios. They can only indicate whether an illegal user is detected but cannot indicate when an illegal user can be

```
Input:
    The process train dataset matrix $X$;
    The label vector of process data, $L := L^1, L^2, \ldots, L^n$;
Output:
    The Process Model, $PM$;
initialize: do preprocess, $scalar = \text{MinMaxScaler ( )}$
initialize: $X = \text{scalar. fit\_transform } (X)$
$PM = \text{GaussianNB ( )}$
$PM.\ \text{fit } (X, L)$
return $PM$
```

Algorithm 3: Process Model's Train.

```
Input:
    The label vector of process data, $L$;
    The process test dataset matrix $X$;
    The process model, $PM$
Output:
    The predicted answer, $Ans$;
    The predicted accuracy information matrix, $Acc$;
initialize: do preprocess, $scalar = \text{MinMaxScaler ( )}$
initialize: $X = \text{scalar. fit\_transform } (X)$
$predicted = PM.\ \text{predict } (X)$
$Ans = \text{metric. classification report } (L, predicted)$
$Acc = \text{metrics. confusion matrix } (Ans, predicted, L)$
return $[Ans, Acc]$
```

Algorithm 4: Process Model's Test.

detected which is more important in a continuous authentication scenario. For example, even if the recognition rate of a model is high, but the detection time is long, the intrusion may have been completed before illegal users are detected, which is unacceptable. Different from the previous performance evaluation metrics, this paper adopts Average Number of Imposter Actions (ANIA) and Average Number of Genuine Actions (ANGA) to evaluate the application effect of the system, where ANIN refers to the average number of behavioural characteristics required for illegal users to be identified as exceptions, and ANGA refers to the average number of behavioural characteristics used by legitimate users to be identified as exceptions. Therefore, ANIA should be as low as possible, so that ANIA users can be identified more quickly and in less time, which can perform fewer illegal operations. ANGA should be as high as possible so that legitimate users can work without interruption as much as possible.

5.3. Experimental Results. In the experiment, 22 participants are divided into two groups: one group comprise legal users' normal use of their own equipment, and the other group comprise illegal users' operation of others' equipment. The whole experimental environment does not have other restrictive requirements. We take the first 70% of the user's input data as training data and the others as test data. The following operations are performed on all users' input data: first, our system uses the training data of legitimate users for model training; second, the test data are used to calculate the Number of Genuine Action (NGA) of the model; finally, the data of illegal users are used to attack and the Number of Imposter Action (NIA) of the model is calculated. The initial trust score of all users is 90. When the trust score is below the threshold of 75, the pop-up alarm will be triggered, and the system will record the verification times of each feature to obtain NIA and NGA, and calculates the ANIA and ANGA. The experimental data of the two groups are shown in Tables 9 and 10.

As shown in Tables 9 and 10, ANIA = 430 and ANGA = 7341, which means that the average illegal user can be identified in the 430 features input, the legal user has an average of 7341 characteristics input. Note that an effective feature here is not a user behaviour. Take a mouse operation as an example; an effective mouse movement that contains multidimensional features such as moving distance, moving speed, and moving direction, so that the authentication speed will accelerate as the user performs the features frequently. The capture period of illegal users is shorter, which can realize the user exception in a short time. The normal using period of the legitimate user is longer; therefore, the daily work will rarely be interrupted. In addition, to speed up the abnormal authentication

```
Input:
    The feature type, $F_t$
    The feature classification results, FC;
    The weight of this feature in its recognition model, $W^f$;
    The trust score after last calculation, $\text{Trust}_{i-1}$;
    The initial trust score $T$, and score threshold $T_{\max}$ and $T_{\min}$;
    The score will trigger system alert $T_{alert}$;
    The reward and punishment score for each feature, $R$, $P$;
    The weight of Three authentication model, $W\colon = W^k, W^m, W^p$;
Output:
    The trust score after this calculation, $\text{Trust}_i$;
    IF alert the system trust, $A$;
initialize: Init $A = 0$, if the first calculation, $\text{Trust}_i = T$;
if $F^t = = 0$ then
    if $FC == 0$ then
        $\text{Trust}_i = \max(\text{Trust}_{i-1} - W^f * W^k * P, T_{\min})$;
    else {$FC == 1$}
        $\text{Trust}_i = \min(\text{Trust}_{i-1} + W^f * W^k * R, T_{\max})$;
    end if
else if $F^t == 1$ then
    replace $W^k$ in the above formula with $W^m$;
else {$F^t == 2$}
    replace $W^k$ in the above formula with $W^p$;
end if
if $\text{Trust}_i < T_{\text{alert}}$ then
    $A = 1$;
end if
return [$\text{Trust}_i, A$]
```

Algorithm 5: Trust Score Calculation.

Table 8: Summary of experiment setting.

Number of participants	15	7
Device types	PC	Notebook
OS	Win7	Win10
Resolution	1440∗900	1920∗1080

Table 9: Illegal user authentication NIA results.

User ID	Keystroke	Mouse	Applicate	NIA
1	124	96	43	263
2	87	157	37	281
3	141	133	43	317
4	180	121	59	360
5	209	117	48	374
6	155	213	51	419
7	281	106	79	466
8	142	231	86	471
9	189	201	97	487
10	143	239	96	521
11	279	385	115	779

Table 10: Legal user authentication NGA results.

User ID	Keystroke	Mouse	Applicate	NGA
1	1009	3042	586	4637
2	1459	3533	589	5581
3	1356	3687	901	5944
4	2217	3791	524	6532
5	2699	3664	627	6990
6	4231	2070	1138	7439
7	3715	2984	872	7571
8	5357	2158	440	7955
9	5751	1752	746	8249
10	4970	3941	656	9567
11	6628	2599	1066	10293

speed or prevent the user from being disturbed, our system can increase or reduce the trust threshold of the trust model.

6. Conclusion

This paper proposes a continuous authentication system based on multidimensional behaviour characteristics, which introduces the trust value that is changed in real-time with the user behaviour characteristics. Only when the trust score is lower than the predefined trust threshold, the current user is considered to be an illegal user and the alarm is triggered. This system fully considers the instability of biological characteristics, avoiding the nonblack and white decision of single extreme characteristics, and improving the use of real users without the relaxation of the abnormal user. In addition, in the calculation of the trust value, the system is based on the accuracy of the multiple classification models, and the reliability of the calculation can be guaranteed.

The MFCA system has the advantages of low cost and user-friendliness because of no additional hardware equipment and no additional users' operations. Therefore, the MFCA system is important for the realization of a continuous user authentication system and especially suitable for office environments with high security requirements such as finance corporations and online examination. The adoption of the MFCA can prevent the insider attacks and support the zero trust architecture. However, how to determine the trust threshold and improve performance need to be considered. Besides, the score-level fusion mechanism introduces additional calculation time, which will increase the fusion time. So, the other fusion mechanism such as rank-level fusion mechanism will be considered in our future work.

Besides, in order to apply our model in real-life scenarios, we must consider the problem of user data privacy protection. A privacy attack on a machine-learning model may expose personal information. For example, the attacker may obtain the user's mouse movement characteristics, keystroke characteristics, and application usage characteristics by attacking our model and infer the user's private information such as login passwords, private letters by analysing the characteristics in a certain period of time. In future work, we will consider analysing possible attack scenarios against the models and introduce data anonymisation and differential privacy mechanisms to protect user data privacy.

Data Availability

The source data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 12 months after the publication of this article, will be considered by the corresponding authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Key Research and Development Program (2018YFE0206800) and in part by the project under contract No. 2019-JCJQ-ZD-182-00-02.

References

[1] S. Yao, J. Guan, Y. Wu, K. Xu, and M. Xu, "Toward secure and lightweight Access authentication in SAGINs," *IEEE Wireless Communications*, vol. 27, no. 6, pp. 75–81, 2020.

[2] I. Stylios, S. Kokolakis, O. Thanou, and S. Chatzis, "Behavioral biometrics & continuous user authentication on mobile devices: a survey," *Information Fusion*, vol. 66, pp. 76–99, 2021.

[3] R. Mehra, A. Meshram, and B. R. Chandavarkar, "Remote user authentication and issues: a survey," in *Proceedings of the 2020 11th, International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, IEEE, Kharagpur, India, July 2020.

[4] M. Abuhamad, A. Abusnaina, D. Nyang, and D. Mohaisen, "Sensor-based continuous authentication of smartphones' users using behavioral biometrics: a contemporary survey," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 65–84, 2020.

[5] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: on the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, 2013.

[6] "Insider threat report [EB/OL]," 2020, https://www.cybersecurity-insiders.com/wp-content/uploads/2019/11/2020-Insider-Threat-Report-Gurucul.pdf,%202021-05.

[7] S. Teerakanok, T. Uehara, and A. Inomata, "Migrating to zero trust architecture: reviews and challenges," *Security and Communication Networks*, vol. 2021, Article ID 9947347, 10 pages, 2021.

[8] R. Spillane, "Keyboard apparatus for personal identification," *IBM Technical Disclosure Bulletin*, vol. 173346 pages, 1975.

[9] G. E. Forsen, M. R. Nelson, and R. J. J. Staron, "Personal attributes authentication techniques," Pattern Analysis and Recognition Corp, Technology Report, NTIS No. 197805, 1977.

[10] R. Gaines, W. Lisowski, and S. Press, "Authentication by keystroke timing: some preliminary results," Rand Corporation: Rand Report R-2560-NSF, The Rand Corporation, Santa Monica, CA, USA, 1980.

[11] R. A. J. Everitt and P. W. Mcowan, "Java-based Internet biometric authentication system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1166–1172, 2003.

[12] N. Zheng, A. Paloski, and H. Wang, "An efficient user verification system using angle-based mouse movement biometrics," *ACM Transactions on Information and System Security*, vol. 18, no. 3, pp. 1–27, 2016.

[13] U. Mahbub, J. Komulainen, D. Ferreira, and R. Chellappa, "Continuous authentication of smartphones based on application usage," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 1, no. 3, pp. 165–180, 2019.

[14] M. Ehatisham-Ul-Haq, M. Awais Azam, U. Naeem, Y. Amin, and J. Loo, "Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing," *Journal of Network and Computer Applications*, vol. 109, pp. 24–35, 2018.

[15] R. Kumar, V. Phoha, and R. Raina, "Authenticating users through their arm movement patterns," 2016, http://arxiv.org/abs/1603.02211.

[16] Y. Li, "Research on gesture recognition model and its application based on wear sensing perception," pp. 1–45, Lanzhou University, Lanzhou, China, 2019, Master's Thesis.

[17] J. Handa, S. Singh, and S. Saraswat, "A comparative study of mouse and keystroke based authentication," in *Proceedings of the 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 670–674, Noida, India, January 2019.

[18] P. H. Pisani, A. C. Lorena, and P. L. F. De Carvalho, "Adaptive biometric systems using ensembles," *IEEE Intelligent Systems*, vol. 33, no. 2, pp. 19–28, 2018.

[19] M. Liu, "Research on authentication technology based on user keystroke behavior," pp. 1–80, Beijing University of Posts and Telecommunications, Beijing, China, 2019, Master's Thesis.

[20] D. Gunetti and C. Picardi, "Keystroke analysis of free text," *ACM Transactions on Information and System Security*, vol. 8, no. 3, pp. 312–347, 2005.

[21] P. Dowland, H. Singh, and S. Furnell, "A preliminary investigation of user authentication using continuous keystroke analysis," in *Proceedings of the IFIP 8th Annual Working Conference on Information Security Management & Small Systems Security*, Las Vegas, NV, USA, September 2001.

[22] T. Shimshon, R. Moskovitch, L. Rokach, and Y. Elovici, "Continuous verification using keystroke dynamics," in *Proceedings of the 2010 International Conference on Computational Intelligence and Security*, pp. 411–415, Naning, China, December 2010.

[23] M. Rybnik, M. Tabedzki, M. Adamski, and K. Saeed, "An exploration of keystroke dynamics authentication using nonfixed text of various length," in *Proceedings of the International Conference on Biometrics and Kansei Engineering*, pp. 245–250, Tokyo, Japan, July 2013.

[24] X. Song, P. Zhao, M. Wang, and C. Yan, "A continuous identity verification method based on free-text keystroke dynamics," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 206–210, Budapest, Hungary, October 2016.

[25] J. Huang, D. Hou, and S. Schuckers, "A practical evaluation of free-text keystroke dynamics," in *Proceedings of the IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pp. 1–8, New Delhi, India, February 2017.

[26] B. Ayotte, M. K. Banavar, D. Hou, and S. Schuckers, "Fast and accurate continuous user authentication by fusion of instance-based, free-text keystroke dynamics," in *Proceedings of the International Conference of the Biometrics Special Interest Group*, pp. 1–6, Darmstadt, Germany, September 2019.

[27] M. Pusara and C. E. Brodley, "User Re-authentication via mouse movements," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pp. 1–8, Association for Computing Machinery, New York, NY, USA, October 2004.

[28] Y. Aksari and H. Artuner, "Active authentication by mouse movements," in *Proceedings of the 24th International Symposium on Computer and Information Sciences*, pp. 571–574, Suzelyurt, Cyprus, September 2009.

[29] B. Sayed, I. Traore, I. Woungang, and M. S. Obaidat, "Biometric authentication using mouse gesture dynamics," *IEEE Systems Journal*, vol. 7, no. 2, pp. 262–274, 2013.

[30] C. Chao Shen, Z. Zhongmin Cai, X. Xiaohong Guan, Y. Du, and R. A. Maxion, "User authentication through mouse dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 16–30, 2013.

[31] C. Shen, Z. Cai, and X. Guan, "Continuous authentication for mouse dynamics: a pattern-growth approach," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, pp. 1–12, Boston, MA, USA, June 2012.

[32] E. Medvet, A. Bartoli, F. Boem, and F. Tarlao, "Continuous and non-intrusive reauthentication of web sessions based on mouse dynamics," in *Proceedings of the 9th International Conference on Availability, Reliability and Security*, pp. 166–171, Fribourg, Switzerland, September 2014.

[33] B. Li, W. Wang, Y. Gao, V. Phota, and Z. Jin, "Hand in motion: enhanced authentication through wrist and mouse movement," in *Proceedings of the IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–9, Rendondo Beach, CA, USA, October 2018.

[34] M. Yildirim and E. Anarim, "Session-based user authentication via mouse dynamics," in *Proceedings of the 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, Sivas, Turkey, April 2019.

[35] Á Fülöp, L. Kovács, T. Kurics, and E. Windhager-Pokol, "Balabit mouse dynamics challenge data set," Available at: https://github.com/balabit/Mouse-Dynamics-Challenge, 2016.

[36] S. Liu, "Research on user behaviour analysis model based on log data," pp. 1–63, Yunnan University, Kunming, China, 2017, Master's Thesis of.

[37] W. Meng, Y. Wang, D. S. Wong, S. Wen, and Y. Xiang, "TouchWB: touch behavioral user authentication based on

web browsing on smartphones," *Journal of Network and Computer Applications*, vol. 117, pp. 1–9, 2018.

[38] J. Wei, "Analysis and research of user access behavior based on DNS log," pp. 1–75, Beijing Jiaotong University, Beijing, China, 2019, Master's Thesis of.

[39] A. Alzubaidi, S. Roy, and J. Kalita, "A data reduction scheme for active authentication of legitimate smartphone owner using informative apps ranking," *Digital Communications and Networks*, vol. 5, no. 4, pp. 205–213, 2019.

[40] N. Eagle, A. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15274–15278, 2009.

[41] S. K. S. Modak and V. K. Jha, "Multibiometric fusion strategy and its applications: a review," *Information Fusion*, vol. 49, pp. 174–204, 2019.

[42] I. Traore, I. Woungang, M. S. Obaidat, N. Youssef, and L. Iris, "Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments," in *Proceedings of the Fourth International Conference on Digital Home*, pp. 138–145, IEEE Computer Society, Guangzhou, China, November 2012.

[43] K. O. Bailey, J. S. Okolica, and G. L. Peterson, "User identification and authentication using multi-modal behavioral biometrics," *Computers & Security*, vol. 43, pp. 77–89, 2014.

[44] L. Fridman, A. Stolerman, S. Acharya et al., "Multi-modal decision fusion for continuous authentication," *Computers & Electrical Engineering*, vol. 41, pp. 142–156, 2015.

[45] S. Mondal and P. Bours, "Combining keystroke and mouse dynamics for continuous user authentication and identification," in *Proceedings of the 2016 IEEE International Conference on Identity, Security and Behavior Analysis*, pp. 1–8, ISBA, Sendai, Japan, February 2016.

[46] I. D. S. Beserra, L. Camara, and M. D. Costa-Abreu, "Using keystroke and mouse dynamics for user identification in the online collaborative game league of legends," in *Proceedings of the 7th International Conference on Imaging for Crime Detection and Prevention (ICDP 2016)*, November 2018.

[47] S. M. Sergio, R. S. Baker, O. C. Santos, and J. González-Boticario, "A machine learning approach to leverage individual keyboard and mouse interaction behavior from multiple users in real-world learning scenarios," *IEEE Access*, vol. 6, pp. 39154–39179, 2018.

[48] K. Quintal, B. Kantarci, M. Erol-Kantarci, A. Malton, and A. Walenstein, "Contextual, behavioral, and biometric signatures for continuous authentication," *IEEE Internet Computing*, vol. 23, no. 5, pp. 18–28, 2019.

[49] R. Wang and D. Tao, "Implicit authentication mechanism based on context awareness for smartphone," *Journal of Beijing University of Posts and Telecommunications*, vol. 42, no. 6, pp. 118–125, 2019.

[50] Y. Yang, J. Sun, and L. Guo, "PersonaIA: a lightweight implicit authentication system based on customized user behavior selection," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 113–126, 2019.

[51] S. Vhaduri and C. Poellabauer, "Multi-modal biometric-based implicit authentication of wearable device users," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3116–3125, 2019.

[52] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[53] C.-C. Chang and C.-J. Lin, "Libsvm," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

[54] S. Almalki, P. Chatterjee, and K. Roy, "Continuous authentication using mouse clickstream data analysis," in *Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 76–85, Springer, Atlanta, GA, USA, July 2019.

[55] I. Rish, "An empirical study of the naive Bayes classifier," *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, pp. 41–46, 2001.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5536865, 8 pages
https://doi.org/10.1155/2021/5536865

Research Article

Task Price Prediction Based on Clustering and DNN in Crowdsensing

Bing Jia,[1] **Xi Luo**,[1] **Tao Feng**,[1] **and Yan Jia**[2]

[1]*College of Computer Science, Inner Mongolia University, Hohhot 010021, China*
[2]*College of Cyber Science, Nankai University, Tianjin 300350, China*

Correspondence should be addressed to Yan Jia; jiay@nipc.org.cn

Received 19 February 2021; Revised 27 June 2021; Accepted 25 August 2021; Published 12 October 2021

Academic Editor: Hao Peng

With the popularization of mobile devices and the development of wireless networks, crowdsensing is devoted to providing universal Internet of Things services. A reasonable task pricing mechanism can not only motivate more users to participate in the sensing task but also help the benign development of crowdsensing platform, so it has gradually become a research hotspot in the field of crowdsensing. Aiming at the common problems of insufficient analysis of task pricing rules and large deviations of pricing prediction models, a task price prediction method based on clustering and DNN is proposed. Using the real historical trade price set as the data source, natural grouping and taxonomic description of task price are realized by exploring sensing task pricing law with complex constraint relation using two-step clustering analysis. On the basis of the above, the price interval prediction model based on DNN is implemented. The experimental results show that the predicting accuracy of the pricing mechanism is higher than 82.7%.

1. Introduction

The increasing demand for practical applications of the Internet of Things, the widespread popularity of mobile smart terminals, and the emergence of the crowd computing model [1] have jointly spawned the emerging concept of crowdsensing. Crowdsensing is a new data acquisition mode that combines crowdsourcing ideas and mobile device perception capabilities and is committed to providing universal Internet of Things services for the public. Crowdsensing uses mobile sensing devices carried by nonprofessional field personnel to realize the distribution of sensing tasks and the collection of sensing data through conscious or unconscious collaboration, which breaks through the barriers that rely solely on professional participation [2]. Crowdsensing integrates GPS, cameras, gyroscopes, microphones, and other sensors. Mobile devices rely on human behavior to perform large-scale, complex sensing tasks and provide rich sensing data. This "people-centered" sensing network [3] overcomes the shortcomings of high networking cost, inflexible deployment, and difficult maintenance in the traditional fixed deployment mode.

Due to the above advantages, crowdsensing is widely used in many aspects of real life. For example, in public security application scenarios, Haddawy et al. developed a smart phone disaster warning system based on Mobile4D using crowdsensing, providing real-time crisis warning and detailed situational awareness information [4]. In the application of environmental monitoring, Oscar et al. monitored air pollution in crowded cities with the help of crowdsensing [5]. In addition, crowdsensing also has a wide range of practical applications in social services [6] and other aspects [7, 8], so it has received extensive attention and a lot of research from domestic and foreign researchers. As an emerging research field of the Internet of Things, the research of crowdsensing mainly includes task pricing [9, 10], task allocation [11, 12], data transmission [13, 14], incentive mechanism [15, 16], etc. With the continuous exploration and resolution of these problems by domestic and foreign researchers, crowdsensing will eventually serve the society in a brand-new way.

Typical crowdsensing is composed of two subjects: a platform and a user carrying a mobile perception device [17], where users include task publishers and task participants, as shown in Figure 1. The platform is mainly responsible for publishing tasks and remuneration for hosting tasks. The task publisher is mainly responsible for perceiving the task release and providing remuneration. The responsibility of the task participant is to complete the task and get paid. It is a reasonable and effective way to pay users who participate in the task in the process of completing the task, so the pricing is particularly important. Essentially, it is to treat the sensing task as a commodity that can be bought and sold in the free market.

The main contributions of this paper are as follows:

(i) We use real historical datasets as data sources to construct task pricing standard datasets to improve the accuracy and efficiency of task pricing law analysis. Natural grouping and taxonomic description of task price are realized by exploring sensing task pricing law with complex constrain relation using two-step clustering analysis; the natural classification and group description of the sensing task prices are realized, thereby reflecting the pricing law of sensing tasks.

(ii) According to the obtained clustering dataset, DNN is used for batch training and analysis and optimization, and the price range prediction model based on deep neural network is realized, which completes the price range prediction of the perception task and provides a scientific basis for the price decision of the sensing task.

In Section 1, the background, characteristics, architecture, and practical application of crowdsensing are briefly described, and the main contributions of this article are explained at the same time. In Section 2, this article briefly reviews the sensing task pricing analysis scheme proposed by domestic and foreign researchers and conducts research to solve the corresponding problems in view of the current status and existing problems of the previous research. Section 3 introduces a task price prediction method based on clustering and DNN. Using the real price set of perception tasks as the data source, TSCA is used to classify and describe the prices of perception tasks naturally, revealing the intrinsic classification of the prices of perception tasks, and at the same time, we use DNN to perform classification on the prices of the classified sensing tasks. We predict and conduct comparative experiments. Section 5 summarizes the full text and points out the next research direction.

2. Related Work

The task pricing of crowdsensing is mainly through the analysis of historical sensing data to explore the way of task pricing rules to determine the price of sensing tasks. The methods used in this pricing model mainly include cluster analysis, multiple regression, bivariate models, and Bayesian models. Literature [18] uses a density-based spatial clustering algorithm to cluster the density areas of the tasks in the task price dataset to optimize the pricing strategy. Literature [19] uses the same clustering method as literature [18] and introduces a proportional sharing mechanism to establish a sensing pricing optimization model that can evaluate task success rates in advance. Literature [20] uses a combination of K-means clustering analysis and multiple nonlinear regression to design the task's pricing function and analyze the reasons why the task is not completed. Literature [21] proposes a task pricing mechanism based on Bayesian model, which transforms a non-submodel optimization problem into a submodel optimization problem. Shao and others proposed a crowdsensing pricing mechanism based on a bivariate pricing model [22]. By calculating the Pearson correlation coefficient between bivariate data and pricing data, it is proved that bivariate is related to the pricing problem. Literature [23] uses the same dataset as the previous method, processes historical datasets through factor analysis, and establishes a perceptual task pricing model. Literature [24] studied the use of autoregressive methods to consider market sentiment indicators to predict US oil prices and concluded that autoregressive methods are not strong in predicting such problems and machine learning methods need to be considered.

In summary, the existing problems in existing research mainly include the following three aspects: lack of multidimensional large sample standard dataset; when exploring the law of task price, the interval price interval is defined only according to the price range of tasks in the data set. At the same time, the range or value of situational factors affecting task pricing is classified according to this interval. This processing method is difficult to fully reflect the internal law of task price in the dataset. On the price prediction method, the model method has not considered the idea of machine learning to price, which makes the price prediction deviation larger. Based on the above research status and existing problems, we propose a task price prediction based on clustering and DNN.

3. Task Price Prediction Based on Clustering and DNN in Crowdsensing

3.1. Price Classification of Perception Tasks Based on TSCA. The multidimensional large sample historical transaction data are obtained by contacting the platform authorization. The data source is Gaia Open Data Program. Through data preprocessing such as data cleaning, data merging, and data transformation, the task pricing standard dataset is constructed, and the task pricing law of the standard dataset is analyzed through TSCA. Price pricing analysis based on TSCA includes two stages: constructing clustering feature tree and natural grouping based on condensed clustering method.

The process of constructing the clustering feature tree is to insert the sample cases in the task pricing standard dataset into the clustering feature tree according to its clustering

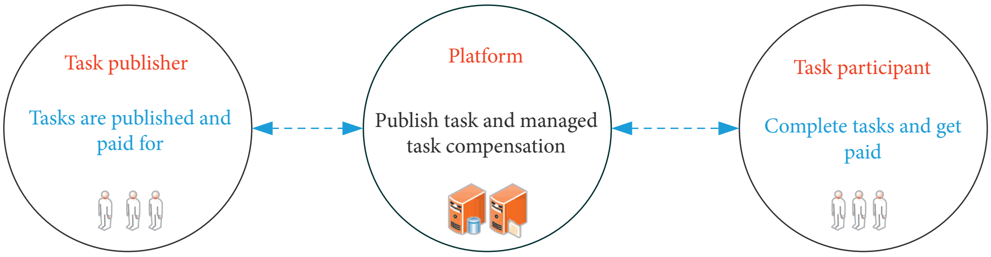

Task publisher
Tasks are published and paid for
Platform
Publish task and managed task compensation
Task participant
Complete tasks and get paid

FIGURE 1: Typical crowdsensing subject structure.

characteristics, so as to realize the growth of the clustering feature tree. At the same time, we first form several small clusters of sample cases in dense regions.

First, we define the cluster features to insert them into the cluster feature tree cluster C_j:$\overrightarrow{F}_j = \langle N_j, \overrightarrow{\widetilde{A}}, \overrightarrow{\widetilde{B}}_j, \overrightarrow{\ddot{\Gamma}}_j \rangle$, where $\overrightarrow{\widetilde{A}}_j$ denotes the linear summation of the attribute values of the sample cases in cluster C_j under continuous factors such as perceived task moving distance and task time consumption; $\overrightarrow{\widetilde{B}}_j$ denotes the sum of squares of sample case attribute values in cluster C_j under various continuous factors; and $\overrightarrow{\ddot{\Gamma}}_{jt}$ is a vector formed by the number of sample cases of each possible value under the classification factors of the first sensing task area, task type, etc.

Secondly, the distance between clusters is calculated by using the logarithmic likelihood formula according to the clustering characteristics. In order to meet the requirements of processing mixed attributes, TSCA uses logarithmic likelihood distance in distance measurement. The logarithmic likelihood distance formula and its parameter definition between cluster C_j and cluster $C_{j'}$ are

$$d\left(C_j, C_{j'}\right) = \eta_j + \eta_{j'} - \eta_{j,j'}, \tag{1}$$

$$\eta_j = -N_j\left(\frac{1}{2}\sum_{s=1}^{D_a} In\left(\widehat{\sigma}_{js}^2 + \widehat{\sigma}_s^2\right) + \sum_{t=1}^{D_b} \widehat{E}_{jt}\right), \tag{2}$$

where $\widehat{\sigma}_{js}^2 = 1/N_j \sum_{n=1}^{N_j} (\widetilde{x}_{jns} - \widetilde{x}_{jn})^2$ represents the variance of the s continuous factor value estimated from the sample case in cluster C_j; $\widehat{\sigma}_s^2 = (1/N) \sum_{n=1}^{N} (\widetilde{x}_{ns} - \widetilde{x}_s)^2$ represents the variance under the sth continuous factor estimated by all sample cases in the perceptual task price data set; and $\widehat{E}_{jt} = -\sum_{k=1}^{\delta_t} (\ddot{\Gamma}_{jtk}/N_j (\ddot{\Gamma}_{jtk}/N_j))$ represents the information entropy under the tth type factor in the cluster.

Finally, the clustering feature of cluster C_j is inserted into the clustering feature tree according to the distance between clusters, so as to realize the growth of clustering feature tree. The logarithmic likelihood distance between cluster C_j and cluster $C_{j'}$ is used to determine whether cluster $C_{j'}$ can be absorbed by cluster C_j. The subclusters corresponding to leaf elements in the final clustering feature tree are used for the next stage of clustering.

The clustering stage is to cluster J subclusters in the preclustering stage to achieve the final results. Firstly, the clustering method is combined according to the distance between clusters, until a large cluster is synthesized. Then, the approximate range of the optimal cluster number is determined by Bayesian information criterion, and the final cluster number is determined according to the distance ratio between clusters. The whole process is also called automatic clustering.

We determine the approximate range of the best cluster number by BIC. Using BIC to calculate the clustering group $J = \left\{C_1, C_2, \ldots, C_j\right\}$ to get the minimum BIC value is the optimal model of J subclusters. Then, the change quantity and ratio of BIC values of adjacent clusters are calculated, and the formulas are (3) and (4), respectively. If $\Delta\mathrm{BIC}(1) < 0$, the optimal number of clusters is 1. Otherwise, the minimum $J1$ is used as the initial estimate of the optimal number of clusters.

$$\Delta_{\mathrm{BIC}}(J) = \mathrm{BIC}\left(C_J\right) - \mathrm{BIC}\left(C_{J+1}\right), \tag{3}$$

$$r_1(J) = \frac{\Delta_{\mathrm{BIC}}(J)}{\Delta_{\mathrm{BIC}}(1)}. \tag{4}$$

Next, the optimal number of clusters is accurately determined by the ratio of the nearest cluster distance in the two clusters. The distance of the closest cluster in the cluster is $d_{\min}(C_J) = \min\left\{d(C_j, C_{j'})\colon C_j \neq C_{j'} \in C_J\right\}$, cluster C_J and C_{J+1}. The distance measurement ratio of the nearest cluster is

$$r_2(J) = \frac{d_{\min}\left(C_J\right)}{d_{\min}\left(C_{J+1}\right)}. \tag{5}$$

The automatic clustering process based on formulas (1)–(5) is shown in Table 1, indicating the process of selecting the number of clusters in clustering analysis. The final determined number of clusters is determined not only by the minimum BIC value but also by the number of clusters with the largest variation ratio and distance measurement ratio of BIC. It can be seen from the table that when the number of clusters is 5, the corresponding distance measurement ratio is the largest, which is 2.191. Therefore, $C_{J^*} = 5$ output is the final automatic clustering result.

By calculating the log-likelihood distance $d(\{x_i\}, C_j)$ of each cluster in x_i and cluster result C_{J^*}, the sample case x_i in the perceptual task dataset is put into the nearest cluster as a single point cluster. With regard to the number of sample cases included in each category, as shown in Table 2, out of 263,598 data, 101,545 sample cases were assigned to the first category (38.5%), with task participants receiving the highest remuneration in this category for completing perceived tasks. A total of 20,441 individuals were assigned to the second category (7.8%), with the lowest remuneration for completing the perception task.

Table 1: Automatic clustering.

Number of clusters	Schwarz's Bayesian criterion (BIC)	BIC change	Ratio of BIC changes	Ratio of distance measures
1	1395505.120	—	—	—
2	1031788.811	−363716.308	1.000	1.529
3	793926.416	−237862.395	0.654	1.178
4	592106.081	−201820.335	0.555	1.994
5	490969.964	−101136.117	0.278	2.191
6	444899.938	−46070.026	0.127	1.403
7	412123.535	−32776.403	0.090	1.263
8	386203.133	−25920.402	0.071	1.004
9	360383.844	−25819.289	0.071	1.152
10	337997.111	−22386.733	0.062	1.035

Table 2: Cluster distribution.

		N	% of combined	% of total
Cluster	1	101545	38.5	25.4
	2	20441	7.8	8.7
	3	27879	10.6	17.0
	4	68952	26.2	10.4
	5	44781	17.0	38.6
Total		263598	—	100.0

The description of the clustering characteristics of TSCA is shown in Table 3. With the task price as the clustering object, five categories after clustering and the specific attributes description of each category are obtained. In addition, factors are sorted according to the importance of price changes. It can be seen from the table that the most important factor affecting the price change of sensing task is the location area of sensing task, followed by the type of mobile devices that collect sensing data.

Based on the above clustering results, the perceived task price is divided into five intervals (P1, P2, P3, P4, and P5) from low to high. The perception task of task price in the P1 interval accounts for the largest proportion of the overall task, which is 38.5%. The factors have the characteristics of the shortest moving distance and the shortest task time. The factors with the above characteristics are regarded as the perception task with the lowest price. The perception task of P5 interval has the characteristics of the longest moving distance and the longest time-consuming task, which is regarded as the highest price perception task category.

3.2. Task Price Classification Prediction Based on DNN. After TSCA, the task set is divided into different categories and can describe the characteristics of the class clearly enough. Next, DNN is used for classification prediction analysis. The DNN model classifier is composed of an input layer, hidden layer, and output layer, and the nodes in each layer are connected in the form of full connection.

The input layer stores the sample cases in the task pricing standard dataset based on situational factors in the form of each column in the matrix. In addition to the attribute of task price, the class labels formed based on TSCA are stored as one-line vectors in the order of data. In addition, in order to accelerate the convergence of network parameters and make parameter initialization more reasonable, it is necessary to normalize the continuous factors such as task moving distance and task time consumption. It is also necessary to extract Onehot feature from the classification factors such as perception task area and task type.

The hidden layer is composed of full connection layer and activation layer. The activation value of the node is weighted summation of the output of the previous layer and the weight of the current layer and is obtained by the nonlinear activation function Tanh function.

The output layer uses the softmax activation function to normalize the values corresponding to all nodes and express them in the form of probability distribution. For the probability of each sample case output belonging to each category in the task pricing standard dataset, the category with the highest probability value is regarded as the most likely classification attribution. Use the softmax activation function to output the posterior probability value. Therefore, it is necessary to define the objective and output the corresponding optimization function, and the cross entropy criterion is used in the price interval prediction model. The role of CE criterion is to measure the closeness between the target classification value and the actual classification value. The smaller the CE value is, the higher the closeness is and the better the price interval prediction model will be.

The price prediction model is optimized by the stochastic gradient descent algorithm in the training process. Each time, a sample is selected from the training set for learning, so as to achieve the purpose of rapid convergence and avoid the occurrence of local optimum.

4. Experiment and Result Analysis

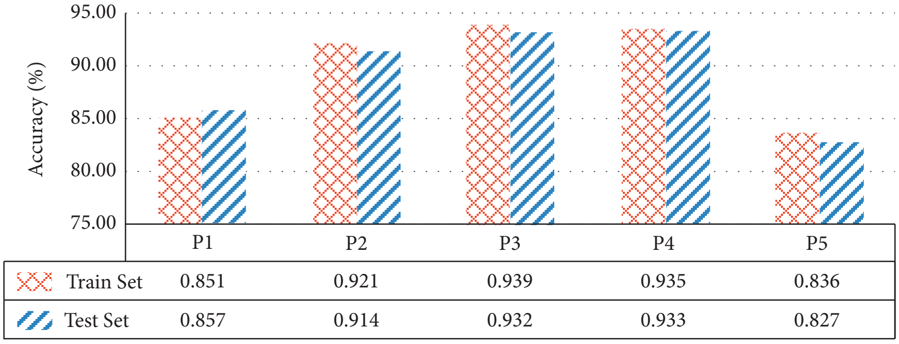
In order to verify the proposed task price prediction method based on clustering and DNN, this section will complete the relevant comparative experiments. The price prediction effect of machine learning method and the method of exploring the price law of perceptual task are compared. Using accuracy as the price interval prediction model evaluation index, the specific formula is as follows:

$$\text{accuracy} = \frac{\sum_i^n \text{correct}_i}{n}. \tag{6}$$

Among them, $i \in \{1, 2, \ldots, n\}$, where n is the number of sample cases in the test set. When the prediction interval is consistent with the actual interval, the value is 1; otherwise, it is 0.

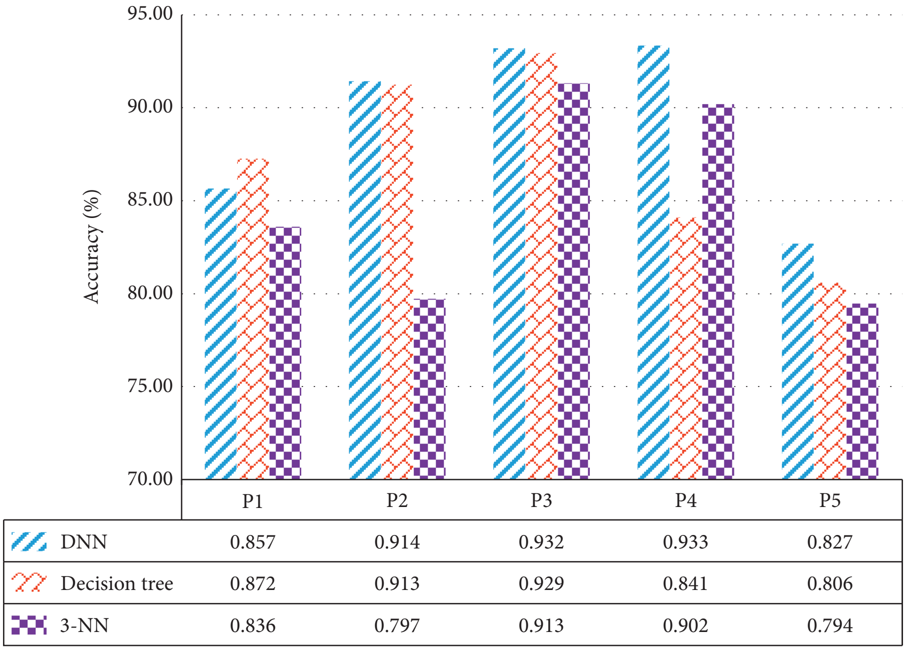
Table 3: Clustering feature description.

Attribute	1	4	5	3	2
Location	B-area (100%)	D-area (100%)	C-area (100%)	A-area (100%)	D-area (46.4%)
MDevices	Dedicated (100%)	Dedicated (100%)	Dedicated (100%)	Dedicated (100%)	Dedicated (88.7%)
SD_distance	5567.72	5864.76	6181.86	7384.69	17913.41
TaskTime	14.89	15.20	16.51	16.07	32.93
TaskType	Common (100%)	Common (97.6%)	Common (100%)	Common (100%)	Common (56.5%)
TwoDistance	3742.97	3956.39	4134.25	5420.10	13635.42
Type	R-time (100%)	R-time (100%)	R-time (100%)	R-time (100%)	R-time (79.9%)
Price	15.92	16.67	17.67	20.32	53.74



	P1	P2	P3	P4	P5
Train Set	0.851	0.921	0.939	0.935	0.836
Test Set	0.857	0.914	0.932	0.933	0.827

Figure 2: Prediction results based on TSCA and DNN.



	P1	P2	P3	P4	P5
DNN	0.857	0.914	0.932	0.933	0.827
Decision tree	0.872	0.913	0.929	0.841	0.806
3-NN	0.836	0.797	0.913	0.902	0.794

Figure 3: Comparative experiment diagram of machine learning methods.

The prediction accuracy of the proposed task price prediction method based on clustering and DNN is shown in the figure. The prediction accuracy of the price interval of the training set tends to be between 83.6% and 93.9%, and that of the test set tends to be between 82.7% and 93.3%. Among them, the accuracy rate of P3 price interval is the highest, which is because the change law of factors affecting price change in this interval is relatively single. The accuracy of the P5 price range is the lowest, which may be due to the factors that affect the price change. As shown in Figure 2, the perception task with such characteristics is classified as the task of the P5 price range.

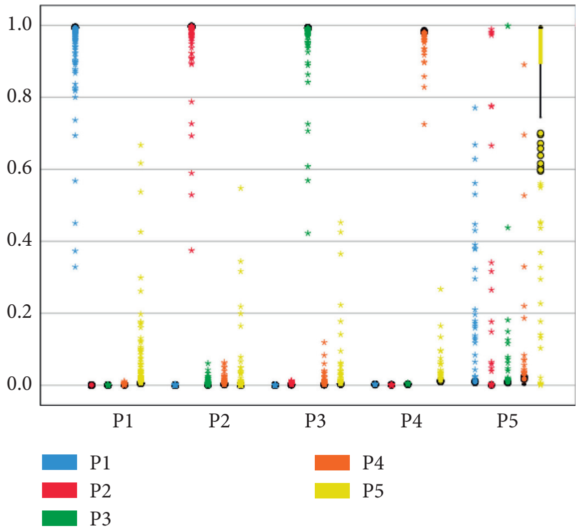


Figure 4: Predicting pseudo-probability of task price interval based on TSCA.

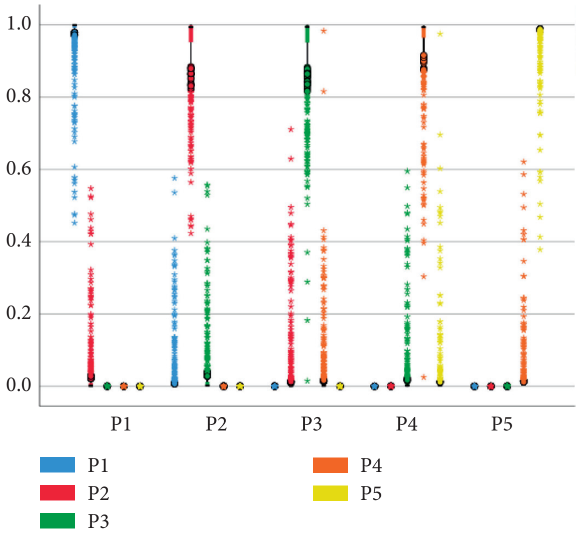


Figure 5: Predicting pseudo-probability of task price interval based on equal frequency division.

Comparison of price prediction accuracy effects of machine learning methods: the model is compared with the decision tree and KNN ($K = 3$) classification prediction model. The results are shown in Figure 3. According to the prediction accuracy of five price intervals, the DNN-based perceptual task price prediction model is the best, and the prediction accuracy tends to be between 82.7% and 93.3%, while the prediction accuracy of decision tree and 3-NN is 80.6%–92.9% and 79.4%–91.3%, respectively.

In addition, the methods for exploring the law of perceived task price are compared. The experimental results are shown in Figures 4–6 by TSCA, equal frequency division, and random division. Among them, the abscissa is the price interval generated by TSCA, equal frequency division, and random division, and the ordinate is the predicted probability. It can be seen from the graph that the use of TCSA to explore the perceived task pricing law and divide the price range according to the law is the best. The probability value of each sample case output belongs to its correct price range is roughly higher than 0.8, and the probability value of other incorrect price ranges is roughly less than 0.2, which can effectively divide the perceived task price range according to the factors affecting the price change.

The probability of task price interval prediction based on equal frequency division is shown in the figure. The predicted probability of outputting the correct price interval is approximately higher than 0.5, and the probability of outputting other incorrect price intervals is approximately less than 0.5, which can basically realize the price interval of distinguishing perception tasks, but the effect is significantly lower than that of TSCA, indicating that the influence of factors on the price of perception tasks needs to be considered.

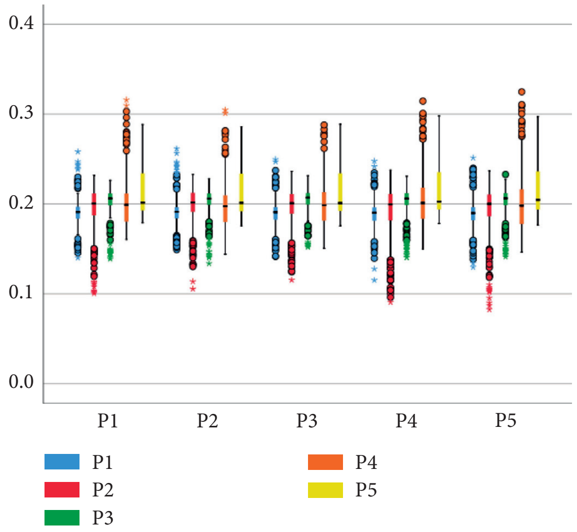


Figure 6: Predicting pseudo-probability of task price interval based on random division.

The probability of task price interval prediction based on random partition is shown in Figure 6. The probability values of the output correct price interval and other incorrect price intervals are roughly between 0.1 and 0.3, which cannot analyze the pricing law of sensing tasks.

Through the experimental comparison of machine learning method and two dimensions before and after clustering, the results show that the method can effectively analyze the pricing law of sensing task according to the factors that affect the price change and divide the price range according to the law, and the model has high prediction accuracy.

5. Conclusions and Future Work

A task price prediction method based on clustering and DNN is proposed. Firstly, the task pricing standard dataset is constructed to improve the accuracy of analyzing the task pricing law and reduce the time consumption of the analysis law. Secondly, the task pricing law is explored by TSCA, and the task price is naturally grouped and classified according to the factors affecting the task price change, so as to achieve the purpose of dividing the price range according to the pricing law. Finally, the price interval prediction model based on DNN is realized. The experimental results show that the prediction accuracy of the pricing mechanism is higher than that of the classification prediction methods such as decision tree and KNN, and the analysis results of the pricing law are significantly better than those of the frequency division method and the random division method. In the future, in the face of a variety of scenarios and tasks, in view of the complex and diverse characteristics of crowdsensing tasks, the pricing prediction analysis of tasks will still be a topic worthy of further research.

Data Availability

The experimental data used in this article were obtained from the public dataset of Didi Travel.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors thank the National Natural Science Foundation of China (grant no. 41761086) and the Natural Science Foundation of Inner Mongolia Autonomous Region (2019MS06030) for the support. The authors also thank the Inner Mongolia Key Laboratory of Wireless Networking and Mobile Computing and the Self-Topic/Open Project of Engineering Research Center of Ecological Big Data, Ministry of Education.

References

[1] H. Psaier, F. Skopik, D. Schall, and S. Dustdar, “Resource and agreement management in dynamic crowdcomputing environments,” in *Proceedings of the 2011 IEEE 15th International Enterprise Distributed Object Computing Conference*, pp. 193–202, IEEE, Helsinki, Finland, September 2011.

[2] B. Guo, Z. Wang, Z. Yu et al., “Mobile crowd sensing and computing,” *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–31, 2015.

[3] M. Srivastava, T. Abdelzaher, and B. Szymanski, “Human-centric sensing,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, vol. 370, no. 1958, pp. 176–197, 2012.

[4] P. Haddawy, L. Frommberger, and T. Kauppinen, “Situation awareness in crowdsensing for disease surveillance in crisis situations,” in *Proceedings of the Seventh International Conference on Information and Communications Technologies and Development (ICTD 2015)*, vol. 1–5, Singapore, May 2015.

[5] A Oscar, C Carlos, C Juan-Carlos, and P. Manzoni, “Crowdsensing in smart cities: overview, platforms, and environment sensing issues,” *Sensors*, vol. 18, no. 2, pp. 460–488, 2018.

[6] M. A. Rahman and M. S. Hossain, “A location-based mobile crowdsensing framework supporting a massive ad hoc social network environment,” *IEEE Communications Magazine*, vol. 55, no. 3, pp. 76–85, 2017.

[7] F. Saremi, O. Fatemieh, H. Ahmadi et al., “Experiences with greengps—fuel-efficient navigation using participatory sensing,” *IEEE Transactions on Mobile Computing, IEEE*, vol. 15, no. 3, pp. 672–689, 2015.

[8] S. Chung and I. Rhee, “vTrack: virtual trackpad interface using mm-level sound source localization for mobile interaction,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pp. 41–44, 2016.

[9] K. Han, H. Huang, and J. Luo, “Quality-aware pricing for mobile crowdsensing,” *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1728–1741, 2018.

[10] J. Sun and H. Ma, “Collection-behavior based multi-parameter posted pricing mechanism for crowd sensing,” in *Proceedings of the 2014 IEEE International Conference on Communications (ICC)*, pp. 227–232, IEEE, Sydney, Australia, 10-14 June 2014.

[11] S. Wu, X. Gao, F. Wu, and G. Chen, “A constant-factor approximation for bounded task allocation problem in crowdsourcing,” in *Proceedings of the GLOBECOM 2017-2017*

IEEE Global Communications Conference, vol. 1–6, 4-8 December 2017.

[12] Z. Duan, W. Li, and Z. Cai, "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems," in *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 635–644, IEEE, Atlanta, GA, USA, 5-8 June 2017.

[13] D. Zhao, H. Ma, Q. Li, and S. Tang, "A unified delay analysis framework for opportunistic data collection," *Wireless Networks*, vol. 24, no. 4, pp. 1313–1325, 2018.

[14] L Wang, D Zhang, H Xiong, J. P. Gibson, C. Chen, and B. Xie, "EcoSense: minimize participants' total 3G data cost in mobile crowdsensing using opportunistic relays," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 965–978, 2016.

[15] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: crowdsourcing with s," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1732–1744, 2016.

[16] J. Lin, M. Li, D. Yang, and G. Xue, "Sybil-proof incentive mechanisms for crowdsensing," in *Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, vol. 1–9, 16-19 April 2018.

[17] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[18] Z. Li, Y. Li, and W. Lu, "Crowdsourcing logistics pricing optimization model based on DBSCAN clustering algorithm," *IEEE Access*, vol. 8, no. 99, pp. 92615–92626, 2020.

[19] S. Guan, "Analysis of optimal pricing model of crowdsourcing platform based on cluster and proportional sharing," in *Proceedings of the 2018 6th International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 99–103, IEEE, Basel, Switzerland, 27-29 Aug. 2018.

[20] X. Yingxin, W. Bofeng, and L. I. Yi, "Task pricing problem based on the multivariate statistical methods," *Journal of Fujian Computer*, vol. 35, no. 5, pp. 33–36, 2019.

[21] K. Han, H. Huang, and J. Luo, "Posted pricing for robust crowdsensing," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 261–270, Association for Computing Machinery, New York, NY, USA, 2016.

[22] J. Shao and S. U. Qifang, "Research on the pricing strategy of crowdsourcing based on bivariate pricing model," *Journal of Taizhou University*, vol. 40, no. 3, pp. 7–14, 2018.

[23] F. Yuting, L. Chen, Y. Ying, and L. Xiangfeng, "Application of supply and demand model in crowdsourcing platform pricing," *Journal of Quantitative Economics*, vol. 35, no. 1, pp. 74–76, 2018.

[24] T. Sohail, *Developing Market Sentiment Indicators for Commodity price Forecasting Using Machine learning*, University of Manitoba, Winnipeg, Canada, 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5553874, 10 pages
https://doi.org/10.1155/2021/5553874

Research Article

Mine Consortium Blockchain: The Application Research of Coal Mine Safety Production Based on Blockchain

Zilin Qiang,[1] Yingsen Wang,[2] Kai Song,[2] and Zijuan Zhao[2]

[1]*School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK*
[2]*College of Information and Computer, Taiyuan University of Technology, Taiyuan 030024, China*

Correspondence should be addressed to Zijuan Zhao; zhaozijuan0064@link.tyut.edu.cn

Received 8 January 2021; Accepted 8 May 2021; Published 17 June 2021

Academic Editor: Zhe-Li Liu

To solve the problem that the safety data in the process of coal mine production are easy to be maliciously tampered with and deleted, a mine consortium blockchain data security monitoring system is proposed. The coal mine consortium blockchain includes supervision department, builds favourable centralized and decentralized production mode, and improves PBFT (Practical Byzantine Fault Tolerance) consensus mechanism to implement practical coal mine safety production. The evaluation shows that the architecture we proposed is more appropriate and efficient for the mine Internet of Things than the traditional blockchain architecture. The Hyperledger Fabric platform is used to build the mine consortium blockchain system to achieve the sensor data reliability, node consensus, safe operation automation management, and major equipment traceability.

1. Introduction

China's demand for coal energy is increasing year by year, and the scale of coal mining is gradually expanding. However, in the actual mining process, there are many mining problems, which may cause serious safety accidents to occur [1]. During the production process, in order to avoid the alarm and continue to produce, artificial tampering and deletion of sensor data result in the safety monitoring system failing to truly reflect the environmental changes in coal mines [2]; operators who do not operate strictly in accordance with regulations and abnormal equipment operation status are the biggest obstacles restricting coal mine production. However, the existing safety monitoring systems, personnel positioning systems, and major equipment systems cannot monitor the occurrence of these unsafe phenomena [3]. Thus, there is an urgent need to develop a platform that can prevent coal mines from tampering with sensor data and discover unsafe human behaviors and unsafe conditions in the production process in a timely manner [4], so as to ensure safe production in coal mines and provide more convenient supervision methods for supervision department.

Blockchain technology is a decentralized distributed ledger technology [5, 6]. Its characteristics of "nonforgery," "full trace," "traceability," "open and transparent," and "collective maintenance" are very consistent with the safety requirements of coal mine production. Literature [7] put forward the idea of applying blockchain technology to safe production information construction, so as to maximize resource sharing and provide convenient conditions for corporate supervision and government supervision. Literature [8] analyzed the construction method of agricultural product quality safety and efficient traceability system based on alliance blockchain. Literature [9] constructed a college safety education examination system based on blockchain technology, greatly enhancing the reliability of test-related data. The above studies have put forward the application concept of blockchain technology in information security from different scenarios, but the application of blockchain technology is still in the stage of theoretical exploration. At present, there is no theoretical research and model construction in the field of coal mine safety production with blockchain technology.

Therefore, a coal mine safety production system mode based on consortium chain technology is proposed in this paper. First, the node architecture of the consortium chain integrated with the safety production model is constructed and the business process is analyzed. Then, specific analysis from the aspects of data reliability, safety production automation, traceability of major equipment, and consensus mechanism to provide a reference for the current theoretical research and actual construction of coal mine safety blockchain are conducted.

2. Consortium Blockchain Model Design

2.1. Application Model Architecture. In the model, the blockchain technology is used to integrate smart contract, consensus mechanism, and distributed database. Meanwhile, the supervision departments are incorporated into the consortium chain system to realize nontampering of data, intelligent production process, and traceability of equipment information. The application model architecture of the consortium chain is shown in Figure 1. The issuance of certificates is completed by Certification Authority (CA) [10] for the realization of identity authentication and authorization functions in the blockchain. Sensor nodes are responsible for recording coal mine data, uploading information to the chain through a consensus mechanism, and recording relevant data in the decentralized distributed ledger of the blockchain. After being authorized, the supervision institution completes the querying and tracing of the coal mine data in the ledger by calling smart contracts. The three are connected through this model to form an organic whole. For unsafe factors such as unsafe behaviors of people and unsafe condition of things, timely supervision and effective accountability can be achieved.

The model specifically includes a data credibility module, a safe operation automation module, and a major equipment traceability module, as shown in Figure 2. The interconnection of mine sensor information is realized, and all the sensor information in the underground is encapsulated into a block [11]. Conduct a consensus review of the information before being chained, and the information cannot be tampered with after being chained, solving the problem of sensor data being tampered with in coal mine [12]; establish a compliant process operation contract in coal mine safety production, add user identity verification, build an equipment information record chain independent of monitoring data, realize equipment information monitoring and traceability, and solve the problem of illegal operation in the safety production process. It can be seen that the system software can be basically divided into three submodules to solve the unsafe behavior of people, the unsafe condition of things, and the unsafe factors of the environment.

2.2. Node Architecture and Business Process. The model node architecture and workflow are shown in Figure 3, including modules such as CA, sorting service, and endorsement node [13].

2.2.1. CA. In this model, the internal authority of the coal mine safety production company acts as a CA (digital certificate authority) node, which relies on the Membership Service Provider (MSP) in the consortium chain system, and it is responsible for receiving the registration application of the user node and returning the registration password for login, so that users can obtain identity certificate. After successful application, the CA will provide the private key to the user node in the form of a password. Finally, users store the certificate signing request and private key in a secure location on the server or local drive. The user's identity will be verified in all operations in the blockchain network to confirm whether it belongs to a sensor node, a supervisory node, or other nodes.

2.2.2. Sorting Service. The sorting service is responsible for sorting the data packets returned by each sensor. The sorting service arranges the transaction information into a certain order based on the rules according to the timestamp and packs it to facilitate data transmission [14] endorsement strategy and endorsement nodes.

The client initiates a sensor data package proposal. The consortium chain system does not automatically send data. Instead, the client specifies which nodes the block is sent to for endorsement verification; that is, the client specifies the endorsement policy [15]. The endorsement strategy defines the necessary combination conditions for the specific node that executes the proposal in the channel (i.e., the private chain) and its response result.

The endorsement node mainly performs verification, simulation execution, and endorsement of the proposal. After each endorsement node receives the block, it will verify the data in it. If the proposal does not meet the endorsement policy, it will not take effect locally.

2.2.3. Node Confirmation. Node confirmation is mainly responsible for checking the legality of transactions and updating and maintaining the state of the blockchain and ledger. Before the data packet is recorded in the ledger, it must be confirmed by the node confirmation, and only the confirmed data can be recorded in the blockchain ledger.

2.2.4. World State. Various information is recorded in the world state, such as sensor node data, smart contract bytecode, data customized by each smart contract, and chain configuration parameters [16]. The world state represents the current state, that is, the current value of each state data recorded.

2.2.5. Client and SDK. Through the client and SDK interface, users can specify endorsement strategies to endorsement nodes, send data information and device information to the sorting service, and participate in transaction verification.

3. Sensor Data Reliability

To realize the interconnection of mine sensor information and the decentralized storage of sensor data, all sensor

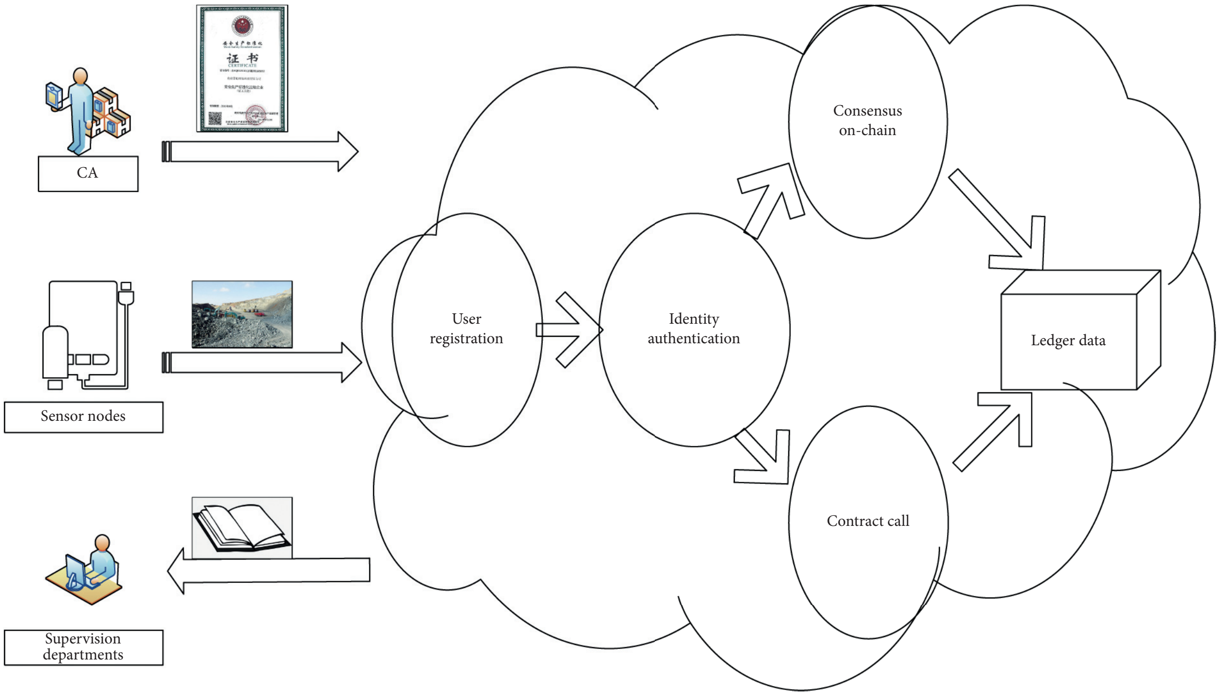


Figure 1: Model architecture of coal mine production consortium chain.

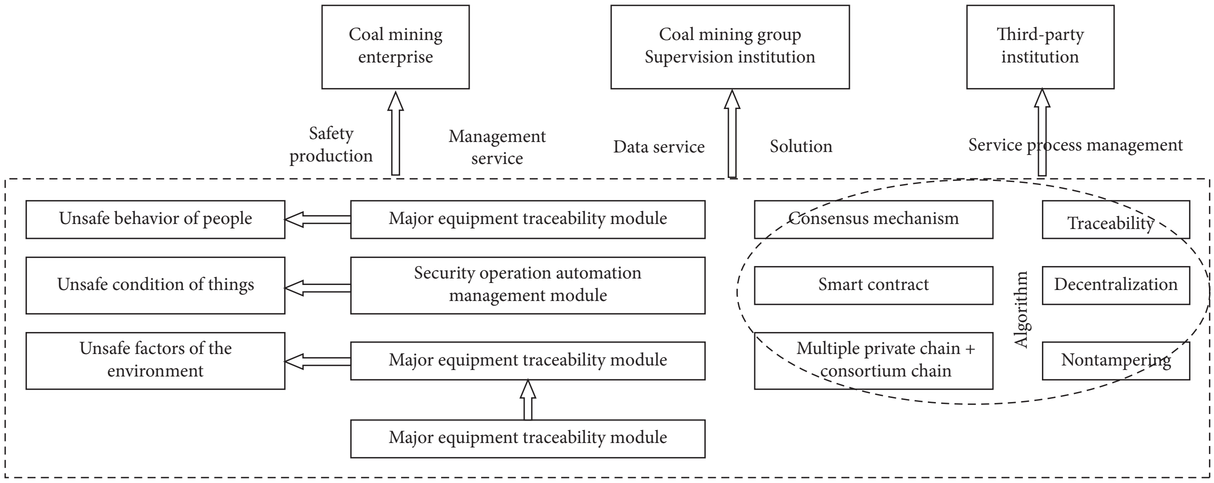

Coal mining enterprise
Coal mining group Supervision institution
Third-party institution
Safety production
Management service
Data service
Solution
Service process management
Unsafe behavior of people
Unsafe condition of things
Unsafe factors of the environment
Major equipment traceability module
Security operation automation management module
Major equipment traceability module
Major equipment traceability module
Consensus mechanism
Smart contract
Multiple private chain + consortium chain
Algorithm
Traceability
Decentralization
Nontampering

Figure 2: Submodule system structure.

information in the mine is encapsulated by the model into blocks. Consensus review is conducted on the information before the cochain, and the information cannot be tampered with after the cochain to ensure that the sensor data during the collection, transmission, and storage is true and reliable.

The completed module can not only provide closed-loop data services of safety management but also be embedded in the coal mine information safety monitoring system software system based on blockchain technology to provide data services. When the module is embedded in the system, the existing security monitoring system is improved, and a reliable system for sensor data is built. The system can set the user's functional authority and data authority according to the module characteristics and preset solutions suitable for different users in various abnormal scenarios. When the module detects that the sensor data have been tampered with, the system dynamically pushes abnormal information and response emergency solutions to users at all levels combined with the data of the existing intelligent perception analysis system of artificial intelligence, Internet of Things, and big data and provides users at all levels with a feedback channel for abnormal information or solution error information. After verification, the effective feedback information will be added to the optimization algorithm of the system. The sensor data reliability includes the following specific functional modules. The structure is shown in Figure 4.

Account management is to record and change the members in the consortium chain network. Part of the

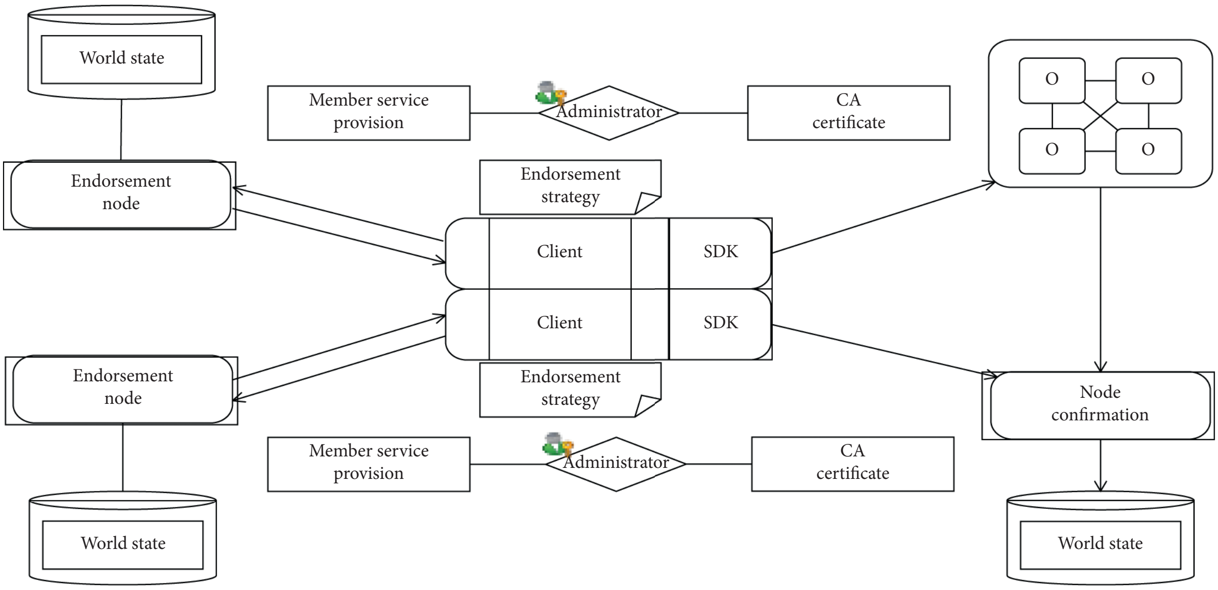


Figure 3: Node architecture and workflow.

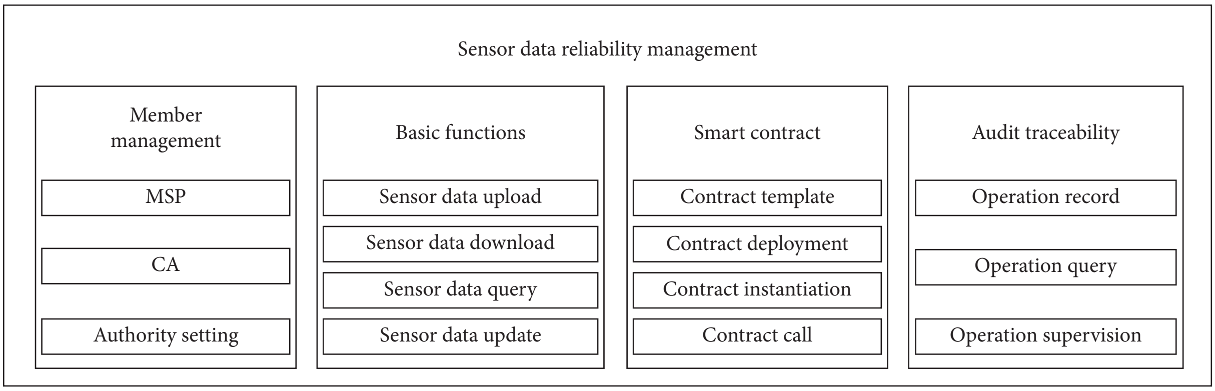


Figure 4: Sensor data reliability.

functions of this module depend on the MSP of the consortium chain, and they are responsible for maintaining the IDs of all nodes in the system and issuing node identity certificates. The basic function module realizes the main functions of the safe production consortium chain, namely, the upload, download, query, and update of sensor data. The smart contract module is responsible for maintaining all chain codes in the system and realizes operations such as storage, compilation, update, and installation of chain codes. For the safety production consortium chain proposed in this paper, this module provides support for basic functions and audit traceability functions. The traceability module is responsible for packaging operation logs and device information into blocks and connecting them into chains while performing operations on the sensor data chain. Similar to the operation of the data link, the query of the operation log is to retrieve all the information in the behavior chain and return the specified log information to realize the postaudit function of the sensitive operation of the sensor data.

4. Safety Operation Automation Management

Smart contract is a computer protocol that can automatically execute the contract, and its characteristic is to allow trusted transactions without a third party [17]. However, at present, there is no complete template for smart contracts, and there are still higher risks and vulnerabilities. The DAO in Ethereum is a typical attack by hackers using smart contract vulnerabilities [18].

The multiple private chain architecture deployed with smart contracts is shown in Figure 5. This model stores data content with different permissions separately into chains and records the hash value returned by the sensor data transmission system in the form of transactions. The node security level is divided into three levels, named “confidential,” “secret,” and “public” [19]. According to the levels and functions of various departments such as coal mine production and supervision, different permissions are assigned to correspond to three private chains. The identity information issued by the MSP of the blockchain is divided into three identities: administrator, authorized member, and ordinary member. General confidential private chain data can only be operated by administrators, secret private chain data can be operated by administrators and authorized members, and public data can be operated by all members including ordinary members. In addition, this model packs operation logs and equipment information on the chain in the form of transactions in the blockchain network to facilitate postaudit and realize the traceability of data sources.

Smart contracts are deployed on three different private chains to ensure data privacy and isolation. Each private chain can deploy multiple smart contracts, and the contracts can call each other. Data can be queried across private chains, but data cannot be changed. Each smart contract specifies the production process of its own equipment or product. If the department refuses to implement the specified production process for saving costs, the testing equipment will record the operating data and automatically deduct funds from the relevant departments and conduct other types of processing based on the smart contract. Inside the smart contract, specific processes and logical relationships are stipulated, such as processes A and B or joint confirmation of organizations A and B.

5. Major Equipment Traceability

For the traditional traceability system, the regulations of traceability system are formulated by relevant institutions such as government and enterprises. After the regulations are formulated, all production links are required to comply with the regulations. A new traceability scheme based on blockchain technology is adopted in this model. The scheme aims to establish a decentralized product traceability system maintained by nontrusted participants. The traceability scheme based on blockchain technology is adopted to store traceability information in the blockchain, and the decentralized characteristics of the blockchain are used to realize a decentralized traceability system [20]. A multiple private chain plus consortium chain architecture with high throughput and high security is used, and the Merkel tree is used to correlate the information stored in the two. The Merkel tree structure is shown in Figure 6.

In the coal mine production environment, coal mine data are detected by sensors in real time. To ensure data traceability of the blockchain system, once the data are entered, they cannot be tampered with and cannot be deleted. Therefore, the amount of data will continue to increase with time, which may cause problems such as excessive data volume. This problem can be effectively solved by the Merkle Tree structure in the safe production blockchain system.

Suppose that there is a transaction with a value of Hash4 in the block to be verified, it is only necessary to know Hash3, Hash12, and Hash5678. The hash of Hash34 and Hash1234 and the root node are calculated. If the finally calculated root node hash is consistent with that recorded in the block header, it means that the transaction exists in the block. The key to Merkel tree verifying transactions lies in two points: (1) Hash is calculated by the Merkel tree layer by layer from bottom to top, so as long as the hash value of another adjacent node is known, it can be calculated up to the root node; (2) the hash value of the root node can be accurately used as the unique digest of a group of transactions [21]. Based on these two points, the existence of a transaction can be verified.

The greatest advantage of the Merkel tree is that each transaction can be deleted directly and individually, and the hash value of this transaction is just kept. In this way, for the entire block, the cryptographic security and integrity of the block are not changed, but the amount of data can be greatly reduced (the hash value has 32 bytes, and a transaction generally takes more than 400 bytes). If there is only one transaction in a block that has no subsequent transactions, deleting all other transactions will greatly reduce the amount of data in the entire block. Hence, in the safe production blockchain, there is usually no need to worry about the problem of excessive data caused by the increasing amount of data with the Merkel tree.

6. Consensus Mechanism

Consensus mechanism refers to the process of reaching a unified agreement on the state of the network in a decentralized manner, which helps to verify that information is added to the ledger, ensuring that only real transactions are recorded on the blockchain [22]. It means that a certain node modifies a certain data in the block during storage, but other nodes have no consensus on this block, and the modification operation is invalid.

There are currently three mainstream algorithms:

(1) Proof of Work (PoW): the principle of the PoW mechanism is that the more work is done, the greater the benefit is. PoW can be divided into two phases, solution and verification. The solution phase requires a lot of complex calculations to compare computing power to obtain a mathematical solution; in the verification phase, the correctness of the mathematical solutions of other nodes can be verified through simple calculations. PoW requires a lot of complex calculations to ensure the correctness and consistency of the blockchain network [23]. Therefore, it is necessary to consume a lot of energy such as electricity, and it takes a long time to reach a consensus, so the PoW mechanism is inefficient.

(2) Proof of Stake: PoS is also a kind of consensus proof, which is similar to equity certificates and voting systems, so it is also called "stake proof algorithm" [24]. It is completed relying on tokens, and the person holding the most tokens will announce the final information, which is not suitable for most blockchain application scenarios.

(3) Practical Byzantine Fault Tolerance: PBFT is also a common consensus proof. It is different from the previous two mechanisms. PBFT is based on calculations and there is no token reward. Everyone on the chain participates in voting. The right to publicize information can be got when the objection is less than (N−1)/3 nodes.

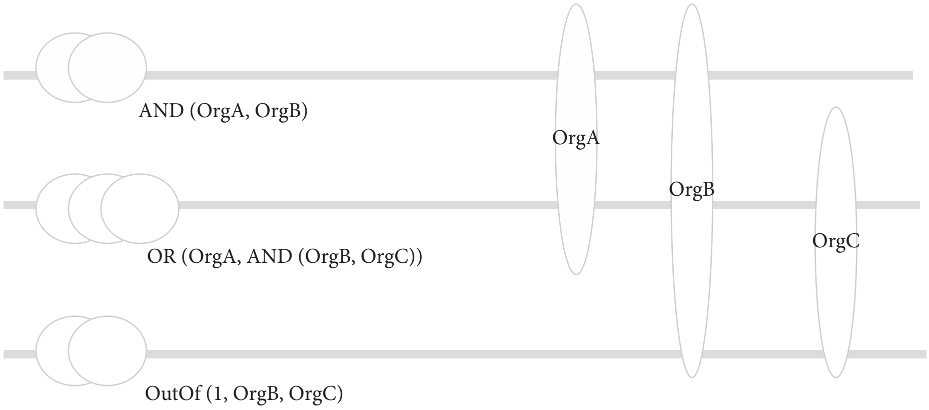


Figure 5: Smart contract deployment architecture.

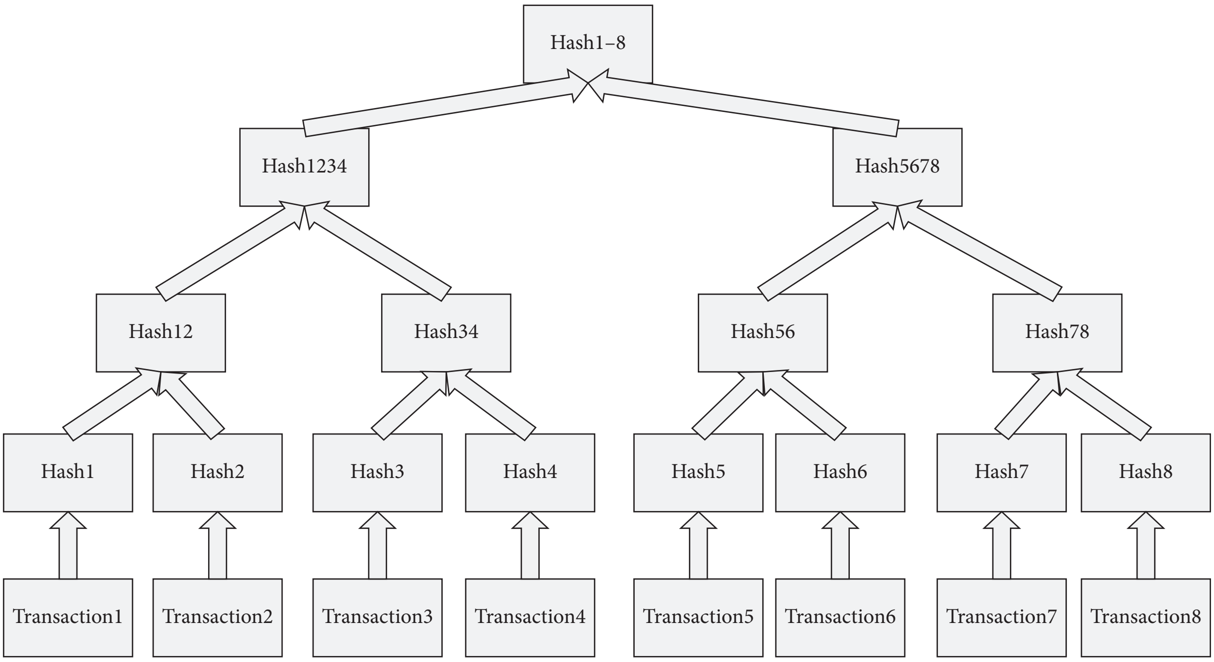


Figure 6: Merkle tree.

The PBFT is adopted in the security management model, which can regularly evaluate the availability of nodes and dynamically add or delete authentication nodes to ensure overall operating efficiency.

In the PBFT consensus process, nodes are divided into master nodes and slave nodes. The master nodes are mainly composed of coal mine safety management departments, and the slave nodes are composed of sensor nodes. The consensus process without node failure is shown in Figure 7. The master node is responsible for data packaging, and the slave node is responsible for data verification. Both types of nodes have the function of interacting with the client.

First, the client sends a request to any node (from node 1 in the figure), and the node broadcasts the data request sent by the client to the entire network. After collecting multiple requests, the master node sorts them into the list, produces preprepared certificates, and broadcasts the preprepared certificates to the entire network. After it is received by the slave node, it verifies the certificate and replies to the master node for information approval. After the master node receives the approval information of all the slave nodes, it packs the approval information and sends it to all the slave nodes, and the slave node verifies the correctness of the approval information of other slave nodes. When all nodes indicate that they receive block information from other slave nodes, other nodes will package the data information on the cochain.

7. Simulation and Analysis

This section tests the performance of the coal mine consortium blockchain and compares it with traditional PBFT, two classic public blockchain platforms, Bitcoin, and Ethereum to highlight the advantages of our model. Finally, the Hyperledger Fabric platform is used to construct the coal mine consortium blockchain system. We focus on the

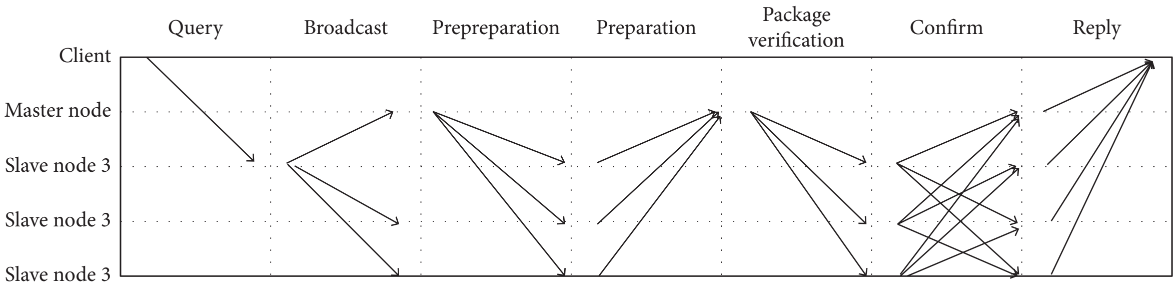


Figure 7: PBFT consensus process.

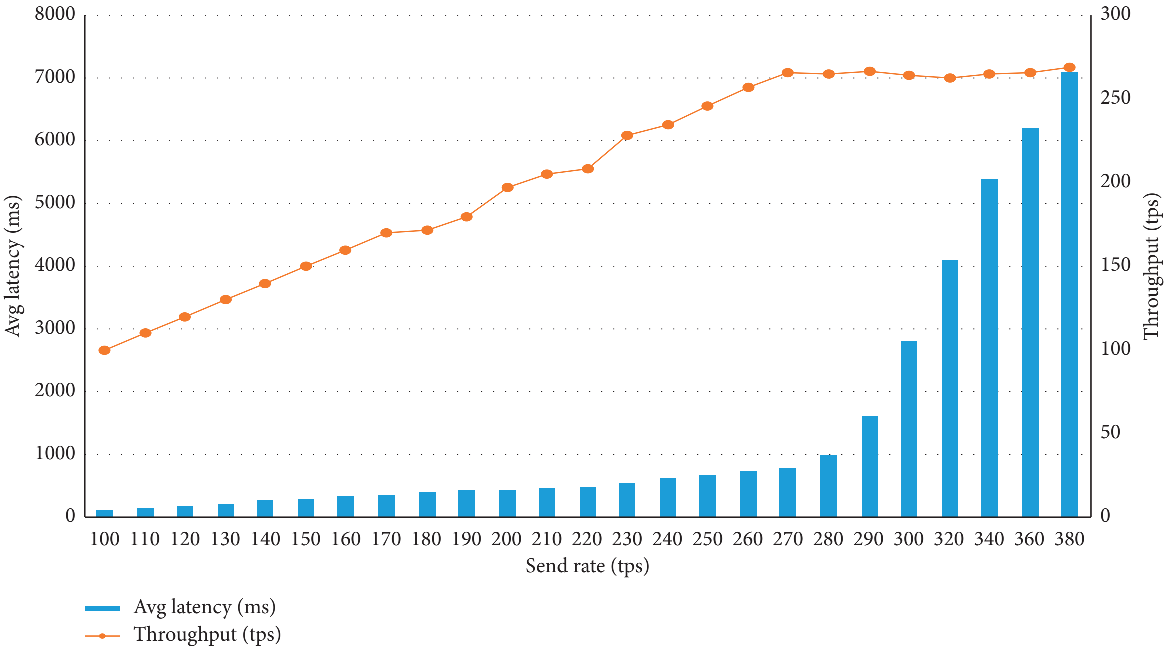


Figure 8: The performance of throughput and latency of coal mine consortium blockchain.

analysis of the indicators of throughput and latency. Throughput is used to describe the number of data requests that can be processed per second in the coal mine consortium blockchain and is an important indicator to measure the performance of the consensus algorithm. The number of transactions packaged per unit time is expressed as TPS (transactions per second) in blockchain: Transactions/Δt. Latency refers to the time from issuing a data request from the master node to the response of the entire blockchain platform.

Hyperledger Caliper is used to perform the pressure test, setting up 100 nodes to continuously send transaction requests to the coal mine consortium blockchain system and observe its throughput and latency by increasing send rates.

Figure 8 shows the performance of the proposed coal mine consortium blockchain model in throughput and latency. Throughput rises as send rate increases, when send rate reaches around 270 tps, it no longer keeps increasing, but maintains a steady state. When send rate was lower than 27 tps, the latency increases slowly and remains below 1500 ms at a relatively low level. When send rate is greater than 270 tps, the latency increases rapidly.

In order to evaluate the effectiveness of the proposed coal mine consortium blockchain model, we compared the throughput of the consortium blockchain with traditional PBFT and two classic blockchain model: Bitcoin and Ethereum. The result shows that the coal mine consortium blockchain model has obvious advantages in throughput. Bitcoin uses the POW consensus mechanism, which requires a lot of computing power and only produces one block every 10 minutes on average. Ethereum uses a POW and POS consensus mechanism; it takes about 10 seconds per block. The data required by the comparison experiment is from two different public data sets [25, 26], which collected part of the data generated by the two public blockchains in the real transaction. Compared with the traditional PBFT, the improved PBFT increases the verification step, but the communication time is greatly reduced, which not only increases the security in the process of block transmitting but also reduces the communication traffic as shown in (Figure 9).

Finally, we use Hyperledger Fabric platform to build the coal mine consortium blockchain system, so as to implement the coal mine output transaction, query, and other operations. Figure 10 shows the mineral trading of the coal mine

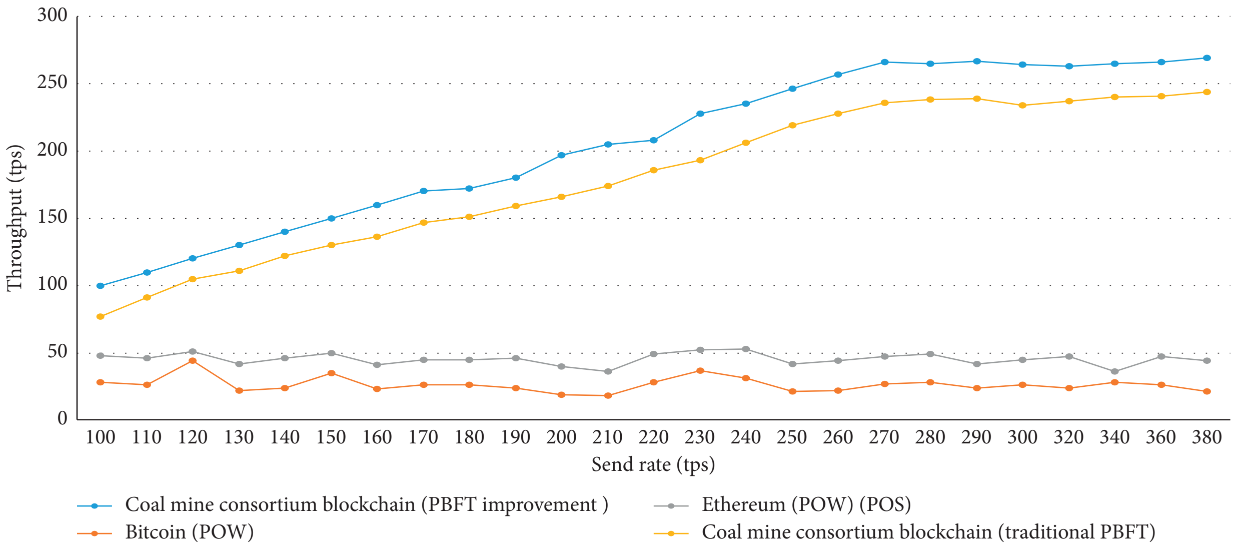
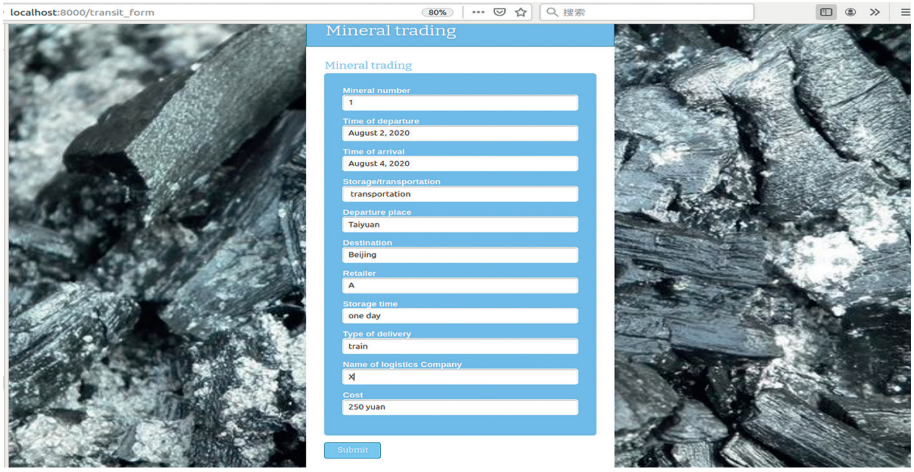


Figure 9: Throughput comparison.



Mineral trading

Mineral trading

Mineral number: 1

Time of departure: August 2, 2020

Time of arrival: August 4, 2020

Storage/transportation: transportation

Departure place: Taiyuan

Destination: Beijing

Retailer: A

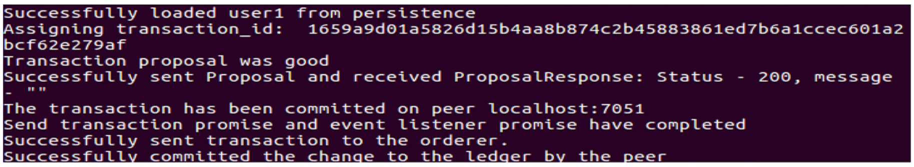
Storage time: one day

Type of delivery: train

Name of logistics Company: X

Cost: 250 yuan

Figure 10: The mineral trading of consortium blockchain.



```
Successfully loaded user1 from persistence
Assigning transaction_id:  1659a9d01a5826d15b4aa8b874c2b45883861ed7b6a1ccec601a2
bcf62e279af
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message
- ""
The transaction has been committed on peer localhost:7051
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer
```

Figure 11: Performance of the underlying of the coal mine consortium blockchain.

Table 1: The test result of the Hyperledger Caliper.

Test	Name	Send rate	Avg latency (s)	Throughput (tps)
1	Transaction	100	0.11	100
2	Transaction	110	0.14	110
3	Transaction	120	0.18	120
4	Transaction	130	0.21	130
5	Transaction	140	0.26	140
6	Transaction	150	290	150
7	Transaction	160	0.29	160

Table 1: Continued.

Test	Name	Send rate	Avg latency (s)	Throughput (tps)
8	Transaction	170	0.36	170
9	Transaction	180	0.39	172
10	Transaction	190	0.43	180
11	Transaction	200	0.44	197
12	Transaction	210	0.44	205
13	Transaction	220	0.49	208
14	Transaction	230	0.54	228
15	Transaction	240	0.62	235
16	Transaction	250	0.68	246
17	Transaction	260	0.73	257
18	Transaction	270	0.78	266
19	Transaction	280	0.99	265
20	Transaction	290	1.6	267
21	Transaction	300	2.8	264
22	Transaction	320	4.1	263
23	Transaction	340	5.4	265
24	Transaction	360	6.2	266
25	Transaction	380	7.1	269

consortium blockchain. Figure 11 shows the performance of the underlying blockchain after the mineral trading, including the hash value of this transaction.

8. Summary

The decentralized, intelligent, and tamper-resistant characteristics of blockchain technology are in line with the requirements of coal mine safety production, and blockchain technology has broad development prospects in the field of coal mine safety production. Our method enhances the safety of coal mine safety production using blockchain. However, we deployed a small number of nodes to test the performance of our platform. And our model assumes that most nodes are normal. So, in a real large-scale deployment, message consistency and accuracy may not be guaranteed. For future work, we will improve our consensus algorithm flow, especially the detailed view-change. And we will deploy our platform in a public blockchain (e.g., Ethereum). Although blockchain technology has different degrees of application and research in various industries, it is still in its infancy, and there are deficiencies such as low efficiency and waste of resources. As a result, blockchain technology has not yet been widely applied.

In cooperation with Jingying Technology in Taiyuan, this paper proposes a coal mine safety production consortium chain model combined with blockchain technology and safety requirements in the field of coal mine production. It incorporates supervision departments into the system to form a good and supervisable business model where centralization and decentralization coexist. The module structure, node structure, and business process of the model are analyzed and corresponding countermeasures are proposed aiming at the problems of data reliability, safety automation management, and major equipment traceability to promote the research and development of blockchain technology in the field of coal mine safety production.

Appendix

We use the Hyperledger Caliper in Section 6 to evaluate the performance of the coal mine consortium blockchain. Transaction requests are continuously sent to the blockchain model over an HTTP port at a specific rate to monitor the related performance of the consortium blockchain. The test result is shown in Table 1.

Data Availability

Previously reported data were used to support this study. The datasets are cited at relevant places within the text as references [19, 20]. And the pressure test of our model is shown in Appendix in this paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Zong, W. Wu, and F. Liu, “Coal mine production safety measures and accident statistics in China,” *Coal Technology*, vol. 39, no. 1, pp. 205–207, 2020.

[2] C. Tunc and S. Hariri, “CLaaS: cybersecurity lab as a service,” *Journal of Internet Services and Information Security*, vol. 5, no. 4, p. 19, 2015.

[3] I. Kotenko, I. Saenko, and A. Kushnerevich, “Parallel big data processing system for security monitoring in internet of things networks,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 8, no. 4, pp. 60–74, 2017.

[4] B. Bordel, R. Alcarria, M. Ángel Manso, and A. Jara, “Building enhanced environmental traceability solutions: from thing-to-thing communications to generalized cyber-physical systems,” *Journal of Internet Services and Information Security*, vol. 7, no. 3, pp. 17–33, 2019.

[5] J. Li, Y. Huang, Y. Wei et al., “Searchable symmetric encryption with forward search privacy,” *IEEE Transactions on Dependable & Secure Computing*, vol. 18, no. 1, pp. 460–474, 2019.

[6] Q. Shao, C. Jin, Z. Zhang et al., "Blockchain: architecture and research progress," *Chinese Journal of Computers*, vol. 41, no. 5, pp. 969–988, 2018.

[7] J. Li, M. Wang, J. Zhang et al., "Application of block chain technology in the construction of safety production informatization," *An Quan*, vol. 41, no. 2, pp. 88–93, 2020.

[8] J. Luo, "Analysis of an efficient traceability system for agricultural product quality and safety based on alliance blockchain," *Computer Knowledge and Technology*, vol. 16, no. 6, pp. 272-273, 2020.

[9] C. Fan, Y. Li, G. Zhang et al., "Study on the blockchain based safety education examination system for colleges and universities," *Guangdong Chemical Industry*, vol. 46, no. 14, pp. 211-212, 2019.

[10] J. Li, H. Yan, Y. Zhang et al., "Efficient identity-based provable multi-copy data possession in multi-cloud storage," *IEEE Transactions on Cloud Computing*, vol. 24, no. 12, 2019.

[11] C. Gritti, M. Önen, R. Molva, S. Willy, and P. Thomos, "Device identification and personal data attestation in networks," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 9, no. 4, pp. 1–25, 2018.

[12] B. Todd and K. Andersson, "Network security of internet services: eliminate DDoS reflection amplification attacks," *Journal of Internet Services and Information Security*, vol. 5, no. 3, p. 22, 2015.

[13] J. Zhang, Y. Qiu, and H. Jin, "Research on the SDK national secret transformation plan of the fabric platform of hyperledger," *Network Security Technology & Application*, vol. 231, no. 3, pp. 37–39, 2020.

[14] Networks—Peer-to-Peer Networking, "Researchers from beijing jiaotong university report details of new studies and findings in the area of peer-to-peer networking (performance analysis of hyperledger fabric platform: a hierarchical model approach)," *News of Science*, vol. 13, no. 3, 2020.

[15] J. Long, "Design of blockchain system in BDCP using hyperledger fabric," in *Proceedings of the 2019 World Symposium on Software Engineering (WSSE 2019)*, pp. 79–83, Wuhan, China, September 2019.

[16] R. Campbell, *Transitioning to a Hyperledger Fabric Quantum-Resistant Classical Hybrid Public Key Infrastructure*, Capitol Technology University, Laurel, ML, USA, 2019.

[17] Z. Zheng, S. Xie, H.-N. Dai et al., "An overview on smart contracts: challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, 2020.

[18] H. Ren and Z. Xie, "The criminal risk of smart contracts in the blockchain 2.0 era and countermeasures—taking the DAO hacking incident as an example," *Crime and Reform Research*, no. 3, pp. 2–7, 2020.

[19] D. Wu, Y. Xiang, C. Wang et al., "Data protection technology for information systems based on blockchain," *Journal of Command and Control*, vol. 4, no. 3, 2018.

[20] C.-S. Shih, W.-Y. Hsieh, and C. L. Kao, "Traceability for vehicular network real-time messaging based on blockchain technology," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 10, no. 4, pp. 1–21, 2019.

[21] X. Weng, P. Zhang, W. Wang et al., "Remote attestation mechanism for platform integrity based on unbalanced-hash tree," *Journal of Computer Applications*, vol. 2, pp. 433–437, 2014.

[22] Y. Huang, B. Li, Z. Liu et al., "ThinORAM: towards practical oblivious data access in fog computing environment," *IEEE Transactions on Services Computing*, vol. 13, no. 4, 2019.

[23] D. Larimer, "Delegated proof-of-stake white paper," *IEEE Access*, vol. 7, 2014.

[24] D. Schwartz, N. Youngs, and A. Britto, "The ripple protocol consensus algorithm," 2015, https://ripple.com/files/ripple_consensus_whitepaper.pdf.

[25] "Bitcoin historical data," 2018, https://www.kaggle.com/mczielinski/bitcoin-historical-data.

[26] Ethereum, "Ethereum historical data," 2021, https://www.kaggle.com/kingburrito666/ethereum-historical-data.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5560185, 9 pages
https://doi.org/10.1155/2021/5560185

Research Article

Characterizing Network Anomaly Traffic with Euclidean Distance-Based Multiscale Fuzzy Entropy

Renjie Zhou,[1,2,3] **Xiao Wang**,[2] **Jingjing Yang**,[4] **Wei Zhang**,[2,3] **and Sanyuan Zhang**[1]

[1]*College of Computer Science and Technology, Zhejiang University, Hangzhou 310058, China*
[2]*College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China*
[3]*Key Laboratory of Complex Systems Modeling and Simulation of the Ministry of Education, Hangzhou Dianzi University, Hangzhou 310018, China*
[4]*Zhuoyue Honors College, Hangzhou Dianzi University, Hangzhou 310018, China*

Correspondence should be addressed to Renjie Zhou; renjie_zhou@163.com, Wei Zhang; magherozhw@hdu.edu.cn, and Sanyuan Zhang; syzhang@cs.zju.edu.cn

Received 19 February 2021; Accepted 21 May 2021; Published 16 June 2021

Academic Editor: Zhe-Li Liu

The prosperity of mobile networks and social networks brings revolutionary conveniences to our daily lives. However, due to the complexity and fragility of the network environment, network attacks are becoming more and more serious. Characterization of network traffic is commonly used to model and detect network anomalies and finally to raise the cybersecurity awareness capability of network administrators. As a tool to characterize system running status, entropy-based time-series complexity measurement methods such as Multiscale Entropy (MSE), Composite Multiscale Entropy (CMSE), and Fuzzy Approximate Entropy (FuzzyEn) have been widely used in anomaly detection. However, the existing methods calculate the distance between vectors solely using the two most different elements of the two vectors. Furthermore, the similarity of vectors is calculated using the Heaviside function, which has a problem of bouncing between 0 and 1. The Euclidean Distance-Based Multiscale Fuzzy Entropy (EDM-Fuzzy) algorithm was proposed to avoid the two disadvantages and to measure entropy values of system signals more precisely, accurately, and stably. In this paper, the EDM-Fuzzy is applied to analyze the characteristics of abnormal network traffic such as botnet network traffic and Distributed Denial of Service (DDoS) attack traffic. The experimental analysis shows that the EDM-Fuzzy entropy technology is able to characterize the differences between normal traffic and abnormal traffic. The EDM-Fuzzy entropy characteristics of ARP traffic discovered in this paper can be used to detect various types of network traffic anomalies including botnet and DDoS attacks.

1. Introduction

The prosperity of network technologies, such as mobile networks and social networks, brings revolutionary changes to our daily lives. However, due to the complexity and fragility of the network infrastructures, network anomalies and attacks frequently cause serious problems and significant loss to people. Researchers are studying various cybersecurity awareness technologies to help people understand the security status and trend of networks. Characterization of network anomaly traffic is one of the key technologies commonly used to model and detect network anomalies and then to raise the cybersecurity awareness capability of network administrators. The existing approaches of network anomaly detection can be mainly classified into six categories [1]: classification-based methods [2–4], clustering-based methods [5–9], statistical methods [10, 11], stochastic methods [12, 13], deep-learning-based methods [14–17], and others [18–21].

Network anomaly detection via traffic feature distributions is becoming more and more popular these days. As the measure of uncertainty, entropy can be used to summarize feature distributions in a compact form [22]. There are many forms of entropy, but only a few have been applied to network anomaly detection [23–27]. On this basis, we apply a Euclidean Distance-Based Multiscale Fuzzy Entropy

(EDM-Fuzzy) algorithm which we proposed to detect abnormal network traffic as a useful supplement of other approaches.

Investigation irregularity of signals generated by complex systems is valuable to predict the future states as well as detect abnormal behaviors [28]. In order to quantitatively analyze signal irregularity and diagnose system anomalies, researchers have proposed various signal complexity and uncertainty indicators, such as algorithmic complexity [29], Shannon Entropy [30], Approximate Entropy [31], Sample Entropy [32], Fuzzy Entropy [33], Multiscale Entropy (MSE) [34], and Composite Multiscale Entropy (CMSE) [35]. Entropy-based technologies have been widely applied in diagnosing the anomalies of various systems. For example, Shannon Entropy was applied in detecting faults of mechanical systems [36], MSE was applied in fault diagnosis of power systems [37], and so on.

However, the existing methods calculate the distance between vectors solely using the two most different elements of the two vectors. Furthermore, the similarity of vectors is calculated with Heaviside function, which has a problem of bouncing between 0 and 1. To this end, we proposed a novel entropy technology named EDM-Fuzzy in the paper [38]. The EDM-Fuzzy technology uses the sum of the Euclidean distances of the elements corresponding to two vectors instead of the largest element difference between the two vectors and uses the hyperbolic function to calculate the similarity between the two vectors. Thus, the EDM-Fuzzy technology avoids the two disadvantages inherent in the other entropy technologies and measures entropy values of system signals more precisely, accurately, and stably. In this paper, we apply the EDM-Fuzzy algorithm to characterize network anomaly traffic. We first briefly introduce the EDM-Fuzzy algorithm and then introduce the botnet CTU-13 dataset and the Distributed Denial of Service (DDoS) attack CICDDoS2019 dataset used in this paper. Then, the basic characteristics of these two datasets are introduced. Then, the EDM-Fuzzy entropy value analysis is performed on two datasets. Finally, we analyze the characteristics of the normal traffic and investigate the characteristics of the malicious traffic by comparing the differences between the normal and malicious traffic.

The rest of this paper is organized as follows. The related works are introduced in Section 2, and the EDM-Fuzzy entropy technology and network traffic traces are introduced in Sections 3 and 4, respectively. Section 5 is the analysis of network anomaly traffic with EDM-Fuzzy entropy. Section 6 concludes the paper and introduces the future work.

2. Related Works

2.1. Network Anomaly Traffic Detection Approaches. Network anomaly traffic detection approaches have been extensively explored. The existing approaches can be mainly classified into six categories [1]: classification-based methods [2–4], clustering-based methods [5–9], statistical methods [10, 11], stochastic methods [12, 13], deep-learning-based methods [14–17], and others [18–21]. A classification-based approach is a supervised learning algorithm. Classification algorithms such as logistic regression, k-nearest neighbor algorithm, decision tree, and support vector machine are commonly used. More recently, several hybrid classification models were proposed [3–5]. However, in most cases, labeling data manually is highly time-consuming and inefficient. Clustering techniques are used to identify clusters and outliers in multiple low-dimensional spaces. The evidence of traffic structure provided by these multiple clusters is then combined to produce an abnormality ranking of network traffic [39]. Several distance-based metrics are commonly used in anomaly detection, such as the Euclidean distance, Manhattan distance, and dynamic time warping (DTW) distance. However, the number of clusters is difficult to decide and different numbers of clusters would produce extremely different results. In a statistical method, an abnormality is often determined by checking whether the traffic complies with the assumed distribution model and whether the value is larger than a preset threshold. The most frequent assumptions are Gaussian distributions, Poisson distributions, multivariate Gaussian distributions, and so on. The model systematically analyzes abnormal behaviors of the network, but detection of such abnormalities is difficult since there will be cases that do not obey the presumed distributions. Stochastic processes like Hidden Markov Model and Conditional Random Field were also frequently applied in detection of traffic anomaly [12, 13]. Due to the success of deep-learning technologies in image processing and natural language processing, they have been intensively studied in network intrusion detection [14, 15], network traffic tracking [16], and network traffic abnormal behavior detection [17]. Besides, time-series density analysis [18], wavelet [19], principal components analysis [20], and ensemble learning technologies [21] have been extensively investigated in network anomaly detection.

2.2. Entropy-Based Technologies. Entropy-based technologies are highly valued in detecting the degree of disorder or irregularity of a complex system. Thus, there have been a number of entropy-based technologies being proposed and being widely applied in detecting anomalies of complex systems. Khan et al. [37] presented an entropy-based approach for detecting faults in power systems. An entropy-based methodology was proposed in paper [40] to extract characteristics from signals of smart meters to effectively classify power quality problems. The Kullback-Shannon Entropy was applied as a standalone feature to predict failure in lubricated surfaces [41].

Pincus [31], Richman, and Moorman [32] and Costa et al. [34, 42] proposed Approximate Entropy, Sample Entropy, and MSE to measure signal complexity, respectively. Although MSE has been widely applied, the variance of the entropy values increases significantly as the time series is coarse-grained for larger time scales [43]. In order to solve the problem, Wu et al. proposed CMSE [35] and introduced a composite averaging method to reduce the variance. Niu and Wang [44] applied CMSE to study the characteristics of stock market indices and found that CMSE is more stable and reliable than MSE. Chen et al. [33] proposed Fuzzy

Approximate Entropy (FuzzyEn) and applied it in the study of surface muscle signal. Wang et al. [45] proposed fractional fuzzy entropy to study physics financial dynamics. Li et al. [46] integrated fractional fuzzy entropy with a binary tree support vector machine to perform early diagnosis of rolling bearing faults. Composite multiscale fuzzy entropy is proposed in paper [47] and is applied to extract the hidden features of vibration signals.

Entropy-based network anomaly detection via traffic feature characterization is becoming more and more popular these days. Ranjan et al. [23] proposed a worm detection algorithm that measures Shannon Entropy values for traffic and alarms on sudden bursts. Gu et al. [24] applied Shannon maximum entropy estimation to draw the network baseline distribution and to build a multiperspective view of network traffic. Paper [25] presented a novel network intrusion detection system using Shannon Entropy and traffic distributions of the source port. Paper [26] proposed a hybrid DDoS detection method, which integrates Kernel Online Anomaly Detection (KOAD), Shannon Entropy, and Mahalanobis Distance. In this study, Shannon Entropy is utilized with an online machine learning method to detect malicious traffic including DDoS attacks and Flash Event traffic. Paper [27] presented anomaly detection in activities of daily living based on entropy measures.

However, there are still two disadvantages in the existing state-of-the-art entropy algorithms, such as MSE, CMSE, RCMSE, MMSE, and FuzzyEn. That is, the existing methods calculate the distance between vectors solely based on the two most different elements of the two vectors. Furthermore, the similarity of vectors is calculated using Heaviside function, which has a problem of bouncing between 0 and 1. In order to address the shortcomings of existing state-of-the-art entropy algorithms, we proposed novel entropy technology [38], named EDM-Fuzzy.

3. EDM-Fuzzy Technology

EDM-Fuzzy measures the distance of the two vectors with Euclidean distance taking all the corresponding elements in the two vectors into the computation. Furthermore, in order to solve the problem of instability, we choose the hyperbolic function as the fuzzy function instead of the Heaviside function to define the similarity between vectors with full-range continuous values from zero to one based on the Euclidean distance of the two vectors. The computation process of EDM-Fuzzy is formally described in Algorithm 1.

The goal of the algorithm is to measure the complexity and irregularity of time series more accurately and stably. The input of the algorithm is a time series $X = \{x_1, x_2, \ldots x_N\}$, time scale τ, vector dimension m, tolerance coefficient r, and standard deviation SD of time series X. The output of the algorithm is the EDM-Fuzzy entropy value of time series X at time scale τ. The general process of the algorithm is first to coarse-grain the time series with time scale τ, then split the time series into m-dimensional vectors, move the vectors to its centroid, and finally, calculate the Euclidean distance of the two vectors and compute the Euclidean distance based on fuzzy sample entropy value of time series. For parameters m and r, m is usually set to 2 and r generally ranges from 0.1 to 0.2. In our experiments, r is set to 0.15; that is, the similarity tolerance is set to 0.15*SD. Here, SD represents the standard deviation of the original time series.

4. Network Traffic Trace

A suitable network traffic trace is essential to the research of the characterization of network anomaly traffic. The traces used in this paper are publicly accessible, within which anomaly activities including botnet and DDoS attack were recorded. Through analysis of these public traces with EDM-Fuzzy algorithm, we can further discover the characteristics of such anomaly activities.

4.1. Botnet Traffic Trace.

The botnet traffic trace used in this section is the CTU-13 trace that was collected and provided by the Stratosphere Laboratory of CTU University in the Czech Republic [48, 49]. This trace contains botnet traffic as well as normal background traffic. The CTU-13 trace contains 13 botnet samples in different scenarios. In each sample, a specific malware is executed and different operations were performed accordingly. The brief information of the trace is shown in Tables 1 and 2.

Table 1 shows the characteristics of 13 types of botnet scenarios. Each type of botnet has different characteristics of malicious behavior. In Table 1, IRC represents the network relay chat protocol, SPAM represents spam, CF represents malicious clicks, PS represents port scan, FF represents fast flux, P2P refers to end-to-end, DDoS refers to Distributed Denial of Service, and US refers to a protocol that is controlled and completed by humans. The basic characteristics of each botnet can be seen in Table 1.

Table 2 shows the duration, the number of data packets, the type of malware, and the number of infected computers of these 13 types of botnet scenarios. The duration of botnet scenarios varies from 15 minutes to 66 hours. The number of infected hosts for most scenarios is 1 host. Neris-3, Rbot-3, Rbot-4, and NSIS.ay scenarios have 10 and 3 infected hosts, respectively.

4.2. DDoS Traffic Trace.

DDoS attack is an abnormal network behavior designed to exhaust server resources. It will cause server congestion and thus will be unable to provide services to users. The traffic trace used in this paper is the CICDDoS2019 which was published by the Canadian Cyber Security Institute (CIC) [50]. The CICDDoS2019 trace contains common and latest DDoS attacks. There are mainly two categories of DDoS attack methods involved in this trace, DDoS reflection attack and DDoS direct attack. DDoS reflection attack method utilizes routers, servers, and other facilities to respond to requests, thus reflecting the attack traffic to hide the source of the attack. The direct DDoS attack method is to directly attack the target using the controlled hosts. Compared with the reflection type attack, the direct attack method has a lower degree of anonymity. The specific attack types and attack duration time in the CICDDoS2019 dataset are shown in Tables 3 and 4.

```
Inputs:
  Time series: $X = \{x_1, x_2, \ldots x_N\}$.
  Time scale: $\tau \geq 1$.
  Vector dimension: $m \geq 2$.
  Tolerance coefficient: $0.1 \leq r \leq 0.2$.
  Standard deviation of time series $X : SD$.
Output:
  EDM-Fuzzy entropy value of time series $X$ at time scale $\tau$.
(1) for $k = 1$ to $\tau$
(2)     $p = \lfloor (N - k + 1)/\tau \rfloor$;
(3)     for $j = 1$ to $p$
(4)         Coarse-graining the time series $y_{k,j}^{(\tau)} = (1/\tau) \sum_{i=(j-1)\tau+k}^{j\tau+k-1} x_i$;
(5)     end for
(6) end for
(7) for $k = 1$ to $\tau$
(8)     for $i = 1$ to $p - m + 1$
(9)         Calculate the mean of each vector
                $u_{k,m}^{(\tau)}(i) = (1/m) \sum_{q=0}^{m-1} y_{k,i+q}^{(\tau)}$;
(10)        Move the vectors
            $Y_{k,m}^{(\tau)}(i) = \{y_{k,i}^{(\tau)}, y_{k,i+1}^{(\tau)}, \ldots, y_{k,i+m-1}^{(\tau)}\} - u_{k,m}^{(\tau)}(i)$;
(11)    end for
(12)    for $i = 1$ to $p - m$
(13)        for $j = i + 1$ to $p - m + 1$
(14)            Calculate the Euclidean distance of the two
            vectors $Y_{k,m}^{(\tau)}(i)$ and $Y_{k,m}^{(\tau)}(j)$:
                $d_{k,m}^{(\tau)}(i, j) = d[Y_{k,m}^{(\tau)}(i), Y_{k,m}^{(\tau)}(j)]$
                $= \sqrt{\sum_{w=0}^{m-1} ((y_{k,i+w}^{(\tau)} - u_{k,m}^{(\tau)}(i)) - (y_{k,j+w}^{(\tau)} - u_{k,m}^{(\tau)}(j)))^2}$;
(15)            Calculate the similarity between $Y_{k,m}^{(\tau)}(i)$ and $Y_{k,m}^{(\tau)}(j)$ vectors
                $S_{k,m}^{(\tau)}(i, j) = \mu(d_{k,m}^{(\tau)}(i, j), r) = (1/(1 + d_{k,m}^{(\tau)}(i, j)/r))$;
(16)        end for
(17)        Calculate the average similarity between vector
            $Y_{k,m}^{(\tau)}(i)$ and the other vectors
             $B_{k,m}^{(\tau,r)}(i) = (1/p - m - 1) \sum_{i=1, j \neq i}^{p-m} S_{k,m}^{(\tau)}(i, j)$;
(18)    end for
(19)        Compute the average of $B_{k,m}^{(\tau,r)}(i)$, that is,
            $B_{k,m}^{(\tau,r)} = (1/p - m) \sum_{i=1}^{p-m} B_{k,m}^{(\tau,r)}(i)$;
(20)        Set dimensional length of vectors to $m + 1$ and repeat step 8~19 to calculate average similarity between each pair of $m + 1$
    points vectors in coarse-grained time series; you can get $A_{k,m+1}^{(\tau,r)}(i)$ and $A_{k,m+1}^{(\tau,r)}$
(21)            $A_{k,m+1}^{(\tau,r)}(i) = (1/p - m - 2) \sum_{j \neq i}^{p-m-1} S_{k,m+1}^{(\tau)}(i, j)$;
(22)            $A_{k,m+1}^{(\tau,r)} = (1/p - m - 1) \sum_{i=1}^{p-m-1} A_{k,m+1}^{(\tau,r)}(i)$;
(23)        Compute the Euclidean distance based on fuzzy
                sample entropy value for every $y_k^{(\tau)}$,
            $Eucli\ de\ anFuzzyEn(y_k^{(\tau)}, m, r) = \lim_{p \longrightarrow \infty} \{-\ln(A_{k,m+1}^{(\tau,r)} / B_{k,m}^{(\tau,r)})\}$;
(24) end for
(25) Compute the fuzzy sample entropy value for the original time series at time scale $\tau$
            $E\ DM\ Fuzzy(m, r) = (1/\tau) \sum_{k=1}^{\tau} EuclideanFuzzyEn(y_k^{(\tau)}, m, r)$.
```

Algorithm 1: EDM-fuzzy algorithm.

Two days of traffic were collected in this trace, which were November 3 and December 1, as shown in Tables 3 and 4, respectively. There were 10 types of DDoS attacks on December 1, that is, NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDPLag, and TFTP attack that lasted from 10 : 30 to 17 : 15. On November 3, there were 7 types of DDoS attacks including PortMap, NetBIOS, LDAP, MSSQL, UDP, UDPLag, and SYN attacks; the duration is from 9 : 40 to 17 : 35. The attack method of each type of DDoS attack in the CICDDoS2019 dataset is shown in Figure 1.

As shown in Figure 1, there are two types of DDoS attacks in the CICDDoS2019 trace, namely, reflection DDoS attacks and direct DDoS attacks. Both DDoS attacks are based on TCP/UDP protocol execution. As shown in the figure above, 9 types of DDoS attacks such as MSSQL, SSDP, DNS, LDAP, NetBIOS, SNMP, PortMap, NTP, and TFTP, are distributed reflective denial attacks, while SYN, UDP, and UDPLag Flood are direct DDoS attacks. In Figure 1, TCP-based attacks include MSSQL, SSDP, and SYN, and UDP-based attacks include NTP, TFTP, UDP, and UDPLag. The remaining types of attacks such as DNS, LDAP,

Table 1: Scenarios of Botnet traffic.

ID	IRC	SPAM	CF	PS	DDoS	FF	P2P	US	HTTP
1	✓	✓	✓	—	—	—	—	—	—
2	✓	✓	✓	—	—	—	—	—	—
3	✓	—	—	✓	—	—	—	✓	—
4	✓	—	—	—	✓	—	—	✓	—
5	—	✓	—	✓	—	—	—	—	✓
6	—	—	—	✓	—	—	—	—	—
7	—	—	—	—	—	—	—	—	✓
8	—	—	—	✓	—	—	—	—	—
9	✓	✓	✓	✓	—	—	—	—	—
10	✓	—	—	—	✓	—	—	—	✓
11	✓	—	—	—	✓	—	—	—	✓
12	—	—	—	—	—	—	✓	—	—
13	—	✓	—	✓	—	—	—	—	✓

Table 2: Traffic volume of 13 types of Botnet scenarios.

ID	Duration (hours)	Packets	Malware type	Infected hosts
1	6.15	71971482	Neris-1	1
2	4.21	71851300	Neris-2	1
3	66.85	167730395	Rbot-1	1
4	4.21	62089135	Rbot-2	1
5	11.63	4481167	Virut-1	1
6	2.18	38764357	Menti	1
7	0.38	7467139	Sogou	1
8	19.5	155207799	Murlo	1
9	5.18	115415321	Neris-3	10
10	4.75	90389782	Rbot-3	10
11	0.26	6337202	Rbot-4	3
12	1.21	13212268	NSIS.ay	3
13	16.36	50888256	Virut-2	1

Table 3: DDoS attack time on November 3.

Type ID	Attack type	Attack time
1	PortMap	9 : 43–9 : 51
2	NetBIOS	10 : 00–10 : 09
3	LDAP	10 : 21–10 : 30
4	MSSQL	10 : 33–10 : 42
5	UDP	10 : 53–11 : 03
6	UDPLag	11 : 14–11 : 24
7	SYN	11 : 28–17 : 35

Table 4: DDoS attack time on December 1.

Type ID	Attack type	Attack time
1	NTP	10 : 35–10 : 45
2	DNS	10 : 52–11 : 05
3	LDAP	11 : 22–11 : 32
4	MSSQL	11 : 36–11 : 45
5	NetBIOS	11 : 50–12 : 00
6	SNMP	12 : 12–12 : 23
7	SSDP	12 : 27–12 : 37
8	UDP	12 : 45–13 : 09
9	UDPLag	13 : 11–13 : 15
10	TFTP	13 : 35–17 : 15

NetBIOS, SNMP, PortMap, and other types of attacks are executed by using TCP or UDP.

5. Analysis of Network Anomaly Traffic with EDM-Fuzzy Entropy

Entropy-based time-series complexity measurement methods are widely used in fault diagnosis and anomaly detection of various complex systems. In this section, we apply EDM-Fuzzy in network traffic anomaly characterization and detection. The analysis of anomaly traffic characteristics based on MSE of Euclidean distance is an important part of the study of abnormal traffic. In this section, two anomalies of botnet and DDoS attack will be analyzed by Euclidean distance multiscale entropy. This section will calculate the entropy value of these two abnormal network protocol time series to obtain the entropy curves of the two and study the characteristics of the abnormal traffic by comparing the difference in the entropy curves.

5.1. Botnet Traffic in ARP. In this section, we will study the EDM-Fuzzy entropy characteristics of 13 types of botnets abnormal ARP traffic in the CTU-13 dataset. According to the TCP/IP architecture, the ARP protocol is located in the IP layer of the network layer, and its main function is to provide address translation services and find the network physical address of the host corresponding to the IP address. We first calculate the entropy values for each type of botnet using ARP protocol traffic data in the CTU-13 dataset at time scales from 1 to 40. The entropy curves of 13 types of botnets in the CTU-13 dataset with scale factors from 1 to 40 are shown in Figure 2.

As can be seen from the figure, there are common trends shared by entropy curves of most types of botnet traffic. More specifically, there is a reflection point for 11 entropy curves (Neris-1, Neris-2, Rbot-1, Rbot-2, Virut-1, Menti, Sogou, Murlo, Neris-3, NSIS.ay, and Virut-2) when the time scale is 20, and the second reflection point appears at the time scale of 30 for all entropy curves. For the above 11 types of botnet ARP traffic, the entropy values between the inflection points increase first and then decrease. The trend of the entropy curves of Rbot-3 and Rbot-4 is different from other types of abnormal behavior. Entropy curves of Rbot-3 and Rbot-4 are in a steady growth state when the time scale is around 20, but when the time scale is 30, there is also an inflection point. Moreover, entropy values of Rbot-4 are significantly larger compared to those of other types of anomalies. The above results illustrate that the attack methods of Rbot-3 and Rbot-4 are different from the other types of botnets. This difference is caused by the way they infect hosts, and the complexity of the botnet is consistent with the complexity of the ARP protocol.

5.2. DDos Traffic in ARP. In this section, we will study the EDM-Fuzzy entropy characteristics of the malicious traffic of the distributed denial attacks on November 3 and December 1 in the CICDDoS2019 dataset. Through analysis of the trend of entropy values, it is possible to understand more characteristics of DDoS attack traffic. As introduced in the dataset, there were seven and ten types of distributed denial

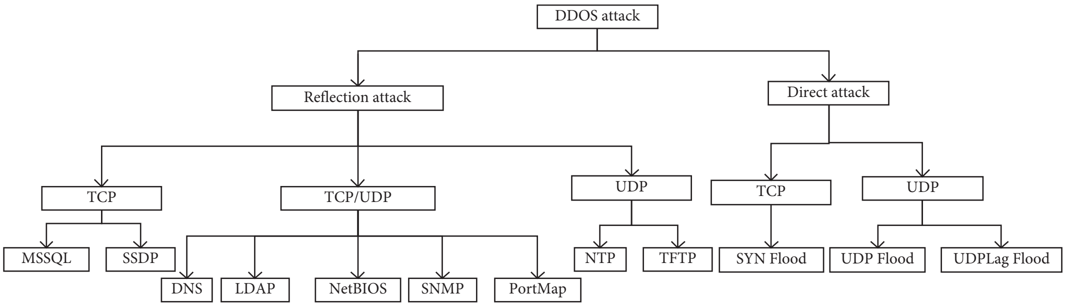


Figure 1: Types of DDoS attacks.

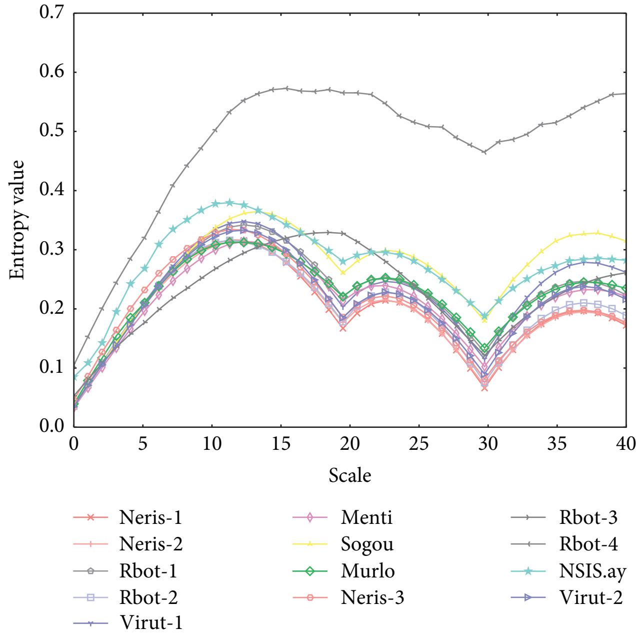


Figure 2: Entropy curves of 13 types of botnets in the CTU-13 dataset.

attacks launched on November 3 and December 1, respectively. In this section, we first calculate the entropy value of the ARP traffic of each type of DDoS attack in the CICDDoS2019 dataset at time scales from 1 to 40, and the entropy value curves are shown in Figures 3 and 4.

Figures 3 and 4 show the entropy curves of the ARP traffic for DDoS attacks on November 3 and December 1, respectively. As can be seen from Figures 3 and 4, there are three characteristics of the entropy values of DDoS attacks on November 3 and December 1. Firstly, for both November 3 and December 1, all of the entropy values of DDoS attacks are larger than 0.18 when the time scale is larger than 4. Secondly, the entropy values of most types of DDoS attacks gradually stabilized to a value between 0.3 and 0.5 when the time scale is larger than 10. There is only the NetBIOS attack that has a relatively big fluctuation that may exceed the upper bound. Thirdly, the entropy values of the same type of DDoS attack for the two different days are quite similar. The possible underlying principle of the characteristics is that, in DDoS attacks, attackers continuously use distributed attacks to attack hosts, and the attacked hosts continue to

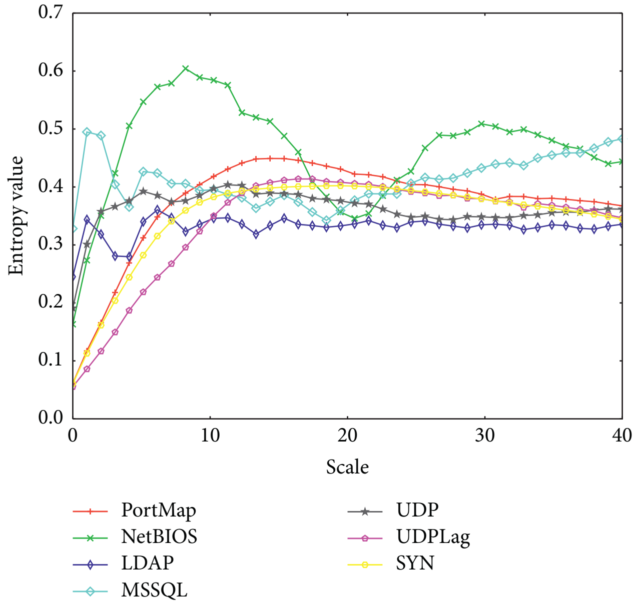


Figure 3: Entropy curves of DDoS attacks on November 3.

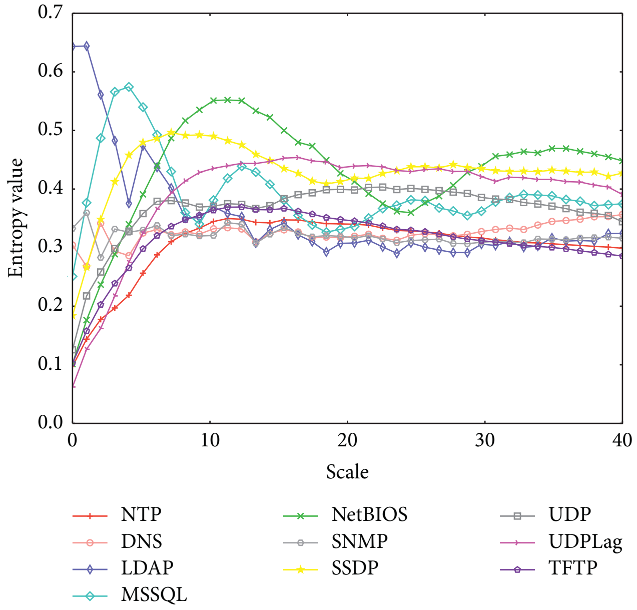


Figure 4: Entropy curves of DDoS attacks on December 1.

communicate during the attack. In the communication process, the ARP protocol continuously performs address resolution, while the number of resolved source IP addresses and destination IP addresses remains stable, so the entropy values of ARP traffic are gradually stabilized.

5.3. Normal Traffic in ARP. In this section, we will analyze the characteristics of network traffic under normal status. The normal traffic trace used in this paper is captured and published by the Stratosphere laboratory.

In order to study the entropy characteristics of normal traffic, the EDM-Fuzzy entropy values are calculated on the CTU-Normal-20 and CTU-Normal-23 traces with time scales from 1 to 40 and the results are shown in Figure 5.

As can be seen from Figure 5, the entropy values of normal ARP traffic exhibit different characteristics. The entropy values grow steadily when the time scale grows from 1 to 30, and then the entropy values grow slowly as the time scale increases. Furthermore, for all time scales, the entropy values of normal ARP traffic are smaller than 0.18.

Compared with the time series of the CTU-13 dataset, Figure 5 shows that the entropy value of the ARP protocol in the CTU-13 dataset exhibits its own unique laws. Compared with the CICDDoS2019 dataset, the basic law of the ARP protocol is that the entropy curve increases first and then gradually stabilizes.

5.4. Malicious versus Normal. In this section, we will compare the ARP traffic entropy curves between botnet, DDoS attack, and normal status and then characterize the differences between normal and abnormal traffic.

By comparing the entropy curves of ARP traffic of botnet, DDoS attack, and normal status, we find out the following main differences between normal traffic and malicious traffic. In the entropy curves of 13 types of botnets, 11 entropy curves (Neris-1, Neris-2, Rbot-1, Rbot-2, Virut-1, Menti, Sogou, Murlo, Neris-3, NSIS.ay, and Virut-2) have a reflection point at the time scale of 20, and all entropy curves have a reflection point at the time scale of 30. In the entropy curves of DDoS traffic, all of the entropy values of DDoS attacks are larger than 0.18 when the time scale is larger than 4, and most types of DDoS attacks gradually stabilized to entropy values between 0.3 and 0.5 when the time scale is larger than 10. In contrast, the entropy values of normal ARP traffic grow slowly as the time scale increases and the entropy values are smaller than 0.18 for all time scales.

In order to be presented more intuitively, the main characteristics of entropy curves of ARP traffic of botnet, DDoS attack, and normal status are listed in Table 5.

On the basis of the above analysis, it is reasonable to summarize that the characteristics of entropy curves of ARP traffic of botnet, DDoS, and normal status are quite distinguishable. Thus, the characteristics are easy to be used to detect these types of network traffic anomalies. In the future, we will study characteristics of EDM-Fuzzy entropy curves of more types of network traffic anomalies and utilize the learned characteristics of network traffic anomalies in combination with intelligent algorithms to automatically detect network anomalies.

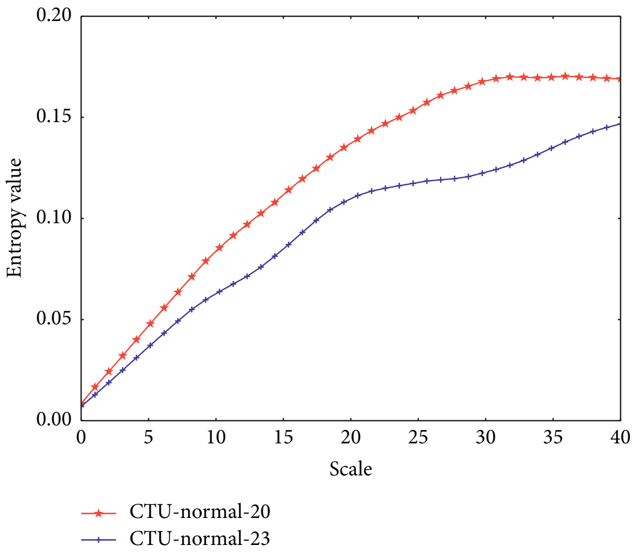


Figure 5: Entropy values of normal ARP traffic.

Table 5: Characteristics of EDM-Fuzzy entropy curves of ARP traffic of botnet, DDoS, and normal status.

	Botnet	DDoS	Normal
Value	Mostly between 0.1 and 0.4.	Larger than 0.18 when the time scale is larger than 4.	Smaller than 0.18.
Trend	Entropy curves of all types of botnet traffic have an inflection point at a time scale of 30; 11 types have an inflection point at a time scale of 20.	Gradually stabilized to a value between 0.3 and 0.5 when the time scale is larger than 10.	Increase steadily from 0 to 0.18.

6. Conclusions

In order to raise the cybersecurity awareness capability of network administrators, it is necessary to develop new technologies for detecting network anomalies more accurately and efficiently. The basis of such network anomaly detection technologies is to understand the characteristics of abnormal network traffic. In this paper, we apply the EDM-Fuzzy technology as a tool to analyze the characteristics of abnormal network traffic such as botnet network traffic and DDoS attack traffic. The EDM-Fuzzy is a technology that we proposed for analyzing and diagnosing faults/anomalies of complex systems by measuring the complexity and regularity of their time-series signals. The experimental analysis shows that the EDM-Fuzzy entropy curve is capable of characterizing the difference between normal traffic and abnormal traffic and the characteristics are easy to be used to detect various types of network traffic anomalies. In the current work, we have not investigated other types of network anomalies and have not finished the automatic detection of network traffic anomalies. In the future, we will investigate EDM-Fuzzy entropy characteristics for more types of network anomalies and then integrate the EDM-Fuzzy entropy and deep-learning technologies to propose the novel network anomaly detection method.

Data Availability

The botnet traffic trace used in this section is the CTU-13 trace that was collected and provided by the Stratosphere Laboratory of CTU University in the Czech Republic [48, 49]. The traffic trace used in this paper is the CICD-DoS2019 which was published by the Canadian Cyber Security Institute (CIC) [48].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was in part supported by the National Key Research and Development Program of China under Grant 2019YFB2102100, the China Postdoctoral Science Foundation under Grant 2016M600465, the Key Research and Development Program of Zhejiang Province under Grant 2019C03134, and the National Natural Science Foundation of China under Grant 61772165.

References

[1] H. Zhang, Y. Luo, Q. Yu, L. Sun, X. Li, and Z. Sun, "A framework of abnormal behavior detection and classification based on big trajectory data for mobile networks," *Security and Communication Networks*, vol. 2020, Article ID 8858444, 15 pages, 2020.

[2] J. Gou, W. Qiu, Z. Yi, Y. Xu, Q. Mao, and Y. Zhan, "A local mean representation-based *K*-nearest neighbor classifier," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 3, pp. 1–25, 2019.

[3] V. Dutta, M. Choraś, R. Kozik, and M. Pawlicki, "Hybrid model for improving the classification effectiveness of network intrusion detection," in *Proceedings of the 13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*, Burgos, Spain, September 2020.

[4] F. Al-Obeidat and E.-S. M. El-Alfy, "Hybrid multicriteria fuzzy classification of network traffic patterns, anomalies, and protocols," *Personal and Ubiquitous Computing*, vol. 23, no. 5-6, pp. 777–791, 2019.

[5] J. Dromard, G. Roudiere, and P. Owezarski, "Online and scalable unsupervised network anomaly detection method," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34–47, 2017.

[6] E. Bigdeli, M. Mohammadi, B. Raahemi, and S. Matwin, "Incremental anomaly detection using two-layer cluster-based structure," *Information Sciences*, vol. 429, pp. 315–331, 2018.

[7] S. Baek, D. Kwon, S. C. Suh, H. Kim, I. Kim, and J. Kim, "Clustering-based label estimation for network anomaly detection," *Digital Communications and Networks*, vol. 7, no. 1, pp. 37–44, 2020.

[8] G. Pu, L. Wang, J. Shen, and F. Dong, "A hybrid unsupervised clustering-based anomaly detection method," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 146–153, 2020.

[9] J. Mao, Y. Hu, D. Jiang, T. Wei, and F. Shen, "CBFS: a clustering-based feature selection mechanism for network anomaly detection," *IEEE Access*, vol. 8, pp. 116216–116225, 2020.

[10] H. H. Jazi, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, "Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling," *Computer Networks*, vol. 121, no. 5, pp. 25–36, 2017.

[11] M. Thottan, G. Liu, and C. Ji, "Anomaly detection approaches for communication networks," in *Algorithms for Next Generation Networks*, G. Cormode and M. Thottan, Eds., Springer, London, UK, pp. 239–261, 2010.

[12] J.-H. Bang, Y.-J. Cho, and K. Kang, "Anomaly detection of network-initiated LTE signaling traffic in wireless sensor and actuator networks based on a Hidden semi-Markov Model," *Computers & Security*, vol. 65, pp. 108–120, 2017.

[13] G. Zheng, X. Xu, and J. Yan, "SD-CRF: a DoS attack detection method for SDN," in *Proceedings of the 2020 IEEE 20th International Conference on Communication Technology (ICCT)*, Nanning, China, October 2020.

[14] Z. Wang, Y. Zeng, Y. Liu, and D. Li, "Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection," *IEEE Access*, vol. 9, no. 1, pp. 16062–16091, 2021.

[15] Z. Wang, Y. Liu, D. He, and S. Chan, "Intrusion detection methods based on integrated deep learning model," *Computers & Security*, vol. 103, p. 102177, 2021.

[16] D. K. K. Reddy, H. S. Behera, J. Nayak, P. Vijayakumar, B. Naik, and P. K. Singh, "Deep neural network based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities," *Transactions on Emerging Telecommunications Technologies*, p. e4121, 2020.

[17] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark," *IEEE Access*, vol. 6, pp. 59657–59671, 2018.

[18] K. Flanagan, E. Fallon, P. Connolly, and A. Awad, "Network anomaly detection in time series using distance based outlier detection with cluster density analysis," in *Proceedings of the 2017 Internet Technologies and Applications (ITA)*, Wrexham, UK, September 2017.

[19] C. B. Zerbini, L. F. Carvalho, T. Abrão, and M. L. Proença, "Wavelet against random forest for anomaly mitigation in software-defined networking," *Applied Soft Computing*, vol. 80, pp. 138–153, 2019.

[20] Sharipuddin, B. Purnama, Kurniabudi et al., "Features extraction on IoT intrusion detection system using principal components analysis (PCA)," in *Proceedings of the 2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI)*, Yogyakarta, Indonesia, October 2020.

[21] Y. Zhong, W. Chen, Z. Wang et al., "HELAD: a novel network anomaly detection model based on heterogeneous ensemble learning," *Computer Networks*, vol. 169, p. 107049, 2020.

[22] P. Bereziński, B. Jasiul, and M. Szpyrka, "An entropy-based network anomaly detection method," *Entropy*, vol. 17, no. 4, pp. 2367–2408, 2015.

[23] S. Ranjan, S. Shah, A. Nucci, M. Munafo, R. Cruz, and S. Muthukrishnan, "DoWitcher: effective worm detection and containment in the internet core," in *Proceedings of 26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, pp. 2541–2545, Anchorage, AL, USA, May 2007.

[24] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC '05)*, p. 32, Berkeley, CA, USA, October 2005.

[25] S. Ransewa, N. Elz, N. Thanon, and S. Intajag, "Anomaly detection using source port data with shannon entropy and EWMA control chart," in *Proceedings of the 2018 18th International Conference on Control, Automation and Systems (ICCAS)*, pp. 596–601, PyeongChang, Korea, October 2018.

[26] S. Daneshgadeh, T. Kemmerich, T. Ahmed, and N. Baykal, "An empirical investigation of DDoS and Flash event detection using Shannon entropy, KOAD and SVM combined," in *Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 658–662, Honolulu, HI, USA, February 2019.

[27] A. Howedi, A. Lotfi, and A. Pourabdollah, "An entropy-based approach for anomaly detection in activities of daily living in the presence of a visitor," *Entropy*, vol. 22, no. 8, p. 845, 2020.

[28] J. S. Cánovas, G. García-Clemente, and M. Muñoz-Guillermo, "Comparing permutation entropy functions to detect structural changes in time series," *Physica A: Statistical Mechanics and its Applications*, vol. 507, no. 1, pp. 153–174, 2018.

[29] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *International Journal of Computer Mathematics*, vol. 2, no. 1–4, pp. 156–168, 1968.

[30] S. Claude, "A mathematical theory of communications," *Bell Labs Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[31] S. M. Pincus, "Approximate entropy as a measure of system complexity," *Proceedings of the National Academy of Sciences*, vol. 88, no. 6, pp. 2297–2301, 1991.

[32] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *American Journal of Physiology: Heart and Circulatory Physiology*, vol. 278, no. 6, pp. 2039–2049, 2000.

[33] W. Chen, J. Zhuang, W. Yu, and Z. Wang, "Measuring complexity using FuzzyEn, ApEn, and SampEn," *Medical Engineering & Physics*, vol. 31, no. 1, pp. 61–68, 2009.

[34] M. Costa, A. L. Goldberger, and C. K. Peng, "Multiscale entropy analysis of biological signals," *Physical Review E*, vol. 71, no. 2, pp. 1–18, 2005.

[35] S.-D. Wu, C.-W. Wu, S.-G. Lin, C.-C. Wang, and K.-Y. Lee, "Time series analysis using composite multiscale entropy," *Entropy*, vol. 15, no. 3, pp. 1069–1084, 2013.

[36] L. Dou, S. Wan, and C. Zhan, "Application of multiscale entropy in mechanical fault diagnosis of high voltage circuit breaker," *Entropy*, vol. 20, no. 5, pp. 325–329, 2018.

[37] I. Khan, Y. L. Xu, S. Kar, M. Chow, and V. Bhattacharjee, "Compressive sensing and morphology singular entropy-based real-time secondary voltage control of multi-area power systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3796–3807, 2019.

[38] R. Zhou, X. Wang, J. Wan, and N. Xiong, "EDM-fuzzy: an euclidean distance based multiscale fuzzy entropy technology for diagnosing faults of industrial systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4046–4054, 2020.

[39] P. Casas, J. Mazel, and P. Owezarski, "UNADA: unsupervised network anomaly detection using sub-space outliers ranking," in *Networking 2011*, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds., Springer, Berlin-Germany, 2011.

[40] F. A. S. Borges, R. A. S. Fernandes, I. N. Silva, and C. B. S. Silva, "Feature extraction and power quality disturbances classification using smart meters signals," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 824–833, 2016.

[41] S. A. Shevchik, F. Saeidi, B. Meylan, and K. Wasmer, "Prediction of failure in lubricated surfaces using acoustic time-frequency features and random forest algorithm," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1541–1553, 2017.

[42] M. Costa, A. L. Goldberger, and C. K. Peng, "Multiscale entropy analysis of complex physiologic time series," *Physical Review Letters*, vol. 89, no. 6, pp. 1–4, 2002.

[43] C. M. Galanakis, "Modeling in food and bioproducts processing using Boltzmann entropy equation: a viewpoint of future perspectives," *Food and Bioproducts Processing*, vol. 106, no. 1, pp. 102–107, 2017.

[44] H.-L. Niu and J. Wang, "Entropy and recurrence measures of a financial dynamic system by an interacting voter system," *Entropy*, vol. 17, no. 5, pp. 2590–2605, 2015.

[45] Y. Wang, S. Zheng, W. Zhang, G. Wang, and J. Wang, "Fuzzy entropy complexity and multifractal behavior of statistical physics financial dynamics," *Physica A: Statistical Mechanics and its Applications*, vol. 506, no. 15, pp. 486–498, 2018.

[46] Y. Li, Y. Yang, X. Wang, B. Liu, and X. Liang, "Early fault diagnosis of rolling bearings based on hierarchical symbol dynamic entropy and binary tree support vector machine," *Journal of Sound and Vibration*, vol. 428, no. 18, pp. 72–86, 2018.

[47] J. Zheng, H. Pan, and J. Cheng, "Rolling bearing fault detection and diagnosis based on composite multiscale fuzzy entropy and ensemble support vector machines," *Mechanical Systems and Signal Processing*, vol. 85, no. 15, pp. 746–759, 2017.

[48] "The CTU-13 Dataset. a labeled dataset with botnet, normal and background traffic on stratosphere research laboratory," https://www.stratosphereips.org/datasets-ctu13.

[49] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100–123, 2014.

[50] Canadian Institute for Cybersecurity, *DDoS Evaluation Dataset (CIC-DDoS2019)*Canadian Institute for Cybersecurity, Fredericton, Canada, 2019, https://www.unb.ca/cic/datasets/ddos-2019.html.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9963829, 11 pages
https://doi.org/10.1155/2021/9963829

Research Article

Can Multipath TCP Be Robust to Cyber Attacks? A Measuring Study of MPTCP with Active Queue Management Algorithms

Yuanlong Cao, Ruiwen Ji, Lejun Ji, Mengshuang Bao, Lei Tao, and Wei Yang

School of Software, Jiangxi Normal University, Nanchang 330022, China

Correspondence should be addressed to Yuanlong Cao; ylcao@jxnu.edu.cn

Received 11 March 2021; Accepted 19 May 2021; Published 28 May 2021

Academic Editor: Zhe-Li Liu

With the development of social networks, more and more mobile social network devices have multiple interfaces. Multipath TCP (MPTCP), as an emerging transmission protocol, can fit multiple link bandwidths to improve data transmission performance and improve user experience quality. At the same time, due to the large-scale deployment and application of emerging technologies such as the Internet of Things and cloud computing, cyber attacks against MPTCP have gradually increased. More and more network security research studies point out that low-rate distributed denial of service (LDDoS) attacks are relatively popular and difficult to detect and are recognized as one of the most severe threats to network services. This article introduces six classic queue management algorithms: DropTail, RED, FRED, REM, BLUE, and FQ. In a multihomed network environment, we perform the performance evaluation of MPTCP under LDDoS attacks in terms of throughput, delay, and packet loss rate when using the six algorithms, respectively, by simulations. The results show that in an MPTCP-enabled multihomed network, different queue management algorithms have different throughput, delay, and packet loss rate performance when subjected to LDDoS attacks. Considering these three performance indicators comprehensively, the FRED algorithm has better performance. By adopting an effective active queue management (AQM) algorithm, the MPTCP transmission system can enhance its robustness capability, thus improving transmission performance. We suggest that when designing and improving the queue management algorithm, the antiattack performance of the algorithm should be considered: (1) it can adjust the traffic speed by optimizing the congestion control mechanism; (2) the fairness of different types of data streams sharing bandwidth is taken into consideration; and (3) it has the ability to adjust the parameters of the queue management algorithm in a timely and accurate manner.

1. Introduction

With the development of social networks and the large-scale application of multiple wireless access technologies (i.e., Bluetooth, Wi-Fi, GPRS, and 4G), more and more mobile social network terminal devices are equipped with multiple network interfaces of different standards. Multihomed terminals can access multiple networks at the same time and achieve multipath data transmission by fitting the bandwidth of multiple links so as to improve data transmission performance, maximize network resource utilization, and improve user experience quality [1] while the multipath TCP (MPTCP) protocol [2] can distribute the data across multiple end-to-end transmission paths, by enabling the use of several network interfaces of devices simultaneously, as shown in Figure 1. As a variant of the TCP technology, MPTCP preserves the standard socket application programming interfaces (APIs) that are used by most Internet applications. Hosts can establish an MPTCP connection by using the existing socket APIs, without requiring any modification or addition to the applications and still being compatible on today's Internet. Therefore, MPTCP protocol is considered to play a huge role in the application of future Internet data transmission services [3, 4]. In spite of MPTCP has some advantages when applied to concurrent transmission in the network, in the real environment, the commonly happening network attacks have led to frequent changes in network quality which makes the network connection has a negative impact on the performance of MPTCP.

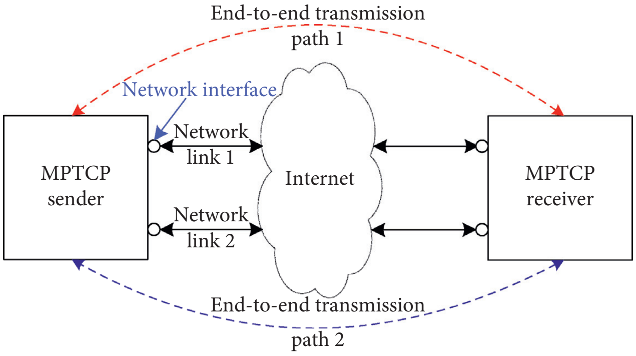


Figure 1: A basic MPTCP usage.

The MPTCP-based multipath data transmission is a process with complex network behavior. Although each transmission path can independently perform data transmission tasks according to its own network conditions, yet when a certain transmission path in the MPTCP multipath transmission system is attacked by a network, the reduction of the path transmission performance or the path failure will affect the transmission performance of other paths so as to produce adverse effects on the overall performance of multipath transmission such as robustness degradation [5]. Due to the large-scale deployment and application of emerging information technologies such as the Internet of Things (IoT) and cloud computing, a growing trend of various network attacks is presented, especially low-rate distributed denial of service (LDDoS) attacks. It has been pointed out by more and more network security research studies that LDDoS attacks are more prevalent and difficult to detect in the network, and it is recognized as one of the most major threats for network services [6, 7].

Taking advantage of the loopholes in the TCP protocol's retransmission timeout (RTO) mechanism, LDDoS sends periodically small pulse traffic separately by controlling multiple puppet machines, so it can reach the victim at the same time and converge into a huge impact traffic, causing the target host resources (such as bandwidth, memory, and CPU) to be exhausted and making the victim lose its ability to respond to the user's reasonable service request for a long-time [8]. Because of the characteristics of small traffic and concealment of LDDoS, when the MPTCP multipath transmission system suffers an LDDoS attack, the attack flow is difficult to be detected so that the defense capabilities of the transmission system will be affected and the robustness of the multi-path transmission system greatly reduced. Therefore, when the MPTCP transmission system is attacked by an LDDoS network, it is extremely important to improve the attack and defense capability of the system network.

The current academic research is mainly on improving the ability of detecting and defending LDDoS attacks from the perspective of extracting the characteristics of the attack flow. In recent years, it has been found by some scholars [9] that networks with different queue management algorithms have different defense capabilities when they are attacked by LDDoS. Queue management algorithms belong to link-based congestion control algorithms, which can be divided into passive queue management (PQM) and active queue management (AQM). By actively discarding or marking some data packets on the router side, network congestion can be effectively avoided and the performance of the TCP protocol can be improved by the queue management algorithm [10].

By tracking the research trends of MPTCP in academic circles, we find that there are insufficient research studies on the survivability analysis and robustness optimization of the MPTCP multipath transmission system. This paper introduces six network queue management algorithms: DropTail, random early detection (RED), fair random early detection (FRED), random exponential marking (REM), BLUE, and fair queuing (FQ). We compare and analyze their performance through simulation experiments. The conclusion of this paper provides a method for enhancing the robustness of the MPTCP network and provides effective suggestions for improving the queue management algorithm. Our research work in this paper is the first time to focus on the robustness of MPTCP from the perspective of AQM, hoping to attract the attention of relevant scholars through our research and promote the research on the adaptability and robustness of MPTCP congestion control.

The rest of this paper is organized as follows. The second part introduces the related research of LDDoS attack detection and defense and queue management mechanism. The third part will briefly introduce the basic ideas, advantages, and disadvantages of the six queue management algorithms. In the fourth part, the simulation experiment design and the evaluation of three performance indexes will be carried on. In the last part, we will give a summary.

2. Related Work

In recent years, various network attacks have shown a significant growth trend due to the large-scale deployment and application of emerging information technologies such as the IoT and cloud computing, especially LDDoS attacks [11]. It uses the weakness of the TCP protocol congestion control mechanism to periodically launch malicious attack traffic at a low rate, thereby reducing network throughput. Compared with traditional DDoS, LDDoS has a low attack rate and strong concealment. It is difficult to be discovered by traditional DDoS attack defense systems. LDDoS attacks can last longer, resulting in a rapid decline of network quality [12]. LDDoS is showing more and more serious harm to the network environment and has become a hot spot in the field of network security research at home and abroad. At present, many research institutions have carried out research on LDDoS attacks. In the network based on TCP protocol, the detection and defense of LDDoS have achieved a series of results.

For the research on LDDoS attacks, Kuzmanovic and Knightly [13] first proposed the concept of "Shrew" attacks. They collected relevant data about LDDoS attacks on the backbone network and conducted related research. After that, many researchers have proposed solutions for the detection and defense of LDDoS attacks by extracting LDDoS traffic characteristics or combining machine learning methods. Sahoo et al. [14] proposed a measurement method based on generalized entropy. According to the characteristics of Software-Defined Network (SDN) data flow, they used information distance to quantify the deviation of traffic under different probability distributions as a metric to detect attack behavior. Ren et al. [15] proposed a smart NCAP that supports LDDoS detection and proposed a method to detect LDDoS attack traffic using a linear multiple regression model with Simple Network Management Protocol (SNMP) content. Agrawal and Tapaswi [16] proposed a power spectral density (PSD)-based method to detect and mitigate LDDoS attacks in the frequency domain. This method can monitor and analyze real-time aggregated traffic for attack detection. Gu et al. [17] proposed a semisupervised clustering detection method with multiple characteristics, which can effectively detect DDoS attacks from massive data streams. Li et al. [18] proposed a DDoS detection model and defense system in a Software-Defined Network environment based on deep learning. The model can learn patterns from network traffic sequences, track network attack activities in a historical manner, and effectively clear DDoS attack traffic. de Lima Filho et al. [19] proposed a DoS detection system based on machine learning and inferred from signatures extracted from network traffic samples. Han et al. [20] proposed a cross-plane DDoS attack defense framework and detection mechanism in a Software Defined Network. The mechanism consists of a coarse-grained flow monitoring algorithm on the data plane and an attack classification algorithm based on fine-grained machine learning on the control plane. Kavitha and Padmavathi [21] developed an effective defense technology against LDoS attacks and proposed an advanced random time queue blocking (ARTQB) scheme. Lin et al. [22] proposed a "double check priority queue" structure, which can effectively reduce the impact of DDoS attacks so that ordinary users can still access the service. Yue et al. [23] found that the random early detection (RED) algorithm is vulnerable to LDoS attacks, which limits the sending rate of TCP senders. Wei et al. [9] analyzed and compared the defense capabilities of three classic queue management algorithms in ad hoc networks attacked by DDoS.

For the detection and defense of network attacks such as LDDoS, different scholars have conducted research from different perspectives. We find that there are few research results that combine queue management algorithms with LDDoS defense. Queue management algorithm is one of the research hotspots in the field of network congestion control. By actively discarding or marking some data packets on the router side, the queue management algorithm can effectively avoid network congestion and improve the performance of the TCP protocol. Researchers have proposed many improved techniques for queue management algorithms. Karmeshu and Bhatnagar [24] proposed an adaptive queue management mechanism with a random drop algorithm. Compared with the existing active queue management algorithm, it significantly improves system performance in terms of throughput, average queue size, utilization, and queue delay. Bisoy and Pattnaik [25] proposed a rate and queue-based active queue management (RQ-AQM) algorithm to improve the stability of network systems supporting TCP streams. Although these algorithms can be applied to various network conditions, most of the existing queue management algorithms are designed without considering their antiattack performance.

Based on the previous research on LDDoS defense methods and queue management algorithms, we introduced six classic queue management algorithms—DropTail, RED, Fred, REM, BLUE, and FQ. And we compare the performance of throughput, delay, and packet loss rate when MPTCP networks are under LDDoS attacks. This paper provides a solution for enhancing the defense capability of the MPTCP transmission system against LDDoS and other network attacks and enriches the theoretical results of the antiattack research of queue management algorithms.

3. Queue Management Algorithm

Currently, the phenomenon of network congestion can be seen everywhere, and the most effective way to solve network congestion is to manage the queues in the network. The queue management mechanism is mainly to control the network transmission node to buffer the transmission of information in the form of a queue. When the queue length reaches a critical value, the corresponding service message is discarded to achieve the purpose of controlling the queue length. Therefore, routers should manage the queues and maintain a small queue length, resulting in a series of queue management algorithms [26]. Current queue management algorithms can be divided into passive queue management algorithms and active queue management algorithms. The idea of PQM is that the queue management module only takes corresponding measures when the queue buffer overflows. The most commonly used in routers is the passive queue management algorithm DropTail. The idea of AQM is to make early judgments and take a series of measures before the queue buffer overflows so as to avoid the occurrence of congestion as soon as possible [27–29]. According to the design principle, the existing AQM algorithms can be divided into three types: queue length-based, network load-based, and both queue length and network load-based. This paper selects five AQM algorithms (RED, FRED, BLUE, REM, and FQ) belonging to different design principles. In addition, these AQM algorithms have been extensively studied by the academic community, and the experimental results have certain representativeness and reference value.

3.1. DropTail Algorithm. The DropTail algorithm is a typical passive queue management algorithm and the most widely used queue management algorithm in the network. The basic idea of the DropTail queue management algorithm is when a data message arrives at a network node, it needs to be queued in different output port buffers. Regardless of the length of its own queue, it will put the data message in the queue and wait to be sent. However, when the data flow is large, the queue length has exceeded the set buffer capacity value, and the network node has no space to temporarily store these new data messages [30]. Therefore, when the network is congested, all newly arrived data packets that are too late to be processed will be stored in the buffer, and these saved data packets will be processed when the system is idle. When the network continues to be congested, the buffer will be filled, and all newly arrived end packets will be discarded. When the sender detects that a data packet is discarded, it will reduce the data transmission rate until the congestion is eliminated.

The advantage of DropTail lies in its simple algorithm. Due to the simple way of processing data messages, it is supported by almost all network node platforms. However, the DropTail algorithm cannot avoid the occurrence of network congestion in advance, so there may be problems such as "global synchronization of TCP flows," continuous full queue status, and deadlock of service flows to the buffer, which affects the overall transmission speed and reduces network efficiency.

3.2. RED Algorithm. The RED algorithm [31] is a typical active queue management algorithm. The basic idea of the RED queue management algorithm is to judge congestion by monitoring the average length of the output port queue of the router. When the average length of the queue reaches a certain threshold, the router will randomly select some newly arrived packets to discard or mark and send a congestion notification. This algorithm can ensure that the sending window is reduced before the queue overflow causes packet loss, thereby reducing the sending rate and alleviating network congestion.

The RED queue management algorithm has two important computer mechanisms [32]. One is to calculate the average queue length and to predict congestion in advance by monitoring the average length of the buffer queue. The other is to calculate the drop probability P of data packets. If the average queue length is within the set threshold range, the arriving packets are discarded according to probability P.

In order to avoid unnecessary congestion control caused by sudden data, the RED queue management algorithm calculates the average queue length using an exponential weighted moving average algorithm; the formula is

$$L_{\text{now}} = (1 - w) \times L_{\text{first}} + w \times L, \tag{1}$$

where L is the current queue length, L_{first} is the previous estimated value of the average queue length, L_{now} is the current average queue length, w is the weighted coefficient of the current queue length, and its value range is between [0, 1].

At the same time, RED queue management mechanism needs to set two thresholds for the average queue length, which are the minimum threshold $L_{\min}$ and the maximum threshold $L_{\max}$. Comparing L_{now} with the threshold $L_{\min}$ and $L_{\max}$, the following operation is performed:

(1) If $L_{\text{now}} < L_{\min}$, all data packets are allowed to enter the queue

(2) If $L_{\text{now}} > L_{\max}$, all data packets are discarded

(3) If $L_{\min} \leq L_{\text{now}} \leq L_{\max}$, then calculate the transition packet loss probability p_a and discard the arriving data packets according to the probability P

In the RED queue management algorithm, the probability P of packet dropping by grouping is calculated according to the following formula:

$$P_a = P_{\max} \times \frac{\left(L_{\text{now}} - L_{\min}\right)}{\left(L_{\max} - L_{\min}\right)}. \tag{2}$$

$P_{\max}$ is the maximum packet loss rate. Obviously, the relationship between P_a, $P_{\max}$, $L_{\min}$, $L_{\max}$, and L_{now} is shown in Figure 2.

Then, the final packet loss probability is $P = (Pa/(1-\text{count} \times Pa))$, where count is the number of packets accepted since the last packet was dropped.

Compared with the DropTail algorithm, the RED algorithm controls the flow rate through its own congestion control, thereby avoiding the problems of long delay time and low link utilization caused by the long-time full queue state of the data receiving node. However, the RED algorithm has parameter sensitivity problems and cannot effectively control the cache size. In addition, the RED algorithm fails to consider different types of data streams on the network when calculating the packet loss probability, so it cannot effectively handle the congestion notification connections of different data streams, resulting in various connections sharing bandwidth unfairly and affecting network performance [33].

3.3. FRED Algorithm. The FRED algorithm belongs to the active queue management algorithm and is a new algorithm based on the improvement of the fairness of the RED queue management algorithm [34]. The FRED queue management algorithm uses accounting for each active flow to make different marking packet decisions for flows that use different bandwidths, thereby improving the fairness of different flows sharing bandwidth. The FRED algorithm is mainly based on the basic framework of the RED algorithm, so the algorithm also has two main parts: calculating the average queue length and calculating the probability of dropped packets, and the calculation formula is consistent with the RED algorithm. However, there is a big difference from the RED algorithm. It needs to recalculate the average queue length in the buffer when the packet arrives and leaves.

Compared with the RED algorithm, the FRED algorithm has more advantages in terms of fairness, and it effectively discriminates and restricts the nonadaptive data flow. However, because the FRED algorithm needs to record the active flows in the entire cache queue and maintain its corresponding flow state, when the number of flows is large, the router will be overloaded and computational overhead will increase.

3.4. REM Algorithm. The REM algorithm is one of the active queue management algorithms, which uses link prices to represent the network congestion metric. The basic idea of the REM queue management algorithm is to use the price concept to detect and control the congestion state of the network. The REM algorithm uses the cumulative sum of the

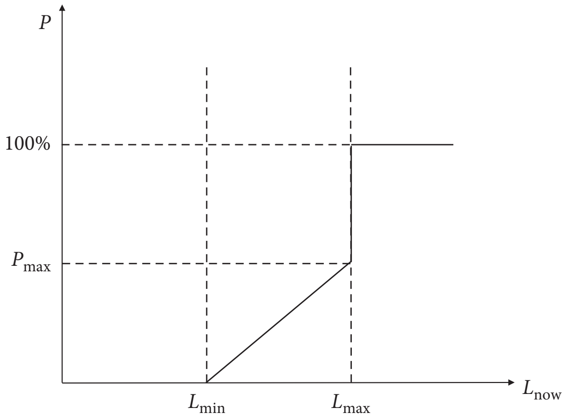


Figure 2: Average queue length and data packet drop probability of the RED algorithm.

price values of all connections on a channel as the congestion measure of this channel. And it embeds the metric value into the end-to-end packet marking probability that can be detected by the source so that the packet arrival rate matches the link bandwidth. Since calculating the data packet arrival rate needs to save certain state information, in order to avoid calculating the data packet arrival rate, the rate difference is approximated by the queue difference [35].

Price $P_l(t)$ of link l is calculated as follows:

$$P_l(t+1) = \left[P_l(t) + \gamma\left(b_l(t+1) - (1-\alpha_l)b_l(t) - \alpha_l b^*\right)\right]^+. \tag{3}$$

Among them, $\gamma > 0$, $\alpha_1 > 0$, $[z]^+ = \max\{0, z\}$, $b_l(t)$ is the instantaneous queue length of the queue of link l at time t, and b^* is the target queue length.

The marking probability of the queue of link l at time t is $m_l(t) = 1 - \phi^{-P_l(t)}$. ϕ is constant, and $\phi > 1$, and the end-to-end marking probability of the message is $1 - \phi^{-\Sigma_l P_l(t)}$. In practice, $\gamma = 0.001$, $\phi = 1.001$, $\alpha_1 = 0.1$, and $b^* = 20$.

The REM algorithm has opened up a new field for flow control, which can achieve the technical goal of AQM, but its performance is not ideal at present.

3.5. BLUE Algorithm. The BLUE algorithm is an active queue management algorithm based on network load. It uses packet loss events and links idle events to manage and notify congestion. The basic idea of the BLUE queue management algorithm is when the queue in the router overflows, the data packets will be continuously discarded. At this time, the BLUE algorithm will increase the probability of discarding the data packets and adjust the sending speed of the data packets. On the contrary, if the link is relatively idle at this time and the queue is empty, then the drop probability is reduced to increase the speed of sending data packets, thereby effectively controlling the speed of sending congestion notification information to improve the performance of the network [36].

The biggest advantage of the BLUE queue management algorithm is that a relatively small buffer can be used to complete congestion control, which improves the throughput of TCP streams and allows routers to have more free space. However, the BLUE algorithm also has a parameter setting problem. When the RTT of a data packet changes greatly or the number of connections suddenly changes, the set parameters will be invalid and the queue will fluctuate between packet loss and low usage.

3.6. FQ Algorithm. The FQ algorithm is an active management algorithm based on fair queues. The FQ algorithm establishes an independent output queue for each connection in the router. The router processes each queue in a round-robin manner to ensure fairness between each data flow. When a line is idle, the router scans all queues in turn and sends out the first packet of the queue each time. When a flow's data packets arrive too fast, its queue will quickly fill up, and new data packets belonging to this flow will be discarded [37].

With the FQ algorithm, it is impossible for each data stream to sacrifice other data streams and occupy more resources. In addition, it separates data streams so that data streams that do not comply with the congestion control mechanism will not affect other streams. So, it provides fairness without sacrificing statistical reuse.

4. Performance Evaluation

4.1. Experimental Design. In order to study the performance of six queue management algorithms when an MPTCP-enabled multihomed network suffers from LDDoS attacks, we develop a basic double dumbbell simulation topology with reasonable LDDoS attack traffic in NS-2 [38], as shown in Figure 3.

The router $R_{1,1}$ on path A is connected to five edge nodes that send UDP attack flows, and router $R_{1,2}$ is connected to five edge nodes that receive attack flows. We set the bandwidth between the node sending and receiving the attack stream and its connected router to 50 Mb and the propagation delay to 25 ms. The bandwidth between $R_{1,1}$ and $R_{1,2}$ and between $R_{2,1}$ and $R_{2,2}$ is set to 5 Mb, the propagation delay is 25 ms, and the queue management algorithm uses the DropTail algorithm. The total simulation time is 60 seconds. LDDoS attacks usually use the UDP protocol with constant bit rate (CBR) traffic to take a lot of bandwidth, so all attackers will generate UDP/CBR packets and start the attack after 2 s. The following three parameter values are used to describe the characteristics of LDDoS attacks [39]:

$$\text{LDDoS}(T, L, R) = \text{LDDoS}(100\text{ms}, 100\text{ms}, 1\text{Mbps}). \tag{4}$$

Among them, T is the attack period, L is the duration of the attack (the width of the attack pulse), and R is the strength of the attack pulse (the attack rate). If the parameter values of T, L, and R are set reasonably, the LDDoS traffic can reject the bandwidth of the regular TCP stream and avoid being detected by the DoS defense system. When the congestion control mechanism is triggered, the data packet will enter the timeout retransmission state. When an LDDoS attack occurs, the higher the R, the greater the bandwidth loss caused.

We choose the most commonly used DropTail algorithm in the simulation experiment to set the parameters of the best

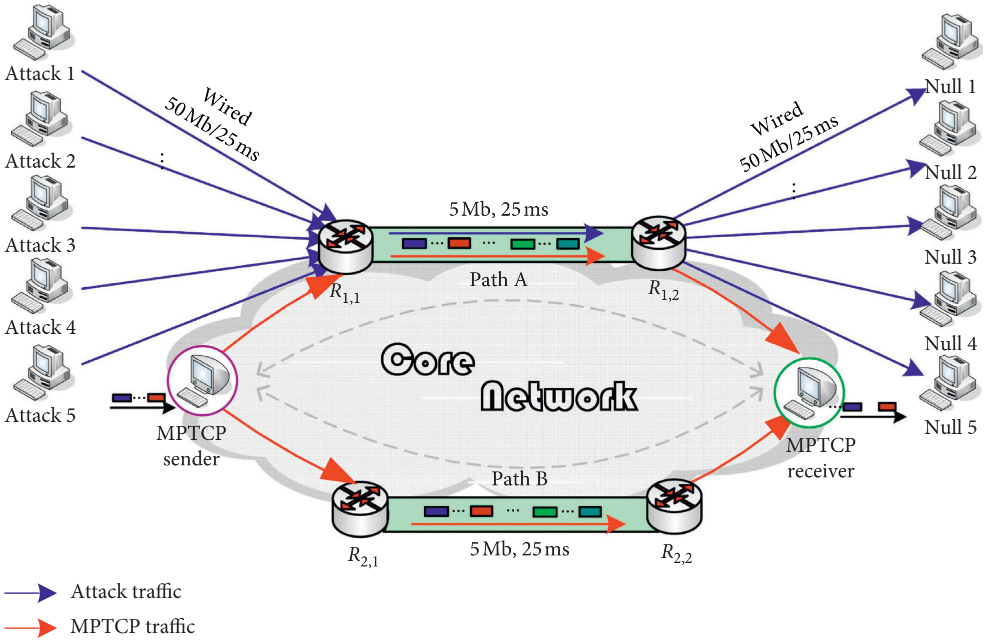


FIGURE 3: A basic dual-dumbbell simulation topology with LDDoS attacks.

attack flow and use the parameters of LDDoS as the fixed values in the comparison experiment. Figure 4 shows the congestion window (cwnd) size of path *A* when LDDoS is attacked and when it is not attacked. When an LDDoS attack is launched (after 2 s), the cwnd size of path *A* drops sharply. This is because the LDDoS attack can use MPTCP's RTO mechanism to make the MPTCP sender stay in the timeout retransmission state on path *A* and cannot exit. The congestion control mechanism of MPTCP is similar to that of TCP. In order to test the congestion of the network, the cwnd size is set to 1 at the initial slow start. As long as the sender judges that the network is congested, it is necessary to set the slow start threshold to half of the sender window value when congestion occurs (but not less than 2), then reset cwnd size to 1, and return to slow start stage. After the 20 s of simulation time in this experiment, the size of cwnd is maintained at 1. The network attack keeps the TCP data packet on path A in the timeout retransmission state, which shows that the LDDoS attack has achieved the best attack effect. This confirms that the DropTail algorithm, which is analyzed later, has a throughput of 0 for the normal TCP data flow on path *A* after the 20 s.

4.2. Simulation Analysis. Based on the MPTCP network, this paper analyzes and compares the defense capabilities of six classic network queue management algorithms such as DropTail, RED, FRED, REM, BLUE, and FQ when they are attacked by LDDoS. During the simulation, we test and analyze the performance of the throughput, end-to-end delay, and packet loss rate.

4.2.1. Comparison of the Throughput Performance. Throughput is the amount of successfully transmitted data per unit time. We measure the throughput between the sender and receiver when the MPTCP-enabled multihomed network is attacked by LDDoS. Figure 5 shows the comparison of the throughput performance of the MPTCP network transmission system using six queue management algorithms in a 60 s simulation time. For the convenience of observation, we plot the comparison of throughput performance when the LDDoS attack reaches stability, that is, after 20 s. Figure 6 tests the throughput of the MPTCP network transmission system (including path *A* and path *B*). Figure 7 tests the throughput of path A attacked by LDDoS.

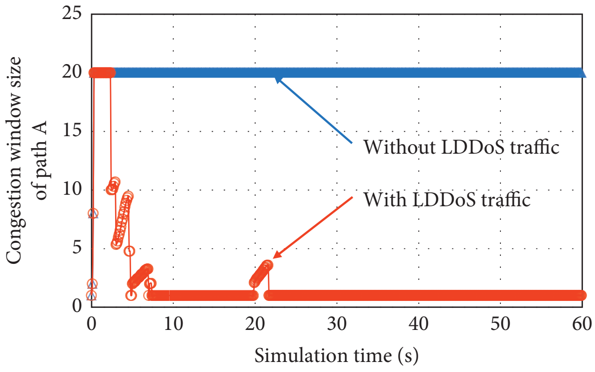


FIGURE 4: The congestion window size of path *A* with or without an LDDoS attack, respectively.

We can see that when path *A* is attacked by LDDoS, using the DropTail algorithm and the BLUE algorithm will lose the data transmission capability of the normal TCP stream after the 20 s. In addition, regardless of the single path or the entire transmission system, the FRED algorithm has the best throughput performance. This is because RED predicts congestion in advance by monitoring changes in the average length of the buffer queue so that data transmission nodes can control traffic speed through their own congestion control, thus avoiding low link utilization due to long periods of full queues.

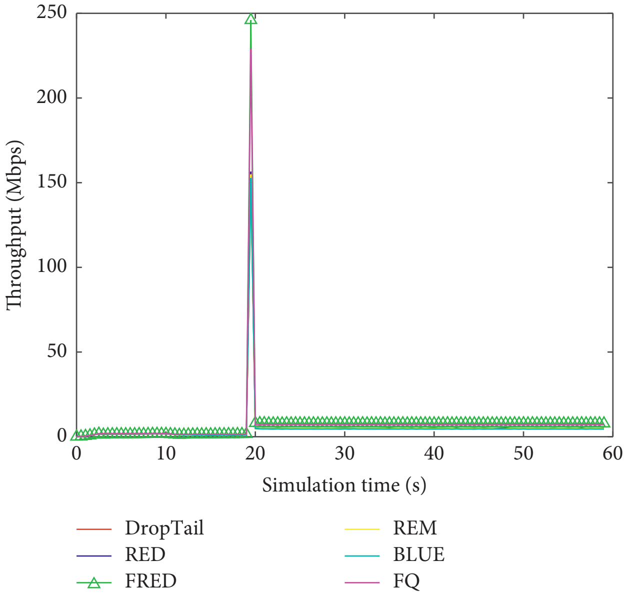


Figure 5: Comparison of the throughput performance of different queue management algorithms.

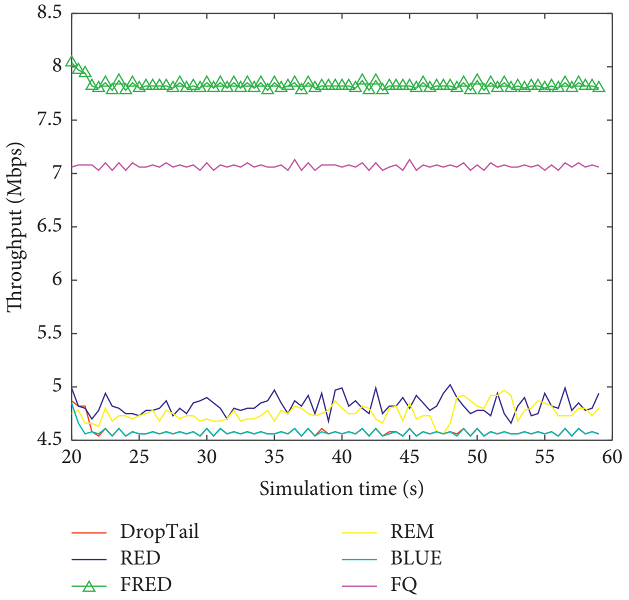


Figure 6: Comparison of the throughput performance of different queue management algorithms after 20 s.

In the MPTCP-enable multihomed networks, different queue management algorithms have different throughput performances when subjected to LDDoS attacks. We find that if the queue management algorithm cannot distinguish between different types of data streams, it is very likely that bad-behaving data streams will occupy a large number of data streams. From this, it can be seen that the queue management algorithm can improve the defense ability in the case of network attacks when considering the fairness of different types of data streams sharing bandwidth, which shows higher throughput performance. However, the RED algorithm fails to consider that the data streams on the network are of different types when calculating the packet loss probability, which leads to unfair sharing of bandwidth among various connections and affects network performance. The FRED algorithm improves the fairness of the RED algorithm. It makes different marking packet decisions by accounting for each active stream, thereby improving the fairness of different streams sharing bandwidth. When data streams with different competition capabilities compete for limited bandwidth, fairness ensures that the throughput performance of less competitive data streams will not suffer great damage, but a part of the transmission capacity is maintained. In addition, the FQ algorithm is slightly worse than FRED in throughput performance, but compared with the other four algorithms, because the fairness between different flows is also considered, when the network is under LDDoS attacks, it also retains a part of the throughput performance. The DropTail algorithm, as a typical passive queue management algorithm, cannot avoid network congestion in advance. When a sudden attack flow is encountered, the queue of the router will always be in a full state, a large number of TCP data flows will jointly slow down the sending rate to reduce congestion, and the utilization rate of the network will decrease accordingly. The REM algorithm and the BLUE algorithm also do not consider the fairness of different data streams sharing bandwidth, so when the number of router connections suddenly changes drastically, it will lead to poor throughput.

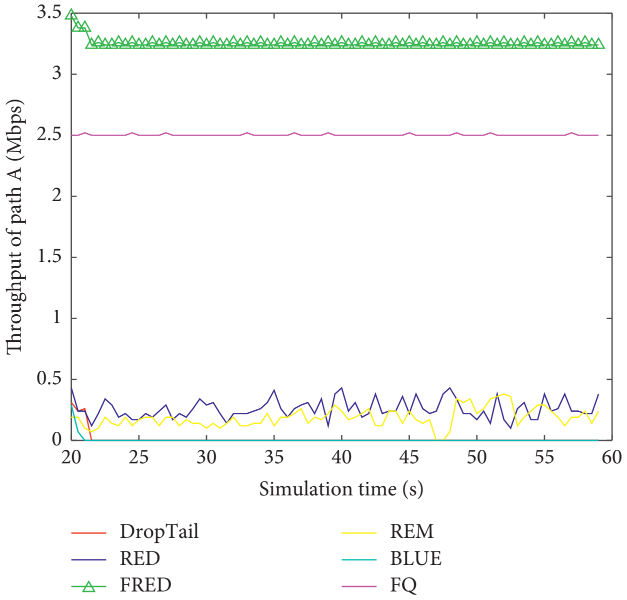


Figure 7: Comparison of the throughput performance of different queue management algorithms on path A after 20 s.

In the MPTCP-enabled multihomed networks, different queue management algorithms have different throughput performance when subjected to LDDoS attacks. If the queue management algorithm cannot distinguish between different types of data streams, it is very likely that bad-behaving data streams will occupy a lot of bandwidth. It can be seen that the queue management algorithm, when considering the fairness of different types of data streams sharing bandwidth, can improve the defense capability in the event of network attacks, which is manifested in higher throughput performance.

4.2.2. Comparison of the End-to-End Delay Performance. The end-to-end delay is the time required for a message or packet to be transmitted from one end of a network to another. It includes transmission delay, propagation delay, processing delay, and queuing delay. We test the delay of TCP data flow on path A. Figure 8 shows the comparison of the delay performance of the six queue management algorithms when the network is under LDDoS attacks. We mainly observe and analyze the comparison of delay performance when the network attack reaches a stable state (20 s).

From Figure 8, the delay performance of the RED algorithm is the best and the FRED performance is second. The RED algorithm can control the flow rate through its own congestion control, thereby avoiding the long delay time caused by the data receiving node due to the full queue state for a long time. The FRED algorithm considers the fairness issue more than the RED algorithm. It needs to record the active flow in the entire cache queue and maintain its corresponding flow state, which causes the router to be overloaded and increases the computational overhead. Therefore, in terms of delay performance, the FRED algorithm is slightly worse than the RED algorithm. The DropTail algorithm only sends a congestion signal to the router or the sender when the queue is full, resulting in prolonged queuing time of data packets in the queue and increased end-to-end delay. In addition, from Figure 8, we find that the time delay data of the DropTail algorithm disappear after 25 s. This is because the TCP stream on path A has been severely attacked by LDDoS and has been in a timeout retransmission state, so the delay of this type of data stream cannot be calculated. Although the BLUE algorithm adjusts the sending speed of data packets when the queue overflows in the router, when the router is attacked by a network such as LDDoS, the set parameters will become invalid. The delay of the REM algorithm shows obvious fluctuations because the algorithm's operating mechanism is to match the data packet arrival rate with the link bandwidth. When subjected to periodic LDDoS attacks, the data packet transmission rate will also change accordingly. We find that the delay of the FQ algorithm is in the middle of the delay of these six algorithms and presents a horizontal straight line. This is consistent with its design philosophy of ensuring fairness between each flow and allowing routers to process each queue in a polling manner.

In the MPTCP-enabled multihomed network, different queue management algorithms have different delay performances when subjected to LDDoS attacks. The queue management algorithm can adjust the flow rate by optimizing the congestion control mechanism and avoiding the long delay time caused by the long-time full queue state of the data receiving node, thereby improving the defense ability in the case of network attacks.

4.2.3. Comparison of the Packet Loss Rate Performance. The packet loss rate refers to the ratio of the number of data packets lost in the test to the data group sent. We test the packet loss of TCP data flow on path A. Table 1 shows the detailed data of the total number of packets, the number of lost packets, and the packet loss rate of the six queue management algorithms when the network is under LDDoS attacks. It can be seen from Figure 9 that the FQ algorithm has no packet loss during the entire data transmission process. The packet loss rates of the FRED, DropTail, BLUE, and REM algorithms are 0.02%, 2.68%, 3.15%, and 4.22%, respectively. The packet loss rate of RED is the highest, with a packet loss rate of 5.23%.

In a network environment, it is entirely possible that an application does not use the TCP protocol. The LDDoS attack flow can bypass the end-to-end congestion control mechanism and send its own data packets to the router arbitrarily, causing normal application data packets to be discarded. The FQ algorithm solves this problem. In the FQ algorithm, the router has a queue for each output line. The router processes packets in a "polling" manner to ensure fairness between each flow. Therefore, the packet loss rate using the FQ algorithm is low. However, when data packets of a flow arrive too fast, its queue will quickly fill up, and new data packets belonging to this flow will also be discarded. Although the RED algorithm proposes a method to deal with sudden data flow, it uses an exponentially weighted moving average algorithm to make the average queue length change relatively slowly, but because the algorithm has the disadvantage of parameter sensitivity, the parameters (such as the maximum threshold $L_{\max}$) cannot be modified in time, resulting in a large number of packets being discarded. Compared with the RED algorithm, the FRED algorithm recalculates the average queue length in the buffer when the packet arrives and leaves. It can more timely and accurately reflect the queue changes and modify the parameters, so the packet loss rate is very low. We find that when DropTail, BLUE, and REM algorithms are attacked by LDDoS and other network attacks, more data packets will be discarded by the queue, which reduces the efficiency of the network.

It can be seen that when it is subjected to network attacks such as LDDoS, the queue management algorithm should have the ability to adjust parameters in a timely and accurate manner so as to effectively ensure the transmission of normal TCP data streams. In addition, improving the fairness of the algorithm can also reduce the packet loss rate and show better transmission performance.

The results show that in the MPTCP-enabled multihomed networks, different queue management algorithms have different throughput, delay, and packet loss rate performance when subjected to LDDoS attacks. In terms of throughput performance, considering fairness, the FRED algorithm has the best performance and the FQ algorithm has the second-highest performance. In view of delay performance, the RED algorithm is the best, and the performance of FRED is slightly worse than that of RED. However, with the development of technology, the small delay gap can

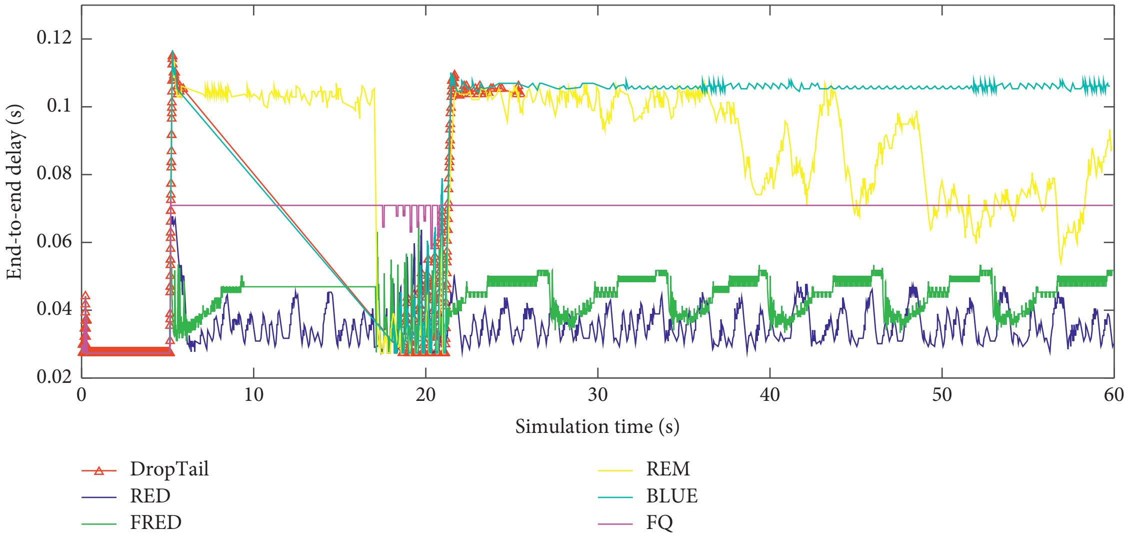


Figure 8: Comparison of the end-to-end delay performance of different queue management algorithms.

Table 1: Packet loss data of different queue management algorithms.

Queue management algorithm	Total number of packages	Number of lost packets	Packet loss rate (%)
DropTail	1046	28	2.68
RED	2014	110	5.23
FRED	16012	4	0.02
BLUE	1112	35	3.15
REM	1894	80	4.22
FQ	12798	0	0.00

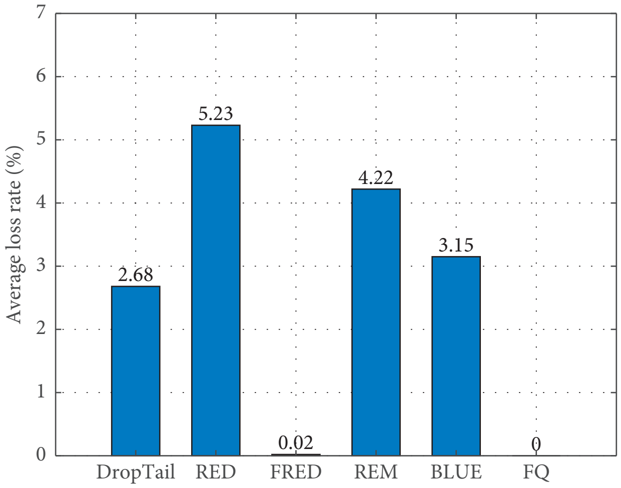


Figure 9: Comparison of the packet loss rate performance of different queue management algorithms.

be made up by increasing the operating speed of hardware devices. In consideration of packet loss rate performance, the FQ and FRED algorithms can maintain a lower packet loss rate when subjected to network attacks of LDDoS compared with other algorithms. Through an overall consideration of the three performance indicators of throughput, delay, and packet loss rate, it is evident that the FRED algorithm has better performance.

5. Conclusion

This paper introduces six queue management algorithms: DropTail, RED, FRED, REM, BLUE, and FQ. Through simulation experiments, we compare the performance of different queue management algorithms in the MPTCP network under LDDoS attack. The results show that in the multihost network using MPTCP, when one of the paths is attacked by LDDoS, the other paths can still transmit normally and the whole system will not collapse. Different queue management algorithms have different throughput, latency, and packet loss rates. Through an overall consideration of the three performance indicators of throughput, delay, and packet loss rate, it is evident that the FRED algorithm has better performance. By adopting an effective queue management algorithm, the MPTCP transmission system can enhance its robustness and defense capability, thus improving transmission performance. In addition, our research conclusions provide effective suggestions for the technical improvement of the queue management algorithm. In the future, the antiattack performance of the algorithm should be taken into consideration when designing and improving the queue management algorithm. An effective queue management algorithm should achieve three aspects: (i) it can adjust the traffic speed by optimizing the congestion control mechanism; (ii) the fairness of different types of data streams sharing bandwidth is taken into consideration; and (iii) it has the ability to adjust the parameters of the queue management algorithm in a timely and accurate manner, thereby effectively ensuring the transmission performance of normal TCP data streams so as to improve the defense capability against network attacks.

Data Availability

No data were used to support this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under grant no. 61962026 and the Natural Science Foundation of Jiangxi Province under grant no. 20192ACBL21031.

References

[1] M. R. Palash and K. Chen, "MPWiFi: synergizing MPTCP based simultaneous multipath access and WiFi network performance," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 142–158, 2020.

[2] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, "TCP extensions for multipath operation with multiple addresses," *Internet Engineering Task Force (IETF) RFC 8684*, 2020.

[3] Y. Cao, M. Collotta, S. Xu et al., "Towards adaptive multipath managing: a lightweight path management mechanism to aid multihomed mobile computing devices," *Applied Sciences-Basel*, vol. 10, no. 1, pp. 1–18, 2020.

[4] F. Song, Z. Ai, Y. Zhou, I. You, K.-K. R. Choo, and H. Zhang, "Smart collaborative automation for receive buffer control in multipath industrial networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1385–1394, 2020.

[5] Y. Cao, J. Chen, Q. Liu, G. Lei, H. Wang, and I. You, "Can multipath TCP be robust to cyber attacks with incomplete information?" *IEEE Access*, vol. 8, pp. 165872–165883, 2020.

[6] X. Zhao, H. Peng, X. Li et al., "Defending application layer DDoS attacks via multidimensional parallelotope," *Security and Communication Networks*, vol. 2020, Article ID 6679304, 11 pages, 2020.

[7] L. Zhou, M. Liao, C. Yuan, and H. Zhang, "Low-rate DDoS attack detection using expectation of packet size," *Security and Communication Networks*, vol. 2017, Article ID 3691629, 14 pages, 2017.

[8] Z. Wu, Q. Xu, J. Wang et al., "Low-rate DDoS attack detection based on factorization machine in software defined network," *IEEE Access*, vol. 8, pp. 17404–17418, 2020.

[9] W. Wei, H. Song, H. Wang, and X. Fan, "Research and simulation of queue management algorithms in Ad Hoc networks under DDoS attack," *IEEE Access*, vol. 5, pp. 27810–27817, 2017.

[10] C. A. Gomez, X. Wang, and A. Shami, "Federated intelligence for active queue management in inter-domain congestion," *IEEE Access*, vol. 9, pp. 10674–10685, 2021.

[11] M. V. Kieu, D. T. Nguyen, and T. T. Nguyen, "A way to estimate TCP throughput under low-rate DDoS attacks: one TCP flow," in *Proceedings of the 2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pp. 1–8, Ho Chi Minh, Vietnam, October 2020.

[12] Z. Li, H. Jin, D. Zou, and B. Yuan, "Exploring new opportunities to defeat low-rate DDoS attack in container-based cloud environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 3, pp. 695–706, 2020.

[13] A. Kuzmanovic and E. Knightly, "Low-rate TCP-targeted denial of service attacks (the shrew vs. the mice and elephant)," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)*, vol. 33, no. 4, pp. 75–86, Karlsruhe, Germany, August 2003.

[14] K. S. Sahoo, D. Puthal, M. Tiwary, J. J. P. C. Rodrigues, B. Sahoo, and R. Dash, "An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics," *Future Generation Computer Systems*, vol. 89, pp. 685–697, 2018.

[15] J. Ren, Y. Liu, J. Wu, J. Li, and K. Wang, "Smart NCAP supporting low-rate DDoS detection for IEEE 21451-1-5 internet of things," in *Proceedings of the 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, pp. 532–535, Taipei, Taiwan, May 2019.

[16] N. Agrawal and S. Tapaswi, "Low rate cloud DDoS attack defense method based on power spectral density analysis," *Information Processing Letters*, vol. 138, pp. 44–50, 2018.

[17] Y. Gu, Y. Wang, Z. Yang et al., "Multiple-features-based semi-supervised clustering DDoS detection method," *Mathematical Problems in Engineering*, vol. 2017, Article ID 5202836, 10 pages, 2017.

[18] C. Li, Y. Wu, X. Yuan et al., "Detection and defense of DDoS attack based on deep learning in OpenFlow-based SDN," *International Journal of Communication Systems*, vol. 31, no. 5, Article ID e3497, 2018.

[19] F. S. de Lima Filho, F. A. F. Silveira, A. D. M. Brito Jr. et al., "Smart detection: an online approach for DoS/DDoS attack detection using machine learning," *Security and Communication Networks*, vol. 2019, Article ID 1574749, 15 pages, 2019.

[20] B. Han, X. Yang, Z. Sun et al., "OverWatch: a cross-plane DDoS attack defense framework with collaborative intelligence in SDN," *Security and Communication Networks*, vol. 2018, Article ID 9649643, 15 pages, 2018.

[21] R. Kavitha and G. Padmavathi, "Advanced random time queue blocking for effective protection of application servers against low-rate DoS attacks," *International Journal of Network Security*, vol. 19, no. 6, pp. 1024–1035, 2017.

[22] C. Lin, H. Lin, T. Wu et al., "Preserving quality of service for normal users against DDoS attacks by using double check priority queues," *Journal of Ambient Intelligence and Humanized Computing*, vol. 4, no. 2, pp. 275–282, 2013.

[23] M. Yue, Z. Wu, and J. Wang, "Detecting LDoS attack bursts based on queue distribution," *IET Information Security*, vol. 13, no. 3, pp. 285–292, 2019.

[24] S. P. Karmeshu and S. Bhatnagar, "Adaptive mean queue size and its rate of change: queue management with random dropping," *Telecommunication Systems*, vol. 65, no. 2, pp. 281–295, 2017.

[25] S. K. Bisoy and P. K. Pattnaik, "RQ-AQM: a rate and queue-based active queue management using feedback control theory," *International Journal of Communication Networks and Distributed Systems (IJCNDS)*, vol. 21, no. 2, pp. 266–295, 2018.

[26] S. K. Mohapatra, S. K. Bisoy, and P. K. Dash, "Stability analysis of active queue management techniques," in *Proceedings of the*

2015 International Conference on Man and Machine Interfacing (MAMI), pp. 1–6, Bhubaneswar, India, December 2015.

[27] R. Al-Saadi, G. Armitage, J. But, and P. Branch, “A survey of delay-based and hybrid TCP congestion control algorithms,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3609–3638, 2019.

[28] Y. Liu, X. Liu, Y. Jing, Z. Zhang, and X. Chen, “Congestion tracking control for uncertain TCP/AQM network based on integral backstepping,” *ISA Transactions*, vol. 89, pp. 131–138, 2019.

[29] M. Khatari, A. A. Zaidan, B. B. Zaidan, O. S. Albahri, and M. A. Alsalem, “Multi-criteria evaluation and benchmarking for active queue management methods: open issues, challenges and recommended pathway solutions,” *International Journal of Information Technology & Decision Making*, vol. 18, no. 4, pp. 1187–1242, 2019.

[30] P. K. Sahu and B. M. Acharya, “Performance analysis of unicasting routing protocols for mobile ad-Hoc network,” in *Proceedings of the 2019 International Conference on Applied Machine Learning (ICAML)*, pp. 286–290, Bhubaneswar, India, May 2019.

[31] A. Pandey, T. Anand, M. Shah, and M. P. Tahiliani, “Adaptive RED for FreeBSD: design, implementation and challenges,” in *Proceedings of the 2019 IEEE Region 10 Conference (TENCON 2019)*, pp. 2340–2344, Kochi, India, Octorber 2019.

[32] S. Patel, “Performance analysis of RED for stabilized queue,” in *Proceedings of the 2014 Seventh International Conference on Contemporary Computing (IC3)*, pp. 306–311, Noida, India, August 2014.

[33] S. Simaiya, A. Shrivastava, and N. P. Keer, “IRED algorithm for improvement in performance of mobile Ad Hoc networks,” in *Proceedings of the 2014 Fourth International Conference on Communication Systems and Network Technologies*, pp. 283–287, Bhopal, India, April 2014.

[34] H. Wu, F. Ren, D. Mu, and W. Pan, “Utilizing TTL to enhance TCP fairness,” in *Proceedings of the 2007 Second International Conference on Communications and Networking in China*, pp. 208–212, Shanghai, China, August 2007.

[35] S. Patel, “Performance analysis and modeling of congestion control algorithms based on active queue management,” in *Proceedings of the 2013 International Conference on Signal Processing and Communication (ICSC)*, pp. 449–454, Noida, India, December 2013.

[36] Y. Irawan and N. Surantha, “Performance evaluation of queue algorithms for video-on-demand application,” in *Proceedings of the 2020 International Conference on Information Management and Technology (ICIMTech)*, pp. 966–971, Bandung, Indonesia, August 2020.

[37] H. Attar, M. R. Khosravi, S. S. Igorovich et al., “Review and performance evaluation of FIFO, PQ, CQ, FQ, and WFQ algorithms in multimedia wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 16, no. 6, Article ID 15501477209, 2020.

[38] Y. Cao, F. Song, Q. Liu, M. Huang, H. Wang, and I. You, “A LDDoS-aware energy-efficient multipathing scheme for mobile cloud computing systems,” *IEEE Access*, vol. 5, pp. 21862–21872, 2017.

[39] M. Baskar, J. Ramkumar, C. Karthikeyan et al., “Low rate DDoS mitigation using real-time multi threshold traffic monitoring system,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 2021, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5593363, 10 pages
https://doi.org/10.1155/2021/5593363

Research Article

Provably Secure Security-Enhanced Timed-Release Encryption in the Random Oracle Model

Ke Yuan,[1,2] **Yahui Wang**,[1,2] **Yingming Zeng**,[3] **Wenlei Ouyang**,[1] **Zheng Li**,[1] **and Chunfu Jia**[4]

[1] *School of Computer and Information Engineering, Henan University, Kaifeng 475004, China*
[2] *International Joint Research Laboratory for Cooperative Vehicular Networks of Henan, Henan University, Kaifeng 475004, China*
[3] *Beijing Institute of Computer Technology and Applications, Beijing 100854, China*
[4] *College of Cybersecurity, Nankai University, Tianjin 300350, China*

Correspondence should be addressed to Zheng Li; lizheng@henu.edu.cn

Received 19 February 2021; Accepted 14 May 2021; Published 27 May 2021

Academic Editor: Hao Peng

Cryptographic primitive of timed-release encryption (TRE) enables the sender to encrypt a message which only allows the designated receiver to decrypt after a designated time. Combined with other encryption technologies, TRE technology is applied to a variety of scenarios, including regularly posting on the social network and online sealed bidding. Nowadays, in order to control the decryption time while maintaining anonymity of user identities, most TRE solutions adopt a noninteractive time server mode to periodically broadcast time trapdoors, but because these time trapdoors are generated with fixed time server's private key, many "ciphertexts" related to the time server's private key that can be cryptanalyzed are generated, which poses a big challenge to the confidentiality of the time server's private key. To work this out, we propose a concrete scheme and a generic scheme of security-enhanced TRE (SETRE) in the random oracle model. In our SETRE schemes, we use fixed and variable random numbers together as the time server's private key to generate the time trapdoors. We formalize the definition of SETRE and give a provably secure concrete construction of SETRE. According to our experiment, the concrete scheme we proposed reduces the computational cost by about 10.8% compared to the most efficient solution in the random oracle model but only increases the almost negligible storage space. Meanwhile, it realizes one-time pad for the time trapdoor. To a large extent, this increases the security of the time server's private key. Therefore, our work enhances the security and efficiency of the TRE.

1. Introduction

Cryptographic primitive of timed-release encryption (TRE) [1, 2] requires the sender to set a specified time for the designative receiver to decrypt the secret message. With TRE, the sender encrypts a message and then sends to the receiver; before the decrypt time that the sender has set arrives, no one can decrypt this ciphertext. With the efforts of many distinguished scholars, TRE has developed into a basic cryptographic primitive, which can be combined with many other cryptographic primitives and applied to different fields, such as regularly posting on the social network [3, 4], edge caching [5], and ciphertext retrieval [6, 7].

According to the latest research, the TRE constructions have been extended from the mathematical problems [8–28] to the physical problems [29, 30] and the blockchain approach [31–34]. At present, a large number of TRE constructions are based on the mathematical problems. In practical terms, the most commonly used model is the noninteractive time server model. In this model, for the time server, neither the sender nor the receiver of the message interacts with it. The time server periodically broadcasts the time trapdoor. The receiver chooses the time trapdoor corresponding to the decryption time of the ciphertext to complete the decryption at the designated time.

However, in the current noninteractive TRE schemes, many time trapdoors related to the time server's private key will be generated. This will cause the attacker to have a certain amount of pairs (time, time trapdoor). Although the problems related to bilinear pairing are difficult to solve, the attacker can still adopt chosen-plaintext attack (CPA) or chosen-ciphertext attack (CCA) to attack the system, which seriously challenges the security of the private key of the time server. Thus, in this paper, we are working on this problem and trying to construct a solution.

1.1. Related Work. TRE was first proposed by May [2] in 1993 and then discussed in detail by Rivest et al. [1] in 1996. Most of the previous schemes can be divided into time-lock puzzles [1, 15, 18, 33] and agents categories. Most agent-class schemes use the time server as the agent, which are divided into interactive model [1, 8, 23, 24] and noninteractive model [9–14, 16, 17, 19–22, 25, 26, 28]. The time server method was originally constructed based on the quadratic residue problem [8]. After that, most of the proposed solutions are based on the assumption of the difficult bilinear pairing class problems, such as bilinear Diffie–Hellman (BDH) assumption [9–11, 13, 19, 21, 22, 24–28], bilinear Diffie–Hellman inversion (BDHI) assumption [12], and bilinear Diffie–Hellman exponent (BDHE) assumption [17].

In the solutions of the noninteractive server model, the time server's private key is used to perform an encryption-like operation on the hash function value of a time point T to generate a corresponding time trapdoor. Therefore, this model produces many pairs (plaintext, ciphertext) related to the private key of the time server. In response to this problem, we need to construct a new solution.

1.2. Our Contributions. We reexamine the noninteractive time server model in which the time server's private key is repeatedly used, resulting in many pairs (plaintext, ciphertext) related to the private key of the time server. In order to solve this problem, we construct a security-enhanced timed-release encryption (SETRE) solution based on the BDH assumption.

As we all know, in the operations of encryption and decryption, we use the private key k to encrypt the plaintext M and get the ciphertext $C = E_k(M)$ and use the private key k to decrypt the ciphertext C and get the plaintext $M = D_k(C)$. Similarly, we let the private key s and the hash function value $H(T)$ of a time point T perform some operations together to generate the corresponding time trapdoor $S_T = E_s(H(T))$; correspondingly, we can get $H(T) = D_s(S_T)$. In the above statement, S_T is equivalent to the ciphertext C, and $H(T)$ is equivalent to the plaintext M. If the attacker has many pairs (plaintext, ciphertext), then the security of the time server's private key will be greatly threatened.

Our SETRE schemes include a concrete scheme and a generic scheme. In our SETRE, the time server will use a random number x as the time server's session private key every time before publishing the time trapdoor. This session private key is combined with the time server's fixed private key to generate the time trapdoor $S_T = E_{(s,x)}(H(T))$ of our SETRE. Therefore, in our SETRE schemes, the secret private key involved in every generated time trapdoor is different. So, we can claim that our schemes realize one-time pad for the time trapdoor. In this case, the attacker can only get a pair of (plaintext, ciphertext) about the time point and its time trapdoor at most. Even if the attacker successfully obtains the private key of the time server corresponding to a time trapdoor, he cannot get the private key of the time server corresponding to other time trapdoors so that the time trapdoor cannot be generated in advance, which ensures that the receiver cannot decrypt in advance.

1.3. Organization. We begin by explaining what is SETRE. In Section 2, we give some cryptographic background and our generic public key encryption scheme. In Section 3, we formally define our SETRE and its simulation security game model. In Section 4, we present the concrete construction of SETRE and give its provably secure proof and the efficiency analysis. In Section 5, we provide the formal definition and construction of the generic SETRE and give its security analysis and efficiency analysis. Finally, we give the conclusion and future work.

2. Preliminary

We give a brief review of the bilinear pairing property, BDH assumption, and our generic public key encryption scheme that needs to be known in this section.

2.1. Properties of Bilinear Pairings. We give a form of bilinear pairings and their properties as described below.

Definition 1. Let G_1 be an elliptic curve discrete logarithm problem (ECDLP) additive group over a finite field, G_2 be a discrete logarithm problem (DLP) multiplicative group over a finite field, and the order of G_1, G_2 be a prime number q. The mapping e: $G_1 \times G_2 \longrightarrow G_2$ is a bilinear pairing mapping if e satisfies

(1) Bilinear property: given any $P, Q, R \in G_1$, the following operations hold:

$$\begin{aligned} e(P+Q,R) &= e(P,R)e(Q,R), \\ e(P,Q+R) &= e(P,Q)e(P,R). \end{aligned} \tag{1}$$

(2) Nondegeneracy: suppose that the generator of group G_1 is P, then the generator of group G_2 is $e(P,P)$.

(3) Computability: given any two elements $P, Q \in G_1$, there must be an effective algorithm for calculating $e(P,Q)$.

2.2. BDH Assumption. Many cryptographic schemes are based on various difficult assumptions related to bilinear pairs, such as the (D)BDH assumption, (D)BDHI assumption, and (D)BDHE assumption [35, 36]. We now give the definition of the BDH assumption used in our SETRE schemes as follows.

Definition 2. Let G_1 be an ECDLP additive group over a finite field, P be the generator of G_1, G_2 be a DLP multiplicative group over a finite field, and the order of G_1, G_2 be a prime number q. Given $P, aP, bP, cP \in G_1^*$ (a, b, and c are evenly distributed in $\mathbb{Z}_q^*$), calculate $e(P,P)^{abc} \in G_2^*$. If $\Pr[\mathscr{A}(P, aP, bP, cP) = e(P,P)^{abc}] \geq \mathscr{E}$, then the advantage of the adversary $\mathscr{A}$ to solve the BDH assumption is $\mathscr{E}$, and $\mathscr{E}$ is negligible.

2.3. General Public Key Encryption Scheme. We simplify and abstract public key encryption (which has a certain characteristic) and only keep three phases which are initialization, encryption, and decryption; then, the general public key encryption (GPKE) scheme can be obtained.

Definition 3. $\mathscr{E}_{\text{GPKE}} = (\text{Setup}, \text{Enc}, \text{Dec})$ is the public key encryption algorithm, where

Setup: generates system public parameters and the user's public key and private key pairs $(\text{upk}, \text{usk}) = (uP, u)$ in which $u \in \mathbb{Z}_q^*$, $P \in G_1$ is a generator of G_1, and G_1 is an additive group

Enc: uses the user's public key uP to encrypt the plaintext to get the ciphertext $C_{\text{GPKE}} = \text{Enc}(M, uP)$

Dec: uses the user's private key u to decrypt the ciphertext to get the plaintext $M = \text{Dec}(C_{\text{GPKE}}, u)$

3. SETRE: Definitions

Suppose Bob is a social network user, and he wants to upload documents scheduled to be published regularly to the social network platform in advance so that he can pay attention to other things without worrying about this matter. And Bob does not want the social network platform to know in advance what he wants to publish. In this application scenario, Bob can use our SETRE solution to solve this problem securely and efficiently. Bob sends the following ciphertext of the document in advance with the designated decryption time:

$$C = \text{Enc}\left(M, ts_{\text{pub}}, ts_{\text{spub}}, \text{upk}, r, T\right), \tag{2}$$

where M is one of the documents planned to be released at a designated time point in the future, ts_{pub} and ts_{spub} are the time server's fixed public key and session public key, respectively, upk is the receiver's public key, r is a random number as a factor of freshness, and T is the designated decryption time. The social network platform can obtain the ciphertext of the document in advance but can only decrypt it in the future after the predetermined decryption time has arrived. We call such a cryptographic scheme noninteractive SETRE.

Definition 4. Our noninteractive concrete SETRE scheme includes three entities which are time server, sender, and receiver and polynomial-time randomized algorithm 7 tuples $\mathscr{E}_{\text{SETRE}} = (\text{Setup}, \text{TS\_KeyGen}, \text{User\_KeyGen}, \text{Enc}, \text{ST\_Rel}, \text{ST\_Rel}, \text{Dec})$, where

Setup: generates a public parameter params from a security parameter

TS_KeyGen: calculates and generates the fixed public/private key pair $(ts_{\text{pub}}, ts_{\text{priv}})$ and the session public/private key pair $(ts_{\text{spub}}, ts_{\text{spriv}})$ of the time server

User_KeyGen: calculates and generates the public/private key pair (upk, usk) of the system user

Enc: calculates the ciphertext C of the plaintext M, by using the public keys $ts_{\text{pub}}, ts_{\text{spub}}$, and upk, and a designated decryption time point T

ST_Rel: calculates a time server's time trapdoor S_T, by using the time server's fixed private key ts_{priv}, a designated decryption time point T, and its corresponding session private key ts_{spriv}

UT_Rel: calculates a user's time trapdoor U_T, by using the receiver's private key usk and a designated decryption time point T

Dec: calculates a plaintext M, by using a ciphertext C, the time server's time trapdoor S_T, and the receiver's time trapdoor U_T; or outputs a "reject" message

We use the simulation security game between the adversary $\mathscr{A}$ and the challenger $\mathscr{B}$ to formally define the security against the active adversary $\mathscr{A}$. The specific formal definition is as follows:

Preparation: public parameters are generated by the system.

Initialization: a pair of designated decryption time points T_0^* and T_1^* to be challenged is selected by the adversary $\mathscr{A}$.

Setup: the public parameters params and public keys upk, ts_{pub}, and ts_{spub} are generated by the challenger $\mathscr{B}$ and sent to the adversary $\mathscr{A}$.

Phase 1: the adversary $\mathscr{A}$ performs m queries of $q_1, q_2, \ldots, q_m$, where query q_i is one of the following:

(1) At any time point, the adversary $\mathscr{A}$ can perform queries of the random oracles H_1 and H_2. In response to H_1 and H_2 queries, the challenger $\mathscr{B}$ keeps two lists of H_1-list and H_2-list.

(2) Time trapdoor queries: time trapdoor query S_{T_i} and U_{T_i} of T_i where $T_i \notin \{T_0^*, T_1^*\}$. The challenger $\mathscr{B}$ responds by running algorithm ST_Rel and UT_Rel to generate the time trapdoors S_{T_i} and U_{T_i} corresponding to the designated decryption time point T_i. The challenger $\mathscr{B}$ then sends S_{T_i} and U_{T_i} to the adversary $\mathscr{A}$.

(3) Decryption queries: decryption query (C_i, T_i) for the designated decryption time point T_i. To decrypt the ciphertext (C_i, T_i), $\mathscr{B}$ runs algorithm Dec and uses the time trapdoors S_{T_i} and U_{T_i}. The challenger $\mathscr{B}$ then sends the decrypted plaintext M to the adversary $\mathscr{A}$.

These queries can be adaptive, which means that the response of q_i can be determined based on the responses of $q_1, q_2, \ldots, q_{i-1}$ previously queried.

Challenge: a pair of designated decryption time points T_0^* and T_1^* to be challenged is selected by the adversary $\mathscr{A}$. The challenger $\mathscr{B}$ selects a random bit $\flat \in \{0, 1\}$, sets the ciphertext to be $(C_\flat^*, T_\flat^*)$, and then sends the challenge ciphertext $(C_\flat^*, T_\flat^*)$ to $\mathscr{A}$.

Phase 2: the adversary $\mathscr{A}$ performs other queries of $q_{m+1}, \ldots, q_{\text{num}}$, and the challenger $\mathscr{B}$ responds as shown in Phase 1.

Guess: in the end, the adversary $\mathscr{A}$ outputs a guess of $\flat' \in \{0, 1\}$. If $\flat = \flat'$, then $\mathscr{A}$ wins the simulation security game.

We call such an adversary $\mathscr{A}$ an IND-sT-CCA adversary, and we can formally define the advantages of $\mathscr{A}$ attack our concrete SETRE scheme $\mathscr{E}$ as

$$\mathrm{Adv}_{\mathscr{E},\mathscr{A}}^{\mathrm{CCA}} = \left| P_r\left[\flat = \flat'\right] - \frac{1}{2} \right|. \tag{3}$$

Definition 5. Our concrete SETRE scheme $\mathscr{E}$ is said to be $(t, q_{H_2}, q_T, q_C, \varepsilon)$-selective designated decryption time, adaptive chosen-ciphertext secure if for any t-time IND-sT-CCA adversary $\mathscr{A}$ that performs at most $q_{H_2} H_2$ queries, q_T chosen designated decryption trapdoor queries, and $q_{\mathscr{C}}$ chosen decryption queries, we have that $\mathrm{Adv}_{\mathscr{E},\mathscr{A}}^{\mathrm{CCA}} < \varepsilon$. In other words, we call that $\mathscr{E}$ is $(t, q_{H_2}, q_T, q_C, \varepsilon)$ IND-sT-CCA secure.

We define our concrete SETRE scheme $\mathscr{E}$ to be IND-sT-CPA secure by simply disallowing the adversary $\mathscr{A}$ to perform decryption queries in the simulation security game described above.

Definition 6. Our concrete SETRE scheme $\mathscr{E}$ is said to be (t, q_T, ε)-selective designated decryption time, adaptive chosen-plaintext secure if $\mathscr{E}$ is $(t, q_{H_2}, q_T, 0, \varepsilon)$-selective designated decryption time, chosen-ciphertext secure. In other words, we call that $\mathscr{E}$ is $(t, q_{H_2}, q_T, \varepsilon)$ IND-sT-CPA secure.

4. Concrete Scheme of SETRE

We will attempt to propose a concrete scheme of SETRE based on the BDH assumption in the random oracle model.

4.1. Construction. The server-passive, scalable, user-anonymous TRE scheme proposed by Black and Chan (abbreviated as BC-TRE) laid the foundation of TRE. We now describe the concrete SETRE construction scheme. The scheme includes the following algorithm 7 tuples:

Setup: generates a public parameter params $= \{G_1, G_2, q, e, P, H_1, H_2, n\}$ from a security parameter k, where G_1 is an ECDLP additive group over a finite field, G_2 is a DLP multiplicative group over a finite field, and the order of G_1, G_2 is a prime number q, e: $G_1 \times G_1 \longrightarrow G_2$ is a bilinear mapping that satisfies Definition 1, $P \in G_1^*$ is the generator of additive group G_1, and H_1: $\{0, 1\}^* \longrightarrow G_1$ and H_2: $G_2 \longrightarrow \{0, 1\}^n$ (n is the length of the plaintext) are hash functions.

TS-KeyGen: the time server selects a random number $s \in \mathbb{Z}_q^*$ as the private key $ts_{\text{priv}} = s \in \mathbb{Z}_q^*$ of the time server and then calculates and generates the time server's public key $ts_{\text{pub}} = sP$. Similarly, the time server selects a random number set as the session private key set $\mathrm{TS}_{\text{spriv}} = \{x_1, x_2, \ldots, x_l\} \in \mathbb{Z}_q^*$ of the time server and then calculates and generates the corresponding time server's session public key set $\mathrm{TS}_{\text{spub}} = \{x_1 P, x_2 P, \ldots, x_l P\} \in G_1^*$ in which $l \approx 175200$ if we assume that a time trapdoor needs to be generated every half an hour and meet the demand for 10 consecutive years.

User-KeyGen: a user selects a random number $u \in \mathbb{Z}_q^*$ as its private key usk $= u \in \mathbb{Z}_q^*$ and then calculates and generates the system user's public key upk $= uP$.

Enc: the sender uses the public key $\mathrm{upk}_r = uP$ of the receiver, the public key $ts_{\text{pub}} = sP$ of the time server, a designated decryption time point $T \in \{0, 1\}^*$, and the time server's session public key $ts_{\text{spub}} = xP$ corresponding to the designated decryption time point $T \in \{0, 1\}^*$ to encrypt the plaintext M as the following operations:

(1) Selects a random number $r \in \mathbb{Z}_q^*$ and calculates $U = rP$
(2) Calculates $S_{\text{pub}} = \mathrm{upk} + ts_{\text{pub}} + ts_{\text{spub}} = uP + sP + xP = (u + s + x)P$
(3) Calculates $K = e(S_{\text{pub}}, rH_1(T)) = e(P, H_1(T))^{r(u+s+x)}$
(4) Outputs the ciphertext C

$$C = \langle U, V \rangle = \langle rP, M \oplus H_2(K) \rangle. \tag{4}$$

TS-Rel: the time server takes its own fixed private key $ts_{\text{priv}} = s$ and the session private key $ts_{\text{spriv}} = x$ of the current release time T and produces the time server's time trapdoor $S_T = (s + x)H_1(T)$.

UT_Rel: the receiver takes the private key usk $= u$ of his own and the current designated decryption time T and produces the user's time trapdoor $U_T = uH_1(T)$.

Dec: the receiver uses the time trapdoors S_T and U_T of the designated decryption time point $T \in \{0, 1\}^*$ to decrypt the ciphertext $C = \langle U, V \rangle$ as the following operations:

(1) Calculates $K' = e(U, S_T + U_T)$
(2) Calculates $V \oplus H_2(K')$ to recover the corresponding plaintext M

Suppose C is the valid ciphertext; then, we have $U = rP$ and $V = M \oplus H_2(K)$. We can verify the correctness of the decryption as described in the following:

$$\begin{aligned}
K' &= e\left(U_T + S_T, U\right)\\
&= e\left(uH_1(T) + (s+x)H_1(T), rP\right)\\
&= e\left((u+s+x)H_1(T), rP\right)\\
&= e\left(H_1(T), P\right)^{r(u+s+x)}\\
&= K,\\
V \oplus H_2(K') &= V \oplus H_2(K)\\
&= M \oplus H_2(K) \oplus H_2(K)\\
&= M.
\end{aligned} \tag{5}$$

4.2. Security of the Scheme. We give the proof that our SETRE scheme is noninteractive and semantically secure against CPA in the random oracle model, supposing that the BDH assumption is true [37].

Theorem 1. *Suppose that there is an adversary $\mathscr{A}$ who can break our SETRE scheme with the advantage of ϵ; then, a challenger $\mathscr{B}$, who can overcome the BDH problem with probability at least $\varepsilon' = \varepsilon / e q_T q_{H_2}$, is constructed, where e is the natural logarithm's base and q_T and q_{H_2} are the maximum number of times we assume the adversary $\mathscr{A}$ can query the time trapdoor and H_2 hash operation.*

Proof. Let $\mathscr{A}$ denote an adversary who has advantage ϵ to break the SETRE. Assume that $\mathscr{A}$ performs no more than q_{H_2} hash operation queries to H_2, no more than q_T user trapdoors, and the time server trapdoor queries, where q_T and q_{H_2} are positive. Let $\mathscr{B}$ denote a challenger who overcomes the BDH problem with probability no less than $\varepsilon' = \varepsilon / e q_{H_2} q_T$. Therefore, if the BDH assumption holds in G_1, then we can ignore ε'; furthermore, the advantage of $\mathscr{A}$ to break the SETRE can be ignored. And $\mathscr{B}$, who simulates as the challenger, will interact with adversary $\mathscr{A}$ as follows:

Preparation: let G_1 be an ECDLP additive group over a finite field, G_2 be a DLP multiplicative group over a finite field, the order of G_1, G_2 be a prime number q, e: $G_1 \times G_1 \longrightarrow G_2$ be a bilinear mapping that satisfies Definition 1, and $P \in G_1^*$ be the generator of additive group G_1. Give the challenger $\mathscr{B}$ the public parameter P, $P_1 = aP = uP + sP + xP$, $P_2 = bP$, and $P_3 = cP \in G_1$; the goal of $\mathscr{B}$ is to calculate the value of $v = e(P,P)^{abc} \in G_2$, where $a, b, c \in Z_q^*$.

Initialization: the adversary $\mathscr{A}$ outputs a pair of designated decryption time points T_0^* and T_1^* to be challenged.

Setup: the challenger $\mathscr{B}$ gives $\mathscr{A}$ the public keys $\mathrm{upk}_r = uP$, $ts_{\mathrm{pub}} = sP$, and $ts_{\mathrm{spub}} = xP$.

Phase 1: the adversary $\mathscr{A}$ initiates $1, \ldots, m$ queries, and $\mathscr{B}$ gives the response, respectively, where for the i-th query, $\mathscr{B}$'s response is described as follows:

(1) H_1 and H_2 queries: every point in time, the adversary $\mathscr{A}$ can perform queries of the random oracles H_1 and H_2. In response to H_1 queries, the challenger $\mathscr{B}$ keeps a list of quadruples $\langle T_j, h_j, m_j, n_j \rangle$, which we will call it the H_1-list and is initially set to be empty. If $\mathscr{A}$ performs a query of H_1 at a time point $T_i \in \{0,1\}^*$, then $\mathscr{B}$ gives the response as follows:

① If the query about T_i has been made before, then $\mathscr{B}$ takes $H_1(T_i) = h_i \in G_1$ as its response.

② If not, $\mathscr{B}$ chooses a new random bit $n_i \in \{0,1\}$ to satisfy $\Pr[n_i = 0] = 1/(q_s + 1)$.

③ $\mathscr{B}$ takes a random number $m_i \in Z_p$.

If $n_i = 0$ holds, $\mathscr{B}$ calculates $h_i \longleftarrow P_2 + m_i \cdot P \in G_1$.

If $n_i = 1$ holds, $\mathscr{B}$ calculates $h_i \longleftarrow m_i \cdot P \in G_1$.

④ $\mathscr{B}$ adds the quadruple $\langle T_i, h_i, m_i, n_i \rangle$ to the H_1-list and takes $H_1(T_i) = h_i \in G_1$ as its response to $\mathscr{A}$.

In the same way, $\mathscr{A}$ can perform a query to H_2 at any point in time. The H_2-list is initially set to be empty. $\mathscr{B}$ gives the response to the query on $H_2(K_i)$ by selecting a new random $V_i \in \{0,1\}^{\log_2 p}$ as the value of $H_2(K_i)$ for every new K_i and adding the tuple (K_i, V_i) to H_2-list. If H_2-list already contains (K_i, V_i), then $\mathscr{B}$ takes (K_i, V_i) from H_2-list and returns it to $\mathscr{A}$ as the response value.

(2) Time trapdoor queries: if the adversary $\mathscr{A}$ performs queries of the time trapdoor at a time point $T_i \notin \{T_0^*, T_1^*\}$, then the challenger $\mathscr{B}$ gives the response as follows:

① $\mathscr{B}$ runs the above H_1 query algorithm and obtains $H_1(T_i) = h_i \in G_1$ and makes $\langle T_i, h_i, m_i, n_i \rangle$ as the corresponding entry in H_1-list.

② If $n_i = 0$, then $\mathscr{B}$ aborts the simulation security game and admits failure.

③ If $n_i = 1$, we obtain $h_i = m_i \cdot P \in G_1$. Let $T_{u_i} = m_i \cdot \mathrm{upk}_r$ and $T_{T_i} = m_i \cdot (ts_{\mathrm{pub}} + ts_{\mathrm{spub}})$; then, we can transform them to get $T_{u_i} = uH_1(T_i)$ and $T_{T_i} = (s + x_i)H_1(T_i)$. Therefore, T_{u_i} is the correct and legal user time trapdoor of T_i, and T_{T_i} is the correct and legal time server trapdoor of T_i. $\mathscr{B}$ gives T_{u_i} and T_{T_i} to $\mathscr{A}$.

Challenge: the adversary $\mathscr{A}$ selects a pair of designated decryption time points (T_0^*, T_1^*) to be challenged. The challenger $\mathscr{B}$ produces the challenge ciphertext as follows:

① The challenger $\mathscr{B}$ runs the above H_1 query algorithm twice to obtain h_0^* and $h_1^* \in G_1$ which satisfy $H_1(T_0^*) = h_0^*$ and $H_1(T_1) = h_1^*$.

② For $i = 0, 1$, we let $\langle T_0^*, h_0^*, m_0^*, n_0^* \rangle$ and $\langle T_1^*, h_1^*, m_1^*, n_1^* \rangle$ to be the corresponding tuples on the H_1-list. If $n_0' = n_1' = 1$, then the challenger $\mathscr{B}$ aborts the simulation security game and admits failure.

③ Obviously, at least one of n_0^* and n_1^* must be equal to zero. $\mathscr{B}$ randomly takes $\flat \in \{0, 1\}$ such that $n_\flat = 0$.

④ $\mathscr{B}$ takes the challenge ciphertext $C_\flat^* = [P_3, J]$ for random $J \in \{0, 1\}^{\log_2 p}$ as its response. Obviously, this challenge implicitly defines $H_2(e(H_1(T_\flat^*), c \cdot \text{upk}_r) \cdot e(H_1$ $(T_\flat^*), c \cdot ts_{\text{pub}}) \cdot e(H_1(T_\flat^*), c \cdot ts_{\text{spub}\flat})) = J$. That is to say,

$$\begin{aligned} J &= H_2\Big(e\Big(cH_1(T_\flat^*), \text{upk}_r + ts_{\text{pub}} + ts_{\text{spub}\flat}\Big)\Big) \\ &= H_2\left(e\left(P_2 + m_\flat^* P, (u + s + x_\flat)P\right)^c\right) \\ &= H_2\Big(e(P, P)^{c(u+s+x_\flat)(b+m_\flat^*)}\Big). \end{aligned} \tag{6}$$

It can be seen that $C_\flat^*$ is the corresponding valid and real ciphertext for $T_\flat^*$.

Phase 2: the adversary $\mathscr{A}$ performs other queries of $q_{m+1}, \ldots, q_{\text{num}}$, and the challenger $\mathscr{B}$ responds as shown in Phase 1.

Guess: in the end, the adversary $\mathscr{A}$ outputs a guess of $\flat' \in \{0, 1\}$ to indicate whether the challenge ciphertext $C_\flat^*$ is a valid ciphertext for $\text{Enc}(\text{upk}_r, ts_{\text{pub}}, ts_{\text{spub}0}, T_0^*)$ or $\text{Enc}(\text{upk}_r, ts_{\text{pub}}, ts_{\text{spub}1}, T_1^*)$. Now, the challenger $\mathscr{B}$ randomly selects a tuple (K_j, V_j) from the H_2-list and outputs $K/e(\text{upk}_r, ts_{\text{pub}}, ts_{\text{spub}\flat}, P_3)^{m_\flat^*}$ as a guess of $e(P, P)^{abc}$. If $\mathscr{A}$ has ever inquired about one of $H_2(e(cH_1(T_0^*), \text{upk}_r + ts_{\text{pub}} + ts_{\text{spub}0}))$ or $H_2(e$ $(cH_1$ $(T_1^*), \text{upk}_r + ts_{\text{pub}} + ts_{\text{spub}1}))$, the H_2-list has a probability of 1/2 that contains (K_j, V_j), $K_j = H_2$ $(e(cH_1(T_\flat^*), \text{upk}_r + ts_{\text{pub}} +$ $ts_{\text{spub}\flat}) = H_2$ $(e(P,$ $P)^{c(u+s+x_\flat)(b+m_\flat^*)}))$. If $\mathscr{B}$ takes this tuple (K_j, V_j) from the H_2-list, then $K/e(\text{upk}_r, ts_{\text{pub}}$ $+ts_{\text{spub}\flat, P_3})^{m_\flat^*} = e(P,$ $P)^{\text{abc}}$.

The whole security simulation game is completed here. Next, we calculate the value of ϵ' which is the lowest probability of $\mathscr{B}$ correctly outputting $e(P, P)^{abc}$. It is easy to know that the premise that it can correctly output its guess value of $e(P, P)^{abc}$ is that the game can continue to the guessing stage without terminating the game in the middle. Now, we analyze the possibility that $\mathscr{B}$ does not terminate the game while the game is in progress. For this purpose, we first give the definition of the following events:

$\mathscr{E}_0$: in the stage when the adversary $\mathscr{A}$ performs queries of the time trapdoor, the challenger $\mathscr{B}$ does not terminate the simulation security game

$\mathscr{E}_1$: in the challenge stage, the challenger $\mathscr{B}$ does not terminate the simulation security game

We first state that, as in [38], events $\mathscr{E}_1$ and $\mathscr{E}_2$ occur with a high enough probability. Next, we give the following three claims.

Claim 1 : in the stage when the adversary $\mathscr{A}$ performs queries of the time trapdoor, the probability that the challenger $\mathscr{B}$ does not terminate the simulation security game is $1/e$ at least. Thus, $P_r[\mathscr{E}_0] \geq 1/e$. □

Proof. When the adversary $\mathscr{A}$ queries for the time trapdoor of time points, for the sake of generality, we suppose that $\mathscr{A}$ does not query the same time trapdoor twice. A trapdoor (the user's time trapdoor or the time server's time trapdoor) query causes $\mathscr{B}$ to terminate the simulation security game with a probability of $1/(q_T + 1)$; therefore, a trapdoor query does not cause $\mathscr{B}$ to terminate the game with a probability of $(1 - 1/(q_T + 1))$. In addition, since the maximum number of times $\mathscr{A}$ can query the time trapdoor is q_T, the probability that the simulation security game will not be terminated after q_T queries is $(1 - 1/(q_T + 1))^{q_T} \geq 1/e$ at least.

Claim 2: in the challenge stage, the probability that the challenger $\mathscr{B}$ does not terminate the simulation security game is $1/q_T$ at least. Thus, $P_r[\mathscr{E}_1] \geq 1/q_T$. □

Proof. If the adversary $\mathscr{A}$ can generate T_0^*, T_1^* with the property $n_0^* = n_1^* = 1$, then the challenger $\mathscr{B}$ will terminate the simulation security game during the challenge stage. Since $\mathscr{A}$ has not queried for the trapdoor for T_0^*, T_1^*, we have that n_0^*, n_1^* are independent of $\mathscr{A}$. Therefore, $P_r[n_\flat^* = 0] = 1/(q_T + 1)$ for $\flat = 0, 1$, and then we have that $P_r[n_0^* = n_1^* = 1] = (1 - 1/(q_T + 1))^2 \leq 1 - 1/q_T$. Therefore, there is a probability of at least $1/q_T$ that $\mathscr{B}$ does not terminate the game.

Since the adversary $\mathscr{A}$ is not allowed to query the time trapdoor of the designated decryption time T_0, T_1 during the game, the events $\mathscr{E}_0$ and $\mathscr{E}_1$ are independent of each other, so we can get $P_r[\mathscr{E}_0 \cap \mathscr{E}_1] \geq 1/eq_T$.

Assume that the adversary $\mathscr{A}$ has acquired the public keys $\text{upk}_r = uP$, $ts_{\text{pub}} = sP$, and $ts_{\text{spub}} = xP$ in the actual attack game. The adversary $\mathscr{A}$ selects a pair of designated decryption time points (T_0^*, T_1^*) to be challenged. The challenger $\mathscr{B}$ produces the challenge ciphertext $C_\flat^* = [P_3, J]$ as a response. Therefore, we have the following Claim 3.

Claim 3: in the actual attack game, the adversary $\mathscr{A}$ has at least the probability of ϵ to perform an H_2 query for one of $H_2(e(cH_1(T_0^*), \text{upk}_r + ts_{\text{pub}} + ts_{\text{spub}0}))$, $H_2(e(cH_1(T_1^*), \text{upk}_r + ts_{\text{pub}} + ts_{\text{spub}1}))$.

Before giving the proof, we first give the definition of the following events:

$\mathscr{E}_2$: in the actual attack game, $\mathscr{A}$ does not query either $H_2(e(cH_1(T_0^*), \text{upk}_r + ts_{\text{pub}} + ts_{\text{spub}0}))$ or $H_2(e(cH_1$ $(T_1^*), \text{upk}_r + ts_{\text{pub}} + ts_{\text{spub}1}))$

$\mathscr{E}_3$: in the guess stage, $\mathscr{A}$ outputs the guess $\flat'$ of $\flat$ satisfying $\flat = \flat'$ □

Proof. When $\mathscr{E}_2$ occurs, it is obvious that the bit $\flat \in \{0, 1\}$ indicates whether $C_\flat^*$ is the challenge ciphertext corresponding to the designated decryption time, which has nothing to do with $\mathscr{A}$'s knowledge. Thus, the probability of $P_r[\mathscr{E}_3]$ is 1/2 at most. In the real attack game, because $\mathscr{A}$ has the advantage of ϵ, we have $|P_r[\mathscr{E}_3] - 1/2| \geq \varepsilon$ and

$P_r[\neg\mathscr{E}_2] \geq 2\varepsilon$. Now, we give the specific argument for the truth of $P_r[\neg\mathscr{E}_2] \geq 2\varepsilon$ as follows:

$$\begin{aligned} P_r[\mathscr{E}_3] &= P_r[\mathscr{E}_3|\mathscr{E}_2]P_r[\mathscr{E}_2] + P_r[\mathscr{E}_3|\neg\mathscr{E}_2]P_r[\neg\mathscr{E}_2] \\ &\leq P_r[\mathscr{E}_3|\mathscr{E}_2]P_r[\mathscr{E}_2] + P_r[\neg\mathscr{E}_2] \\ &\leq \frac{1}{2}P_r[\mathscr{E}_2] + P_r[\neg\mathscr{E}_2] \\ &\leq \frac{1}{2} + \frac{1}{2}P_r[\neg\mathscr{E}_2]P_r[\mathscr{E}_3] \\ &\geq P_r[\mathscr{E}_3|\mathscr{E}_2]P_r[\mathscr{E}_2] \\ &\geq \frac{1}{2}P_r[\mathscr{E}_2] \geq \frac{1}{2} - \frac{1}{2}P_r[\neg\mathscr{E}_2]. \end{aligned} \tag{7}$$

From the above two formulas, we know that $\epsilon \leq |P_r[\mathscr{E}_3] - (1/2)| \leq (1/2)P_r[\neg\mathscr{E}_2]$. Thus, we have $P_r[\neg\mathscr{E}_2] \geq 2\varepsilon$ in the actual attack game.

If the challenger $\mathscr{B}$ does not terminate the game, it means that, in the process of simulating the actual attack game, the adversary $\mathscr{A}$ has queried one of $H_2(e(cH_1(T_0^*), \mathrm{upk}_r + ts_{\mathrm{pub}} + ts_{\mathrm{spub0}}))$, $H_2(e(cH_1(T_1^*), \mathrm{upk}_r + ts_{\mathrm{pub}} + ts_{\mathrm{spub1}}))$. Thus, $P_r[\neg\mathscr{E}_2] \geq 2\epsilon$.

Claim 4: the probability that the challenger $\mathscr{B}$ can solve the BDH problem successfully in the guess stage is ϵ/q_{H_2}. □

Proof. Assuming the event of Claim 3 occurs, the value of one of the two cases of $e(cH_1(T_\flat^*), \mathrm{upk}_r + ts_{\mathrm{pub}} + ts_{\mathrm{spub}\flat})$ will be stored in the H_2-list. Consequently, in the guess stage, the challenger $\mathscr{B}$ has at least the probability of $1/q_{H_2}$ to select the correct pair from the H_2-list. Therefore, on the premise that $\mathscr{B}$ does not terminate the simulation game, the possibility that $\mathscr{B}$ can successfully solve the BDH problem is $\epsilon/(q_{H_2})$.

According to Claims 1 and 2, during the simulation game, the probability that the challenger $\mathscr{B}$ will not terminate the game is at least $1/eq_T$. And according to Claim 4, if $\mathscr{B}$ does not terminate the simulation security game, the probability that $\mathscr{B}$ can successfully solve the BDH problem is ϵ/q_{H_2}. Therefore, through the security simulation game of the aforementioned adversary $\mathscr{A}$ and challenger $\mathscr{B}$, the possibility of successfully solving the BDH problem is $\varepsilon/eq_Tq_{H_2}$. Thus, Theorem 1 is proved. □

4.3. Efficiency Analysis. We contrast between our SETRE scheme and two representative noninteractive time server TRE schemes: the classic BC-TRE scheme put forward by Blake and Chan [9] and the AnTRE scheme, which has highest efficiency up till now, put forward by Chalkias et al. [12].

We let BP be a notation of the bilinear pairing operation, PA_{ec} and PM_{ec} be a notation of point addition and point multiplication operations in G_1 separately. Let Exp_{ec} be a notation of the exponentiation operation in G_2 and Inv be a notation of the modular inverse operation in $\mathbb{Z}_q^*$. Let H_1 represent a hash function operation that maps binary strings of any length to an element in group G_1, H_2 represent the hash function operation that maps an element in group G_2 to a string of $\log_2^q$ length 0 and 1, and H_3 represent the hash function operation of mapping a binary string of any length to an element of $\mathbb{Z}_q^*$. Based on the MIRACL large integer library, we program and implement the basic operations described above, in which the relevant parameters are set as follows: the elliptic curve is a supersingular elliptic curve E: $y = x^3 + 1 \bmod p$ on the finite field F_p (p is a 512-bit large prime number), and its prime order q is a 160-bit prime number; the bilinear map uses the Tate pairing algorithm to map the aforementioned discrete logarithm subgroup on the elliptic curve to the discrete logarithm subgroup on F_{p^2}. The configuration of the running environment is as follows: Intel(R) Core(TM) i5-4210M @ 2.60 GHz microprocessors, 64 bit and 8 GB memory, Microsoft Visual Studio 2010. 987654321 is the seed that generates the associated random numbers. We take the calculation time of Exp_{ec} as the basic unit so that the calculation results are not related to the specific computer performance. We then calculate and record the ratio of the calculation time of each related basic operation in these schemes to the calculation time of Exp_{ec}, as shown in Table 1.

Table 1: Calculation cost of related basic operations relative to the Exp_{ec} operation.

Related basic operations	Notation	Relative cost
Bilinear pairing	BP	3.4457
Point addition in G_1	PA_{ec}	0.0072
Point multiplication in G_1	PM_{ec}	1
Exponentiation in G_2	Exp_{ec}	0.3220
Modular inverse in $\mathbb{Z}_q^*$	Inv	0.0030
Hash function: $\{0,1\}^* \longrightarrow G_1$	H_1	0.3368
Hash function: $G_1 \longrightarrow \{0,1\}^{\log_2^q}$	H_2	0.0782
Hash function: $\{0,1\}^* \longrightarrow \mathbb{Z}_q^*$	H_3	0.0030

In our SETRE scheme, the TS-Rel stage requires one PM_{ec} and one H_1 to calculate $S_T = (s + x)H_1(T)$, and the total calculation cost of the TS-Rel stage is 1.003. The Enc stage requires one PM_{ec} for rP, two PA_{ec} for S_{pub}, one H_1, one PM_{ec}, and one BP for $e(rH_1(T), S_{\mathrm{pub}})$, and one H_2 for $H_2(K)$, and the total calculation cost of the Enc stage is 5.875. The Dec stage requires one H_1, one PM_{ec}, one PA_{ec}, and one BP for $K' = e(U, S_T + uH_1(T))$ and one H_2 for $M \oplus H_2(K)$, and the total calculation cost of the Dec stage is 4.868. We sum up the calculation cost of the schemes of BC-TRE, AnTRE, and our SETRE as shown in Table 2. It should be pointed out that the hash functions H_1 and H_2 in the scheme of AnTRE are approximately equivalent to H_3 in Table 1, and the hash functions H_3 and H_4 in the scheme of AnTRE are approximately equivalent to H_2 in Table 1.

Table 2 shows that our SETRE scheme has improved by 32.4% and 10.8%, respectively, compared with the schemes of BC-TRE and AnTRE. In addition, in the aspect of security,

Table 2: Calculation cost of BC-TRE, AnTRE, and our SETRE.

Scheme	Phase		
	BC-TRE [9]	AnTRE [12]	Our SETRE
TS − Rel	$\mathrm{PM}_{ec} + H_1 = 1.337$	$\mathrm{PM}_{ec} + \mathrm{Inv} + H_3 = 1.006$	$\mathrm{PM}_{ec} + H_1 = 1.337$
Enc	$3\mathrm{BP} + 2\mathrm{PM}_{ec} + H_3 = 12.340$	$4\mathrm{PM}_{ec} + \mathrm{PA}_{ec} + \mathrm{Exp}_{ec} + \mathrm{BP} + 2H_2 + 2H_3 = 7.934$	$2\mathrm{PM}_{ec} + 2\mathrm{PA}_{ec} + H_1 + \mathrm{BP} + H_2 = 5.875$
Dec	$\mathrm{BP} + \mathrm{Exp}_{ec} + H_1 + H_2 = 4.183$	$\mathrm{BP} + \mathrm{PM}_{ec} + 2H_2 + H_3 = 4.605$	$H_1 + \mathrm{PM}_{ec} + \mathrm{PA}_{ec} + \mathrm{BP} + H_2 = 4.868$
Total	17.86	13.55	12.08

our SETRE scheme realizes one-time pad for the time trapdoor. Therefore, compared with the previous schemes, our SETRE greatly improves the security of the time server's private key. In terms of storage, we need 160 bits of storage space for each private key of the time server and 1024 bits for each public key of the time server. Therefore, if it is assumed that the time server broadcasts a time trapdoor every half an hour and needs to store the session time server private key and public key for 10 years, then the additional storage space required by our scheme is $24 * 2 * 365 * 10 * (1024 + 160)$ bit $\approx 24.7\,\mathrm{MB}$, which only adds almost negligible storage burden to the time server.

5. Generic Scheme of SETRE

We will attempt to propose a generic scheme of SETRE based on GPKE and call it generic SETRE, abbreviated as GSETRE.

5.1. Formal Definition.

We now formalize the definition of our GSETRE scheme.

Definition 7. Our GSETRE scheme includes three entities which are time server, sender, and receiver and a polynomial-time randomized algorithm 6 tuples $\mathscr{E}_{\mathrm{GSETRE}} = (\mathrm{Setup}, \mathrm{TS\_KeyGen}, \mathrm{User\_KeyGen}, \mathrm{Enc}, \mathrm{Rel}, \mathrm{Dec})$, where

Setup: generates a public parameter params from a security parameter

TS_KeyGen: calculates and generates the fixed public/private key pair $(ts_{\mathrm{pub}}, ts_{\mathrm{priv}})$ and the session public/private key pair $(ts_{\mathrm{spub}}, ts_{\mathrm{spriv}})$ of the time server

User_KeyGen: calculates and generates the public/private key pair (upk, usk) of the system user

Enc: inputs $M, \mathrm{upk}, T, ts_{\mathrm{pub}}$, and ts_{spub} that correspond to the designated decryption time point T to the Enc algorithm of $\mathscr{E}_{\mathrm{GPKE}}$ and outputs the ciphertext C_{GSETRE}

Rel: given the private key ts_{priv} of the time server, a designated decryption time point T, and its corresponding session private key ts_{spriv} and produces a time server's time trapdoor S_T

Dec: inputs S_T and usk into the Dec algorithm of $\mathscr{E}_{\mathrm{GPKE}}$ and outputs plaintext M or $\perp$

5.2. Construction.

We construct a $\mathscr{E}_{\mathrm{GSETRE}}$ scheme by introducing $\mathscr{E}_{\mathrm{SETRE}}$ into $\mathscr{E}_{\mathrm{PKE}}$. $\mathscr{E}_{\mathrm{GSETRE}}$ includes the following algorithm 6 tuples:

Setup: this algorithm is consistent with the Setup algorithm of our concrete $\mathscr{E}_{\mathrm{SETRE}}$ scheme

TS_KeyGen: this algorithm is consistent with the TS_KeyGen algorithm of our concrete $\mathscr{E}_{\mathrm{SETRE}}$ scheme

User_KeyGen: this algorithm is consistent with the User_KeyGen algorithm of our concrete $\mathscr{E}_{\mathrm{SETRE}}$ scheme

Enc: the sender uses $\mathrm{upk}_r = uP$ of the receiver, $ts_{\mathrm{pub}} = sP$ of the time server, a designated decryption time point $T \in \{0,1\}^*$, and $ts_{\mathrm{spub}} = xP$ corresponding to the designated decryption time point $T \in \{0,1\}^*$ to encrypt the plaintext M as the following operations:

(1) Uses upk_r to encrypt the plaintext M and calculates $\mathscr{E}_{\mathrm{PKE}}$'s ciphertext $C_{\mathrm{GPKE}} = \mathrm{Enc}(M, uP)$.
(2) Chooses a function $f(\cdot)$ randomly and calculates $U = f(\cdot) \cdot P$.
(3) Uses C_{GPKE}, ts_{pub}, ts_{spub}, and T to calculate

$$\begin{aligned} V &= C_{\mathrm{GPKE}} \cdot e\left(H_1(T), f(\cdot) \cdot \left(ts_{\mathrm{pub}} + ts_{\mathrm{spub}}\right)\right) \\ &= C_{\mathrm{GPKE}} \cdot e\left(H_1(T), f(\cdot) \cdot (s + x)P\right). \end{aligned} \tag{8}$$

(4) Outputs $\mathscr{E}_{\mathrm{GSETRE}}$'s ciphertext $C_{\mathrm{GSETRE}} = \langle U, V \rangle = \langle f(\cdot) \cdot P, \mathrm{Enc}(M, uP) \cdot e(H_1(T), f(\cdot) \cdot (s + x)P) \rangle$.

Rel: this algorithm is consistent with the ST_Rel algorithm of our concrete $\mathscr{E}_{\mathrm{SETRE}}$ scheme

Dec: the receiver uses the time trapdoors S_T of the designated decryption time point $T \in \{0,1\}^*$ and the private key $\mathrm{usk}_r = u$ of the receiver to decrypt the ciphertext $C_{\mathrm{GSETRE}} = \langle U, V \rangle$ as the following operations:

(1) Calculates $C'_{\mathrm{GPKE}} = (V / e(S_T, U))$
(2) Calculates $\mathrm{Dec}(C'_{\mathrm{GPKE}}, u)$ to recover the corresponding plaintext M

Suppose C_{GSETRE} is the valid ciphertext; then, we have $U = f(\cdot) \cdot P$ and $V = C_{\mathrm{GPKE}} \cdot e(H_1(T), f(\cdot) \cdot (s + x)P)$. We can verify the correctness of the decryption as described in the following:

$$
\begin{aligned}
C'_{\mathrm{GPKE}} &= \frac{V}{e\left(S_T, U\right)} \\
&= \frac{V}{e\left((s+x)H_1(T), f(\cdot)\cdot P\right)} \\
&= \frac{C_{\mathrm{GPKE}}\cdot e\left(H_1(T), f(\cdot)\cdot (s+x)P\right)}{e\left(H_1(T), f(\cdot)\cdot (s+x)P\right)} \\
&= C_{\mathrm{GPKE}}\mathrm{Dec}\left(C'_{\mathrm{GPKE}}, u\right) \\
&= \mathrm{Dec}\left(C_{\mathrm{GPKE}}, u\right) = M.
\end{aligned}
\tag{9}
$$

5.3. Security and Efficiency Analysis. From the perspective of security, since the $\mathscr{E}_{\mathrm{GSETRE}}$ scheme is obtained by introducing $\mathscr{E}_{\mathrm{SETRE}}$ into the $\mathscr{E}_{\mathrm{GPKE}}$ scheme, which is equivalent to encapsulating the $\mathscr{E}_{\mathrm{GPKE}}$ scheme's ciphertext, the security of the $\mathscr{E}_{\mathrm{GPKE}}$ scheme will be enhanced after introducing $\mathscr{E}_{\mathrm{SETRE}}$. Firstly, the decryption operation needs to decrypt the $\mathscr{E}_{\mathrm{SETRE}}$'s ciphertext to get the ciphertext of the $\mathscr{E}_{\mathrm{GPKE}}$ scheme. However, the decryption of $\mathscr{E}_{\mathrm{SETRE}}$ requires a valid time trapdoor, and the attacker cannot construct the required time trapdoor without knowing the time server's private key and session private key. Secondly, decrypting the $\mathscr{E}_{\mathrm{GPKE}}$ ciphertext requires the private key of the legitimate receiver.

From the perspective of efficiency, compared with the $\mathscr{E}_{\mathrm{GPKE}}$ scheme, the $\mathscr{E}_{\mathrm{GSETRE}}$ scheme adds other additional operations in the encryption and decryption process, which inevitably leads to a decrease in efficiency. However, when using the idea of the general scheme to construct a concrete scheme, the parameters of the $\mathscr{E}_{\mathrm{SETRE}}$ scheme can be integrated into the same logical step of the $\mathscr{E}_{\mathrm{GPKE}}$ scheme as far as possible, so as to minimize the decline of efficiency. In addition, in terms of storage space, the time server only needs to add a small amount of storage space, as described in the above section.

6. Summary and Outlook

With the purpose of enhancing TRE security, a concrete SETRE scheme and a generic SETRE scheme based on the BDH assumption in the random oracle model are put forward. In our SETRE schemes, the time server uses a different session key to generate an "encryption-like" trapdoor at different time points. This operation uses the idea of one-time pad for the generation of each time trapdoor, which prevents the time trapdoor from being known in advance due to the leakage of the time server's private key and thus prevents the ciphertext from being decrypted in advance.

To ensure the anonymity of each system user identity to the time server, most current TRE solutions use broadcast to distribute time trapdoors. If time trapdoors are broadcast in a coarse-grained manner, many users may not have corresponding time trapdoors for the specified decryption time. In order to meet the time trapdoor specified by the user as far as possible, it is required to broadcast the time trapdoors with fine granularity, but this would waste communication resources. Therefore, designing a TRE that can support the specified arbitrary release time, anonymize the user identity, and prevent the time server from denial-of-service attacks will be a very practical and challenging task in the future. In addition, we will explore the combination of TRE with other cryptographic primitives, such as order-revealing encryption [39], so that more scenarios can have the function of controlling the decryption time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

A preliminary version of this paper appears in the "International Symposium on Security and Privacy in Social Networks and Big Data-6th International Symposium SocialSec 2020, Tianjin, China," in September 26-27, 2020, based on https://link.springer.com/book/10.1007%2F978-981-15-9031-3?page=3#toc.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by grants from the National Key R&D Program of China (no. 2018YFA0704703) and the National Natural Science Foundation of China (nos. 61802111, 61972073, and 61972215).

References

[1] R. L. Rivest, A. Shamir, and D. A. Wagner, "Time-lock puzzles and timed-release crypto," Technical Report MIT/LCS/TR-684, MIT LCS Tech, Cambridge, MA, USA, 1996.

[2] T. May, *Timed-release Crypto*, Unpublished manuscript, 1993.

[3] B. Alexander, D. Levshun, N. Krasilnikova et al., "Determination of young generation's sensitivity to the destructive stimuli based on the information in social networks," *Journal of Internet Services and Information Security (JISIS)*, vol. 9, no. 3, pp. 1–20, 2019.

[4] M. Kolomeets, A. Benachour, D. El Baz et al., "Reference architecture for social networks graph analysis," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 10, no. 4, pp. 109–125, 2019.

[5] X. Wei, J. Liu, Y. Wang, C. Tang, and Y. Hu, "Wireless edge caching based on content similarity in dynamic environments," *Journal of Systems Architecture*, vol. 115, Article ID 102000, 2021.

[6] W. Yu, S. Lv, X. Guo, Z. Liu, Y. Huang, and B. Li, "Fsse: forward secure searchable encryption with keyed-block chains," *Information Sciences*, vol. 500, pp. 113–126, 2019.

[7] X. Jiang, X. Ge, Y. Jia, F. Kong, X. Cheng, and H. Rong, "An efficient symmetric searchable encryption scheme for cloud

storage," *Journal of Internet Services and Information Security (JISIS)*, vol. 7, no. 2, pp. 1–18, 2017.

[8] M. M. Casassa, H. Keith, and S. Martin, "The hp time vault service: exploiting ibe for timed release of confidential information," in *Proceedings of the 12th International Conference on World Wide Web*, pp. 160–169, ACM, Budapest, Hungary, May 2003.

[9] A. C.-F. Chan and I. F. Blake, "Scalable, server-passive, user-anonymous timed release cryptography," in *Proceedings of 25th IEEE International Conference on Distributed Computing Systems. ICDCS 2005*, pp. 504–513, IEEE, Colombus, OH, USA, June 2005.

[10] Y. H. Hwang, D. H. Yum, and P. J. Lee, "Timed-release encryption with pre-open capability and its application to certified e-mail system," *Lecture Notes in Computer Science*, Springer, Berlin, Germany, pp. 344–358, 2005.

[11] W. D. Alexander and Q. Tang, "Revisiting the security model for timed-release encryption with pre-open capability," in *Proceedings of International Conference on Information Security*, pp. 158–174, Springer, Valparaiso, CL, USA, October 2007.

[12] K. Chalkias, D. Hristu-Varsakelis, and G. Stephanides, "Improved anonymous timed-release encryption," in *Proceedings of European Symposium on Research in Computer Security ESORICS 2007*, pp. 311–326, Springer, Dresden, Germany, September 2007.

[13] J. H. Cheon, N. Hopper, Y. Kim, and I. Osipkov, "Provably secure timed-release public key encryption," *ACM Transactions on Information and System Security*, vol. 11, no. 2, pp. 1–44, 2008.

[14] A. Fujioka, Y. Okamoto, and T. Saito, "Generic construction of strongly secure timed-release public-key encryption," in *Proceedings of Australasian Conference on Information Security and Privacy*, pp. 319–336, Springer, Melbourne, Australia, July 2011.

[15] M. Mohammad, T. Moran, and V. Salil, "Time-lock puzzles in the random oracle model," in *Proceedings of Advances in Cryptology-CRYPTO 2011, LNCS 6841*, pp. 39–50, Springer, Santa Barbara, CA, USA, August 2011.

[16] J. Xiong, F. Li, J. Ma, X. Liu, Z. Yao, and P. S. Chen, "A full lifecycle privacy protection scheme for sensitive data in cloud computing," *Peer-to-peer Networking and Applications*, vol. 8, no. 6, pp. 1025–1037, 2015.

[17] K. Yuan, Z. Liu, C. Jia, J. Yang, and S. Lv, "Public key timed-release searchable encryption in one-to-many scenarios," *Acta Electronica Sinica*, vol. 43, no. 4, pp. 760–768, 2015.

[18] N. Bitansky, S. Goldwasser, A. Jain, O. Paneth, and B. Waters, "Time-lock puzzles from randomized encodings," in *Proceeding of Acm Conference on Innovations in Theoretical Computer Science*, pp. 345–356, ACM, Cambridge, MA, USA, January 2016.

[19] S.-Y. Huang, C.-I. Fan, and Y.-F. Tseng, "Enabled/disabled predicate encryption in clouds," *Future Generation Computer Systems*, vol. 62, pp. 148–160, 2016.

[20] S. Namasudra, "An improved attribute-based encryption technique towards the data security in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 9, p. e4364, 2017.

[21] W. Chen, Y. Wang, Z. Qin, and X. Liu, "Research on timed access of sensitive data based on dual encryption," *Journal of University of Electronic Science and Technology of China*, vol. 46, no. 3, pp. 588–593, 2017.

[22] C.-I. Fan, J.-C. Chen, S.-Y. Huang, J.-J. Huang, and W.-T. Chen, "Provably secure timed-release proxy conditional reencryption," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2291–2302, 2017.

[23] S. Y. Patil and J. N. Archana, "Conjunctive keyword search with designated tester and timing enabled proxy reencryption in health cloud," *International Journal for Innovative Research in Science and Technology*, vol. 4, no. 3, pp. 78–85, 2017.

[24] Q. Huang, Y. Yang, and J. Fu, "Secure data group sharing and dissemination with attribute and time conditions in public cloud," *IEEE Transactions on Services Computing*, vol. 99, 2018.

[25] Y. Watanabe and J. Shikata, "Timed-release computational secret sharing and threshold encryption," *Designs, Codes and Cryptography*, vol. 86, no. 1, pp. 17–54, 2018.

[26] H. Cao, K. Yuan, Y. Wang, Y. Yan, L. Zhou, and X. Chai, "Bidding model based on timed-release encryption and blockchain," *Journal of Henan University (Natural Science)*, vol. 49, no. 2, pp. 210–217, 2019.

[27] G. Choi and S. Vaudenay, "Timed-release encryption with master time bound key (extended)," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 10, no. 4, pp. 88–108, 2019.

[28] J. Hong, K. Xue, Y. Xue et al., "TAFC: time and attribute factors combined access control for time-sensitive data in public cloud," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 158–171, 2020.

[29] D. Unruh, "Revocable quantum timed-release encryption," in *Proceedings of Advances in Cryptology-EUROCRYPT 2014*, pp. 129–146, Springer, Copenhagen, Denmark, May 2014.

[30] T. Wang, Y. He, and L. Li, "New timed-release encryption based on indistinguishability obfuscation," *Application Research of Computers*, vol. 34, no. 9, pp. 2795–2798, 2017.

[31] L. Jia, F. Garcia, and M. Ryan, "Time-release protocol from bitcoin and witness encryption for sat," *Korean Circulation Journal*, vol. 40, no. 10, pp. 530–535, 2015.

[32] C. Li and B. Palanisamy, "Decentralized release of self-emerging data using smart contracts," in *Proceedings of 2018 IEEE 37th Symposium on Reliable Distributed Systems*, pp. 213–220, IEEE, Salvador, Brazil, October 2018.

[33] L. Jia, T. Jager, S. A. Kakvi, and W. Bogdan, "How to build time-lock encryption," *Designs, Codes and Cryptography*, vol. 86, pp. 2549–2586, 2018.

[34] W. Lai Jr, H. Chih-Wen, and J.-L. Wu, "A fully decentralized time-lock encryption system on blockchain," in *Proceedings of 2019 IEEE International Conference on Blockchain*, pp. 302–307, IEEE, Atlanta, GA, USA, July 2019.

[35] B. Cui, Z. Liu, and L. Wang, "Key-aggregate searchable encryption (kase) for group data sharing via cloud storage," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2374–2385, 2016.

[36] H. Tsuchida, T. Nishide, and E. Okamoto, "Expressive ciphertext-policy attribute-based encryption with fast decryption," *Journal of Internet Services and Information Security (JISIS)*, vol. 8, no. 4, pp. 37–56, 2018.

[37] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proceedings of Advances in Cryptology-EUROCRYPT 2004*, pp. 506–522, Springer, Interlaken, Switzerland, May 2004.

[38] J.-S. Coron, "On the exact security of full domain hash," in *Proceedings of Advances in Cryptology-CRYPTO 2000*, pp. 229–235, Springer, Santa Barbara, CF, USA, August 2000.

[39] Z. Liu, L. Jin, S. Lv et al., "Encodeore: reducing leakage and preserving practicality in order-revealing encryption," *IEEE Transactions on Dependable and Secure Computing*, vol. XX, 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5533417, 14 pages
https://doi.org/10.1155/2021/5533417

Research Article

User Identification Based on Integrating Multiple User Information across Online Social Networks

Wenjing Zeng,[1] **Rui Tang**,[1] **Haizhou Wang**,[1] **Xingshu Chen**,[1,2]
and Wenxian Wang[2]

[1] *School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China*
[2] *Cyber Science Research Institute, Sichuan University, Chengdu 610065, China*

Correspondence should be addressed to Haizhou Wang; whzh.nc@scu.edu.cn

Received 6 January 2021; Revised 24 April 2021; Accepted 11 May 2021; Published 26 May 2021

Academic Editor: Zhe-Li Liu

User identification can help us build more comprehensive user information. It has been attracting much attention from academia. Most of the existing works are profile-based user identification and relationship-based user identification. Due to user privacy settings and social network restrictions on user data crawl, user data may be missing or incomplete in real social networks. User data include profiles, user-generated contents (UGCs), and relationships. The features extracted in previous research may be sparse. In order to reduce the impact of the above problems on user identification, we propose a multiple user information user identification framework (MUIUI). Firstly, we develop multiprocess crawlers to obtain the user data from two popular social networks, Twitter and Facebook. Secondly, we use named entity recognition and entity linking to obtain and integrate locations and organizations from profiles and UGCs. We also extract URLs from profiles and UGCs. We apply the locations jointly with the relationships and develop several algorithms to measure the similarity of the display name, all locations, all organizations, location in profile, all URLs, following organizations, and user ID, respectively. Afterward, we propose a fusion classifier machine learning-based user identification method. The results show that the F1 score of MUIUI reaches 86.46% on the dataset. It proves that MUIUI can reduce the impact of user data that are missing or incomplete.

1. Introduction

With the development of social networks and their diversity, the number of active users on social networks has increased year by year. According to the report of Statista, the number of Facebook active users reached more than 2.7 billion in July 2020, and the number of Twitter active users reached 353 million in July 2020 [1]. People may have accounts on several social networks simultaneously. People can use Twitter to follow the latest developments in their areas of interest, use Facebook to post life trends and keep in touch with friends in life, use LinkedIn to post career information and keep contact with colleagues, and use Foursquare to post locations [2, 3]. If we can match accounts of an individual in different social networks, we can integrate his more comprehensive personal information and draw out their complete friend relationships [4]. This will facilitate social network friend recommendation [5], information diffusion [6], privacy protection [7, 8], community detection [9], etc. [10].

User identification across social networks is also called matching user accounts, user recognition, matching user accounts, user matching, or anchor linking [10]. In recent years, there have been many existing works on user identification across social networks. Most existing works use attributes in the profile to user identification [4, 11–15], such as display name, profile photo, and location. Due to user privacy settings, users may fill in fake information or choose not to fill in. These limitations make these methods quite fragile [16]. Some existing works are relationship-based user identification [17–19]. Relationships have higher discriminability, which is difficult to fake [10]. However, taking into user privacy settings and social network restrictions on data crawl, we may only get part of relationships. This will result in sparse and incomplete relationships. And, a number of existing works also use UGCs to user identification [4, 20].

These methods are usually based on posting time, location, writing style, or similarity of content [4]. However, they may ignore other information contained in the content, such as organizations and URLs. Because of user privacy settings and social network restrictions on data crawl, UGCs may not complete.

For most users, profiles, UGCs, and relationships may be missing or incomplete in real social networks. The features extracted in previous research may be sparse. If more effective features can be extracted from public and available user data, the impact of the above problems can be reduced. Therefore, our paper uses public multiple user information to perform user identification. The main advantages of MUIUI and contributions of our work are

(1) A complete user identification framework: we propose a complete user identification framework MUIUI, which is from data collection to user identification detection. Firstly, we crawl user data from two popular social networks and extract multiple user information from user data, which include profiles, UGCs, and relationships. Then, we extract features from multiple user information. Finally, we employ a fusion classifier to address the user identification problem.

(2) Conducted on popular social networks: this paper focuses on two popular social networks, Twitter and Facebook. We expand the raw dataset, which are those proposed in [21–25], crawled during November 2012 [9]. We screen the users in the raw dataset who are still alive and take them as positive samples. We construct negative samples which display names similar to the display names of half of positive samples. All negative samples and positive samples constitute the dataset used in this paper. We develop multiprocess crawlers to obtain the user data, include profiles displayed in December 2019, and UGCs and relationships published before January 2020, until we reach the limits of the social networks. We can disclose the dataset used in this paper. The MUIUI framework is conducted on this dataset.

(3) Extracted a set of effective features: we use named entity recognition to extract locations and organizations from profile and UGCs and regard them as all locations and all organizations. We use the entity link method to associate the alias of the locations and organizations. We propose methods to calculate the similarity of all locations, the similarity of all organizations, and the similarity of URLs in profile and UGCs. We apply the following relationship jointly with the location in profile to conduct user identification. The experiments prove that the features extracted in this paper are effective for user identification. The experiments also indicate that using multiple user information, we can improve the performance of user identification.

In the rest of this paper, Section 2 presents some related works. Section 3 introduces the basic background and formalizes the problem statement. In Section 4, we describe the user identification framework MUIUI. We do three experiments and compare with three existing works in Section 5. Finally, Section 6 concludes the paper and makes prospects for future work.

2. Related Works

In recent years, there have been much research studies on user identification across social networks. The existing research can be roughly divided into four categories: profile-based user identification, UGC-based user identification, relationship-based user identification, and user identification based on profile and user relationship.

Profile-based user identification only uses profile to identify users. In online social networks, attributes in a profile include the display name, user ID, introduction, location in profile, work education experience, and profile photos. Most research studies use one or more of these attributes. It can prove that these attributes are helpful for user identification. Some existing works only use one attribute for user identification, such as only use display name [11, 13, 26–28], only use profile photos [29], and only use locations [30–33]. These studies prove the feasibility of one attribute to perform user identification. As we know, social networks do not only contain a single attribute. And, applying several attributes jointly can improve the performance of user identification [10]. Li et al. [34] used display names and user IDs to link user identities. Motoyama and Varghese used various attributes, such as display name, location in the profile, age, and email, to link user identities [35]. Due to user privacy settings, users may fill in fake profile information or choose not to fill in. The accuracy of profile-based user identification will decrease.

UGC-based user identification only uses UGCs to identify users. Attributes in a UGC include locations, organizations, time, content, and writing style. Li et al. [4] calculated the similarity of UGCs on spatial, temporal, and content dimensions. Then, they proposed a cascaded three-level machine learning method to solve user identification. Goga et al. [36] used three features extracted from UGCs, such as location attached to UGCs, timestamp, and writing style, to identify users. Because of user privacy settings and social network restrictions on data crawl, UGCs may not complete. The robustness of above identification methods may be poor.

Relationship-based user identification only uses relationships to identify users. Xuan et al. [17] found that users usually maintain a similar circle of friends on different social networks. They use relationships and propose FRUI. Zhang et al. [18] proposed the energy model COSNET by considering the local and global similarities between multiple networks. Zhou et al. [19] sampled the network and learned the vector representation of network nodes. They aligned anchor nodes through neural networks and link users with dual learning and policy gradient. Some researchers also apply the graph embedding to the user identification. Man et al. [37] used the network embedding method to explore the network structure and identify users through cross-network mapping. Zhou et al. [38] proposed a nonpriority knowledge method FRUI-P based on social relationships. Liu et al. [25] embedded both the following relationship and follower relationship into the network structure to identify users. There are some existing works based on profile and user relationship. Zhang and Yu [39] combined user attributes and network structure to link potential multiple shared entities. Li et al. [10] combined user display name and social network information redundancy to identify users. Zhang et al. [40] extract features from display name, location in the profile, and relationships to identify users. Due to social network restrictions on data crawl, difficulty in obtaining multilevel relationships and the highly dynamic topology of social network [41], relationships will be sparse, incomplete, and unstable.

Relationships can be divided into the following relationships and follower relationships [25]. Due to the openness of social networks, any user can follow other users. A user may not know the person who is following him. Therefore, we only focus on the following relationships. Nowadays, due to user privacy settings and social network restrictions on data crawl, profiles, UGCs, and relationships may be missing or incomplete or fake in real social networks. This paper digged out a set of effective features that is extracted from public and available user data and can reduce the impact of user data which are missing or incomplete.

3. Problem Formulation

Suppose there are two social networks, Twitter and Facebook, represented by G^t and G^f. Use $G^t = \{V^t, E^t\}$ to define social network G^t, where V^t represents the set of all user accounts and E^t represents the set of relationships. User data of user v_i^t include profile, user-generated contents, and relationships. His profile includes display name name_i^t, location loc_i^t, user ID id_i^t, and work education experience we_i^t. His user-generated contents UGC_i^t includes locations UGCL_i^t, organizations UGCO_i^t, and URLs UGCU_i^t. His relationships include the following relationships and follower relationships. The definition of social network G^f is the same as G^t. As shown in Figure 1, we can define user identification across social networks as follows.

User identification: determine whether the user v_i^t in the social network G^t and the user v_k^f in the social network G^f are the same natural person in reality. If they belong to the same natural person, then the user v_i^t and the user v_k^f are called anchor users.

As shown in Figure 2, this paper mainly solves the user identification between two popular social networks, that is, to determine whether two user accounts from two social networks belong to the same natural person. Of course, this method can also be applied to user identification between multiple social networks. The dataset in this paper contains a part of ground truth, that is, anchor link users. We use $A = \{(v_i^t, v_k^f), v_i^t \in V^t, v_k^f \in V^f\}$ to define anchor users. User identification can also be defined as judging whether the user v_i^t and the user v_k^f are anchor users (v_m^t, v_n^f).

4. Model and Solution Framework

The framework proposed in this paper is mainly used for user identification when profiles, UGCs, and relationships are missing or incomplete. Firstly, we introduce the framework as a whole. Then, we specifically introduce the feature extraction methods. Finally, we introduce the fusion classifier machine learning-based user identification method.

4.1. MUIUI Framework.

4.1. MUIUI Framework. The MUIUI framework includes data crawl and storage module, feature extraction module, and detection module. The MUIUI is shown in Figure 3.

The data crawl and storage module mainly collects user data from Twitter and Facebook and stores it in the MySQL database. This paper uses multiprocess crawlers to crawl user data from Twitter and Facebook. The user data include profile, UGCs, and relationships.

The feature extraction module mainly extracts effective features from multiple user information, which extracts from user data. We obtain fourteen features from a display name and use them as the similarity of display name. The named entity recognition method is used to obtain all locations and organizations from UGCs and profile. We use entity link method to disambiguate and integrate them using the entity link method. We extract all URLs from UGCs and profile. And, extract organizations from the work education experience and combine them with the following relationships to calculate the similarity of the following organizations. We propose several algorithms to measure the similarity of the display name, all locations, all organizations, location in profile, all URLs, following organizations, and user ID, respectively. Combining the above features, a 20-dimensional feature vector is finally obtained.

The 20-dimensional feature vector is input to the detection module to perform user identification. In fact, the detection module uses a fusion classifier. We use the stacking method to fuse three base classifiers which have better performance. The output result of detection module is anchor users or nonanchor users.

4.2. Feature Extraction. Generally, user data contain multiple user information. We can extract several effective features from it. In the following, we exploit multiple user information from network G^t and G^f.

4.2.1. Similarity of Display Name. The display name is closely related to the user. It may not be unique in social networks. At present, some existing works only use the display name as the only attribute for user identification [11, 13, 14]. Compared with other attributes, the display name is easier to obtain. Nevertheless, the user can change the display name at will. The robustness of user identification

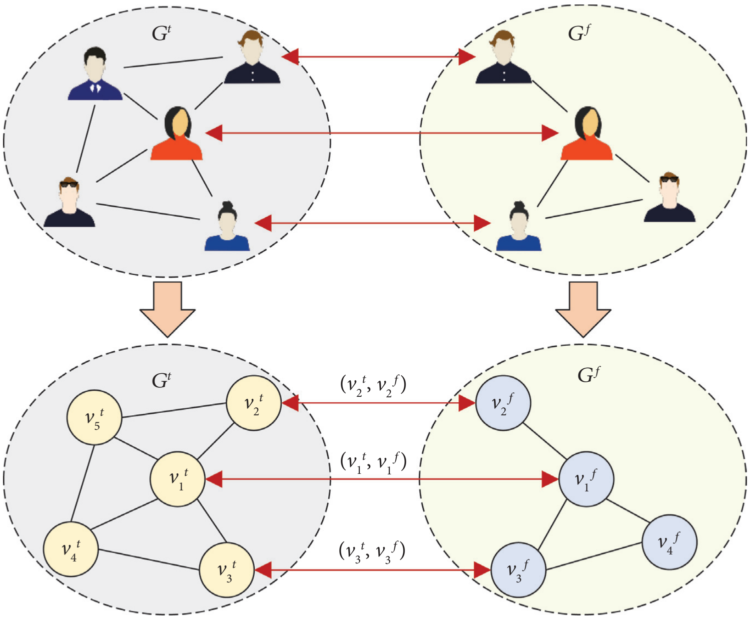
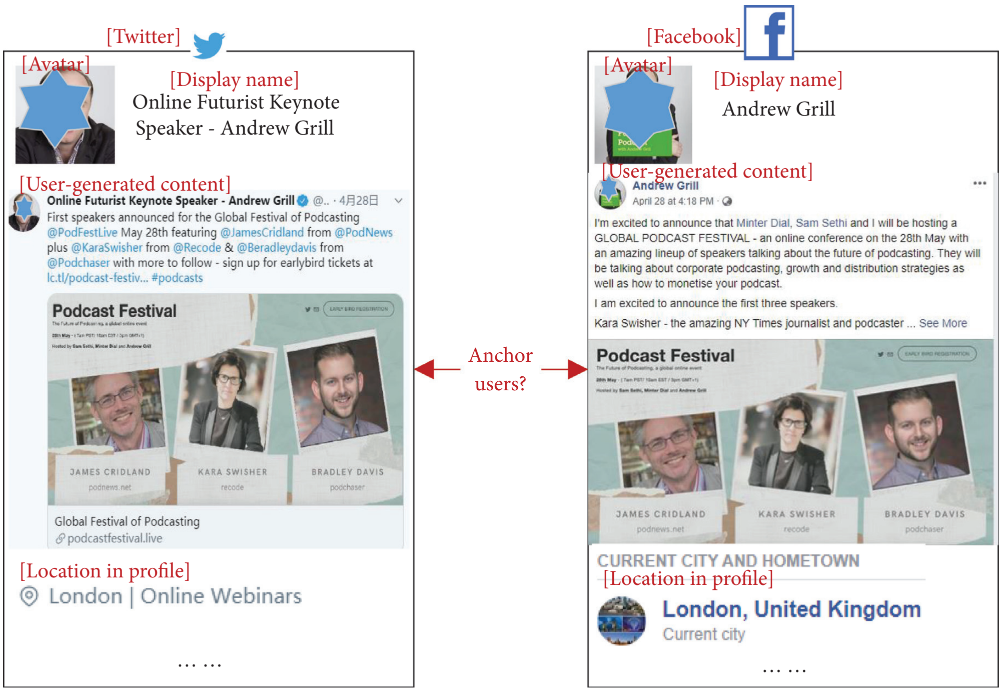


FIGURE 1: Illustration of user identification across G^t and G^f.



Online Futurist Keynote Speaker - Andrew Grill @.. · 4月28日

First speakers announced for the Global Festival of Podcasting @PodFestLive May 28th featuring @JamesCridland from @PodNews plus @KaraSwisher from @Recode & @Beradleydavis from @Podchaser with more to follow - sign up for earlybird tickets at lc.tl/podcast-festiv... #podcasts

Global Festival of Podcasting
podcastfestival.live

Andrew Grill
April 28 at 4:18 PM ·

I'm excited to announce that Minter Dial, Sam Sethi and I will be hosting a GLOBAL PODCAST FESTIVAL - an online conference on the 28th May with an amazing lineup of speakers talking about the future of podcasting. They will be talking about corporate podcasting, growth and distribution strategies as well as how to monetise your podcast.

I am excited to announce the first three speakers.

Kara Swisher - the amazing NY Times journalist and podcaster ... See More

FIGURE 2: Illustration of user identification across social networks.

based on display name is poor. Li et al. [11] extracted 14 features from the display name. This paper uses their method to obtain features vector X_{ik}^{name} from two display name name_i^t and name_k^f of user v_i^t and v_k^f. We use it as similarity of display name.

4.2.2. Similarity of All Locations and Similarity of All Organizations. In social networks, users may disclose their location in profile, work education experience, and UGCs. The work education experience is filled in by the user and is closely related to the user. The work education experience includes organizations, such as the company where the user works and the school where the user studies. Some social networks include work education experiences directly in the profile (such as LinkedIn and Facebook), and some social networks work education experiences are hidden in the profile (such as Twitter). This paper mainly analyzes the two social networks, Twitter and Facebook. So, for Twitter, we use their introductions as the work education experiences. The content of the UGCs also contains much-hidden

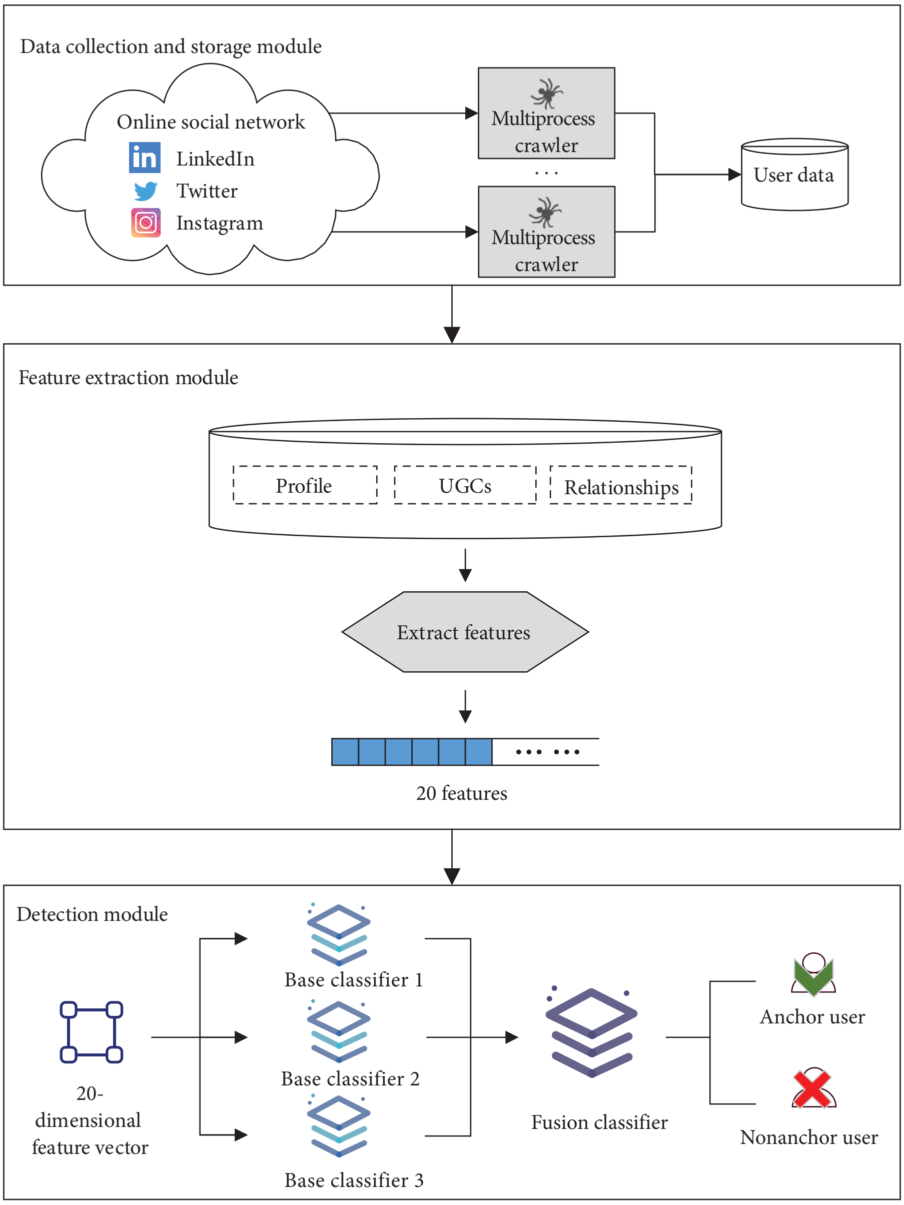


Figure 3: MUIUI framework diagram.

information. For example, locations related to the user, URLs shared by the user, and organizations that the user is concerned about. Named entity recognition can identify named entities from text data. This paper uses named entity recognition to obtain a set of locations and organizations from the content of the UGCs and work education experience. All locations include the location in the profile and the locations involved in the UGCs. Meanwhile, all organizations include the organizations included in the work education experience and the organizations involved in the UGCs.

Since all locations and organizations are closely related to the users themselves, all locations and organizations involved in user public information in different social networks will overlap. Moreover, the more a user mentions the location and organization, the more important it is. The same entity may have many aliases and named entity recognition may also be wrong. The entity link method can solve the above problems. All recognized locations and organizations are mapped to the Wikipedia entry IDs, where names pointing to the same entity are mapped to the same ID. Furthermore, delete entities that do not exist in Wikipedia entries to improve accuracy. This paper uses the named entity recognition method provided by the spacy (https://spacy.io/) library and entity link method provided by the entity link open-source framework Dexter (https://dexter.isti.cnr.it/). The Dexter uses English Wikipedia to implement entity link.

For user v_i^t and user v_k^f, the similarity of all locations and similarity of all organizations can be calculated as follows:

Step 1: for user v_i^t, a set of locations UGCL_i^t and a set of organizations UGCO_i^t are obtained from the content of the UGCs through named entity recognition. Similarly, we obtain a set of locations UGCL_k^f and a set of organizations UGCO_k^f of user v_k^f.

Step 2: for user v_i^t, we obtain a set of locations and a set of organizations from work education experience through named entity recognition. Then, we merge them with the two sets obtained in step 1 to obtain a new set of locations $\mathrm{LOC}_i^t = \{\mathrm{UGCL}_i^t, \mathrm{loc}_i^t\}$ and a new set of organizations $\mathrm{ORG}_i^t = \{\mathrm{UGCO}_i^t, \mathrm{WEO}_i^t\}$. Similarly, we obtain a new set of locations LOC_k^f and a new set of organizations ORG_k^f of user v_k^f.

Step 3: use entity link method to map LOC_i^t and ORG_i^t of user v_i^t into the location ID set LID_i^t and the organization ID set OID_i^t of user v_i^t. Similarly, location ID set LID_k^f and the organization ID set OID_k^f of v_k^f of user v_k^f are obtained.

Step 4: for each $\mathrm{lid}_{im} \in \mathrm{LID}_i^t$ and $\mathrm{lid}_{kn} \in \mathrm{LID}_k^f$, we calculate the weight λ_{im}^t of location ID lid_{im} and the weight λ_{kn}^f of location ID lid_{kn}. For each $\mathrm{oid}_{im} \in \mathrm{OID}_i^t$ and $\mathrm{oid}_{kn} \in \mathrm{OID}_k^f$, we calculate the weight μ_{im}^t of organization ID oid_{im} and the weight μ_{kn}^f of organization ID oid_{kn}.

Step 5: calculating $\mathrm{sim}_{\mathrm{loc}}$ and $\mathrm{sim}_{\mathrm{org}}$ by equations (1) and (2),

$$\mathrm{sim}_{\mathrm{loc}} = \sum_{\mathrm{lid}_{im} \in \mathrm{LID}_i^t, \mathrm{lid}_{kn} \in \mathrm{LID}_k^f} \lambda_{im}^t * \lambda_{kn}^f, \tag{1}$$

$$\mathrm{sim}_{\mathrm{org}} = \sum_{\mathrm{oid}_{im} \in \mathrm{OID}_i^t, \mathrm{oid}_{kn} \in \mathrm{OID}_k^f} \mu_{im}^t * \mu_{kn}^f, \tag{2}$$

where λ_{im}^t is the frequency of lid_{im} in LID_i^t and λ_{kn}^f is the frequency of lid_{kn} in LID_k^f. μ_{im}^t is the frequency of oid_{im} in OID_i^t and μ_{kn}^f is the frequency of oid_{kn} in OID_k^f.

4.2.3. Similarity of Location in the Profile. The location in the profile may be his/her current city or his/her hometown. It is more accurate than the location information extracted from the content of UGCs. Therefore, the similarity of location in the profile is taken as one feature. The profile's location filled in by the same user in different social networks should be closely related [40]. However, there are many aliases for the same location. This paper uses the API provided by pickpoint (https://app.pickpoint.io/) to convert location names into their latitude and longitude. The similarity of location in the profile is calculated based on the latitude and longitude of locations and is expressed by equation (5):

$$ll\left(\mathrm{loc}_k^f, \mathrm{loc}_i^t\right) = \sqrt{\sin^2\left(\frac{\mathrm{lat}_i^t - \mathrm{lat}_k^f}{2}\right) + \cos\left(\mathrm{lat}_i^t\right)\cos\left(\mathrm{lat}_k^f\right)\sin^2\left(\frac{\mathrm{lon}_i^t - \mathrm{lon}_k^f}{2}\right)}, \tag{3}$$

$$d\left(\mathrm{loc}_k^f, \mathrm{loc}_i^t\right) = 2R \times \arcsin\left(ll\left(\mathrm{loc}_k^f, \mathrm{loc}_i^t\right)\right), \tag{4}$$

$$\mathrm{sim}_{\mathrm{home}} = 1 - \frac{d\left(\mathrm{loc}_k^f, \mathrm{loc}_i^t\right)}{C}, \tag{5}$$

where $d(\mathrm{loc}_k^f, \mathrm{loc}_i^t)$ in equation (4) can be measured by equation (3), loc_i^t and loc_k^f represent the location in profile of user v_i^t and user v_k^f, respectively, lat_i^t and lat_k^f are the latitudes of loc_i^t and loc_k^f, respectively, lon_i^t and lon_k^f are the longitudes of loc_i^t and loc_k^f, respectively, and C is a constant, mainly used to normalize the value of $d(\mathrm{loc}_k^f, \mathrm{loc}_i^t)$ (the value of C is 19,860).

4.2.4. Similarity of all URLs. UGCs often include some URLs. These URLs may be the links of UGCs on other social networks, or the links that the user is interested in, or the links related to work education experiences of the user. This paper finds that users may share the same URLs on different social networks. Users may fill in the URL in their profiles, which are often closely related to users. It may be the company web page URL, or the personal web page URL, or homepage URLs of other social networks. Based on these extracted URLs, the similarity of all URLs can be calculated.

We use a method similar to Agarwal's URL extraction methods [12] to extract URLs' set UGCU_i^t and UGCU_k^f from the profile and UGCs, respectively. The calculation method of $\mathrm{sim}_{\mathrm{URL}}$ is shown in equation (6):

$$\mathrm{sim}_{\mathrm{URL}} = \sum_{\mathrm{URL} \in \mathrm{UGCU}_i^t \cap \mathrm{UGCU}_k^f} \gamma^t * \gamma^f, \tag{6}$$

where γ^t and γ^f represent the number of occurrences of the URL in URLs' set UGCU_i^t and URLs' set UGCU_k^f, respectively. URL belongs to the intersection of UGCU_i^t and UGCU_k^f.

4.2.5. Similarity of the following Organizations. Some social networks divide relationships into following relationships and follower relationships, such as Twitter. Following relationships refer to other users that the target user is following. Meanwhile, follower relationships refer to other users following the target user [25]. Due to the openness of social networks, anyone can become a user's follower. Therefore, we use following relationships and work education experiences to calculate the similarity of the following organizations. The work education experience was introduced in Section 4.2.2. Work education experience includes the organizations where the user works or studies, and these organizations often have their official social accounts in social networks. This paper found that users often follow the official social accounts of organizations that work or study.

This paper mainly analyzes two social networks, Twitter and Facebook. We suppose Twitter is a social network G^t and Facebook is a social network G^f. Because different social networks contain different user information, this paper extracts the organization from the work education experience of Facebook users and obtains the following relationships from Twitter users. Firstly, we extract the homepage URLs from the following users on Twitter and use the entity recognition method to extract the organizations from work education experiences on Facebook. Secondly, we use Google's advanced search method to obtain the official accounts' homepage URLs of the organizations on Twitter (for example, we need to obtain the official account of Apple on Twitter. Google search method is Apple + site: twitter.-com). Finally, calculate the similarity of following organizations. For user v_i^t and user v_k^f, the similarity of the following organizations' detailed algorithm is shown in Algorithm 1.

4.2.6. Similarity of User ID. The user ID can uniquely identify a user in the social network. In Twitter and Facebook, the initial value of the user ID is usually automatically generated by the social network, and the initial user ID has a strong correlation with the user's display name. The user can also modify it to a familiar string, but it must be unique. Some research [12] found that user ID can be used for user identification. Therefore, this paper takes the similarity of user ID as one classification feature. The user ID is usually a short string composed of numbers, letters, and underscores so that the string similarity calculation method can be used. This paper uses the Jaro–Winkler algorithm, which is often used to calculate English names' similarity. This algorithm increases the initial characters' weight and makes the string similarity more dependent on the initial part of the string. For user v_i^t and user v_k^f, the calculation method of $\mathrm{sim}_{\mathrm{userid}}$ is

$$d_j = \frac{1}{3}\left(\frac{m}{\left|id_i^t\right|} + \frac{m}{\left|id_k^f\right|} + \frac{m-t}{m}\right), \tag{7}$$

$$\mathrm{sim}_{\mathrm{userid}} = d_j + L \cdot p\left(1 - d_j\right), \tag{8}$$

where id_i^t and id_k^f represent the user ID of user v_i^t and user v_k^f, respectively. m is the number of matching characters and t is the number of transpositions. $|\mathrm{id}|$ is the length of user ID and d_j is the Jaro similarity for user ID id_i^t and user ID id_k^f. L is the length of common prefix at the start of the string up to a maximum of four characters and p is a constant scaling factor for how much the score is adjusted upwards for having common prefixes (the value of p is 0.1 in Jaro–Winkler).

4.3. Fusion Classifier. For the same dataset, the effects of different classifiers will also vary. Zhang et al. [40] use logistic regression (LR) and multilayer perceptron (MLP) classifiers to user identification. Liu et al. [42] use support vector machine (SVM) as the model classifier. Zafarani and Liu [43] use logistic regression (LR) as the model classifier. Li et al. [10] use gradient boosting (GB) classifier and tune the parameters of GB to user identification. Li et al. [11] use seven supervised machine learning models and tested them on the training set. Finally, the best model logistic regression with built-in cross-validation (LRCV) is selected as the classifier. These prove that base classifiers can already solve the classification problem well. Li et al. [4] performed ten cross-validations on the classification effect of 10 base classifiers and selected three better base classifiers to construct the fusion classifier. It also proves that the fusion classifier is generally better than the base classifier.

This paper mainly uses a supervised machine learning model to identify anchor users based on the above features. This paper uses 13 classifiers as the base classifiers, including multinomial Naive Bayes (MNB), Gaussian Naive Bayes (GNB), logistic regression (LR), logistic regression with built-in cross-validation (LRCV), support vector machine (SVM), Gaussian process classification (GPC), k-nearest neighbor (KNN), stochastic gradient descent (SGD), multilayer perceptron (MLP), decision tree (DT), random forest (RF), GraBoosting (GraB), and AdaBoost (AdaB). Then, we select three base classifiers with a better performance. Finally, the stacking method is used to fuse three base classifiers to obtain a fusion classifier.

5. Experimental Evaluation

5.1. Experimental Dataset. This paper focuses on two popular social networks Twitter and Facebook. We expanded the raw dataset which are those proposed in [21–25] and crawled during November 2012 [9]. We screened the users in the raw dataset who are still alive and took them as positive samples. We re-crawl 2397 pairs of Twitter and Facebook users in the raw dataset. As a result, 1292 pairs of Twitter and Facebook user accounts were found as still alive. To improve the classifier's performance, 1292 pairs of negative samples are added to the dataset, and half of the negative samples have similar display names to the positive samples. These 2584 pairs of samples are used as the experimental dataset.

We developed multiprocess crawlers to obtain the profiles of the dataset in December 2019 and to obtain UGCs and relationships of the dataset before January 2020, until the limits of the social network. The UGCs can be divided into original and repost. In this paper, we consider the reposted contents to be part of the UGCs, and the same content reposted multiple times will only be regarded as once. Both Twitter and Facebook users in the dataset are native speakers of English.

5.2. Evaluation Metrics. In the experiments, accuracy, recall, precision, and F1 score are used to evaluate the framework. In this paper, positive samples indicate anchor users, and negative samples indicate nonanchor users.

A confusion matrix is shown in Table 1. TP is the number of samples whose predicted and actual values are both positive. TN is the number of samples whose predicted and actual values are both negative. FN is the number of samples whose predicted is negative but is actually positive. FP is the number of samples whose predicted is positive but is actually negative.

Accuracy (ACC) is the ratio of correct predictions in all samples and is expressed by equation (9):

$$\text{accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}}. \tag{9}$$

Recall (REC) is the ratio of the both predicted and actual are positive samples in all actual samples and is expressed with equation (10):

$$\text{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{10}$$

Precision (PRE) is the ratio of the both predicted and actual are positive samples in all predicted samples and is expressed by equation (11):

$$\text{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{11}$$

F1 score is the harmonic mean of precision and recall and is expressed with equation (12):

$$f1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{12}$$

Area under curve (AUC) is area under the ROC curve. AUC can evaluate two-class classifiers. If a classifier has larger AUC, the accuracy of the classifier will be higher.

5.3. Experiments and Analysis. To prove that the MUIUI is an effective user identification framework even when user data are incomplete or missing, this paper makes statistics on the missing and incomplete user data in the dataset, as shown in Table 2. The numerical value in Table 2 is the number of users whose user data are missing or incomplete. Missing information means that the user has not filled in the information or has not disclosed it. Incomplete information means that the user has disclosed and filled in the information, but only part of them can be obtained due to social network restrictions. The false locations are judged by whether location names can be converted into latitude and longitude. If location can be transformed, it is true. Besides, if a user fills in the location is "Earth" or other meaningless nouns, they will also be regarded as false information.

According to the statistics in Table 2, the user data in the dataset used in this paper are missing or incomplete, except for display names. This dataset is crawled from real social networks by multiprocess crawlers. It also proves that user data have varying degrees of missing, falsity, and incompleteness in real social networks. To evaluate the effectiveness of the MUIUI framework, we compare MUIUI with three existing methods: the method proposed by Li [11], the OPL method proposed by Zhang [15], and the ALLEN-LR method proposed by Zhang [40]. The experiments use the dataset introduced in Section 5.1, which has 1292 pairs of anchor users (positive samples) and 1292 pairs of nonanchor users (negative samples). The dataset includes 1881 Twitter users and 1305 Facebook users.

5.3.1. Comparison on Base Classifiers. Use 13 base classifiers to identify users based on dataset introduced in Section 5.1. The base classifiers include multinomial Naive Bayes (MNB), Gaussian Naive Bayes (GNB), logistic regression (LR), logistic regression with built-in cross-validation (LRCV), support vector machine (SVM), Gaussian process classification (GPC), k-nearest neighbor (KNN), stochastic gradient descent (SGD), multilayer perceptron (MLP), decision tree (DT), random forest (RF), GraBoosting (GraB), and AdaBoost (AdaB). These classifiers can be implemented through scikit-learn [44], and all the parameters use their default values. In the experiments, the ratio of positive sample to negative sample is 1 : 1, and the ratio of the training set to the test set is 2 : 1. These 13 base classifiers are tested with the retraining process, and the average results are shown in Figure 4.

According to the results of Figure 4, RF, Grab, and AdaB have the best performance. Grab and AdaB are strong classifiers. A strong classifier is a classifier with higher accuracy, and it works better than weak classifiers. Grab and AdaB belong to strong classifiers and other base classifiers belong to weak classifiers. This is why Grab and AdaB are significantly higher than other classifiers. For RF, if the number of trees (that is, the dimensions of features) is larger, the RF classification performance will be better. The features of this paper reach 20 dimensions, that is, the number of trees is large. So, the RF works better. Therefore, we choose RF, Grab, and AdaB as base classifiers and use the stacking method to construct a fusion classifier as the final classifier.

5.3.2. The Ratio of Positive Sample to Negative Sample. The ratio of positive sample to negative sample in the training dataset may affect user identification framework. In order to choose the ratio of positive sample to negative sample in the MUIUI, the following experiments are based on the ratio of 8 : 1, 6 : 1, 4 : 1, 2 : 1, 1 : 1, 1 : 2, 1 : 4, 1 : 6, and 1 : 8 to train the MUIUI and compare it with the method proposed by Li [11], the OPL method proposed by Zhang [15], and the ALLEN-LR method proposed by Zhang [40]. The results are shown in Figures 5(a)–5(d).

According to the results in Figure 5(a), the accuracy first drops and then rises. Because the number of samples is the smallest at 1 : 1, the accuracy reaches a minimum at 1 : 1. From

```
Input: the following users $FL_i^t$ of user $v_i^t$, the work education experiences $we_k^f$ of user $v_k^f$.
Output: $sim_{org-follow}$.
$UURL_i^t \Leftarrow$ the homepage URLs extracted from the following users $FL_i^t$
$UURL_k^f = \varnothing$;
$WEORG_k^f \Leftarrow$ the organizations extracted from work education experiences $we_k^f$ by named entity recognition
for each $weorg_{kn}^f \in WEORG_k^f$ do
    $uurl_{kn} \Leftarrow$ the official account's homepage URL of $weorg_{kn}^f$ on twitter obtained by using Google's advanced search method
    $UURL_k^f = UURL_k^f + uurl_{kn}$;
end
$sim_{org-follow} = |UURL_i^t \cap UURL_k^f|$
```

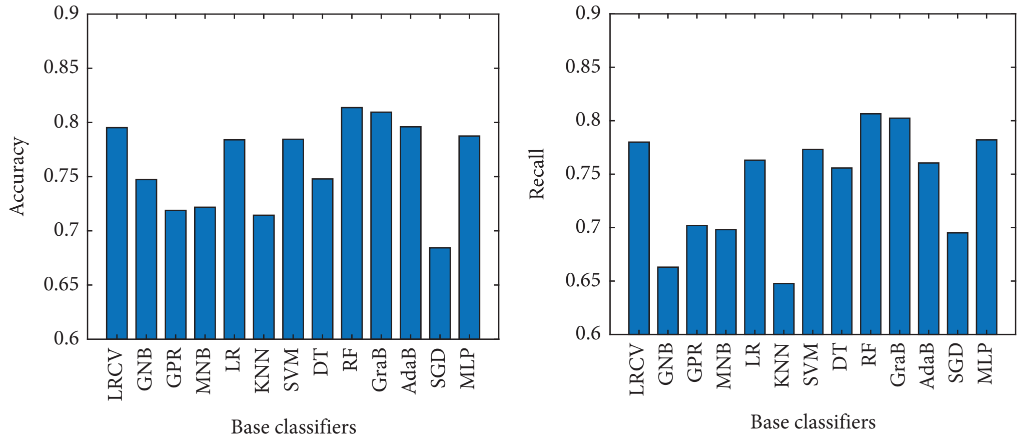
Algorithm 1: Similarity of following organizations.

Table 1: Illustration of confusion matrix.

Actual values	Predicted values	
	Positive samples	Negative samples
Positive samples	TP	FN
Negative samples	FP	TN

Table 2: Statistics on dataset.

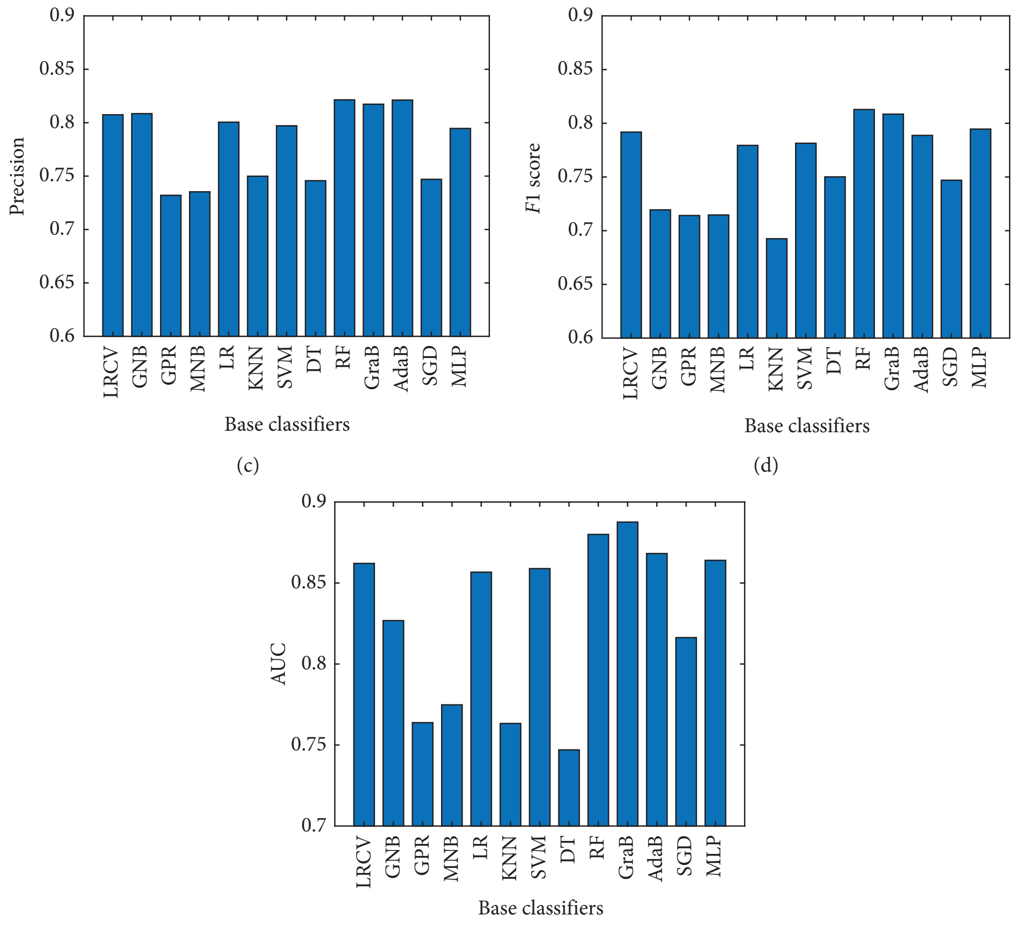
Social network	Missing display name	Missing or false location	Missing user-generated content	Missing relationship	Incomplete relationship
Twitter	0	480	62	219	769
Facebook	0	387	0	518	3



(a)

(b)

Figure 4: Continued.

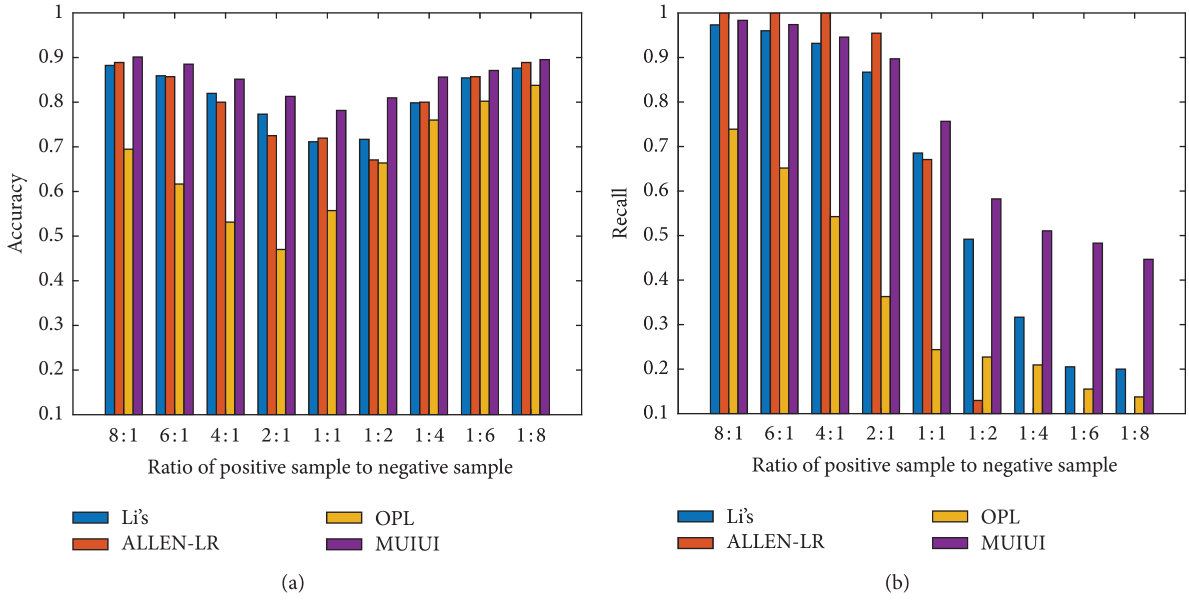


(c)

(d)

(e)

FIGURE 4: Performance comparison of 13 base classifiers. (a) Accuracy. (b) Recall. (c) Precision. (d) F1 score. (e) AUC.



(a)

(b)

FIGURE 5: Continued.

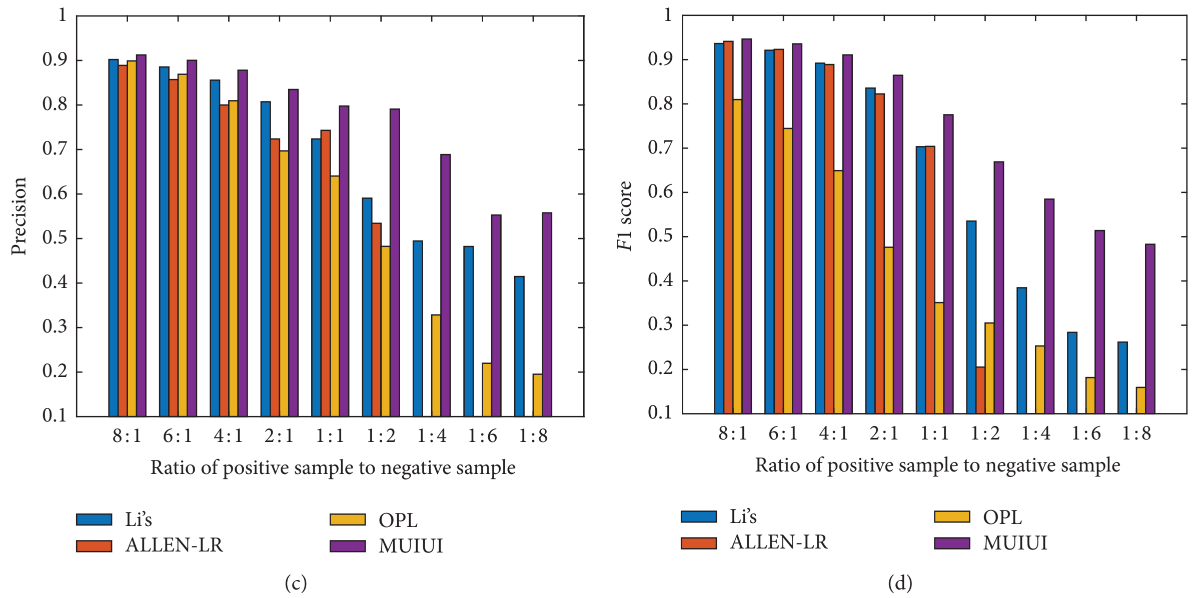


Figure 5: Results with different ratios of positive sample to negative sample. (a) Accuracy. (b) Recall. (c) Precision. (d) *F*1 score.

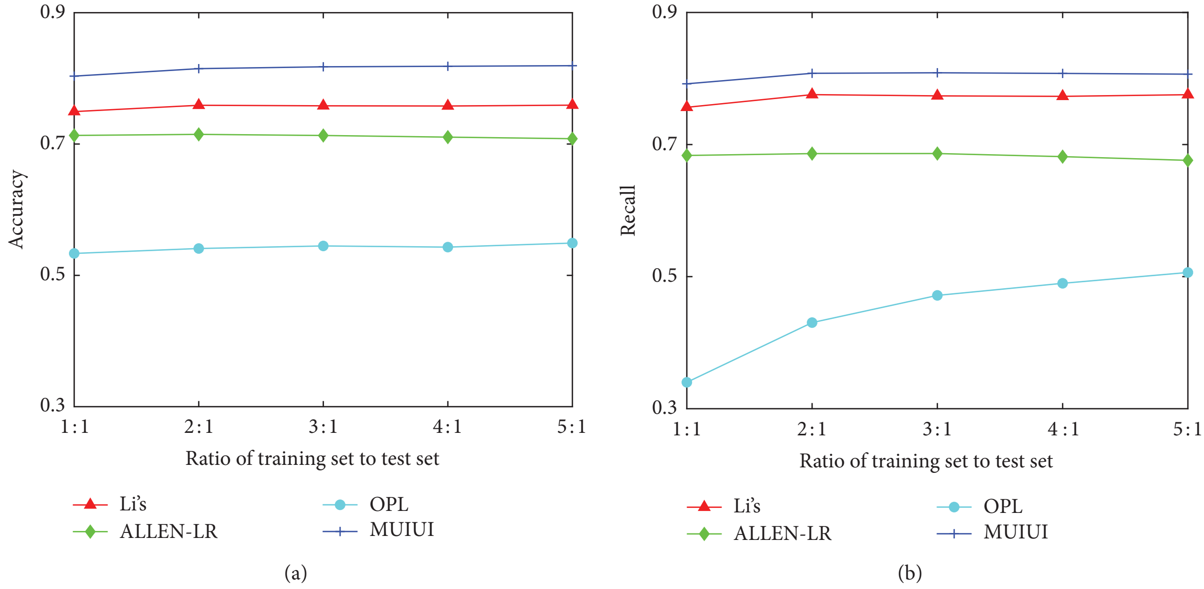


Figure 6: Continued.

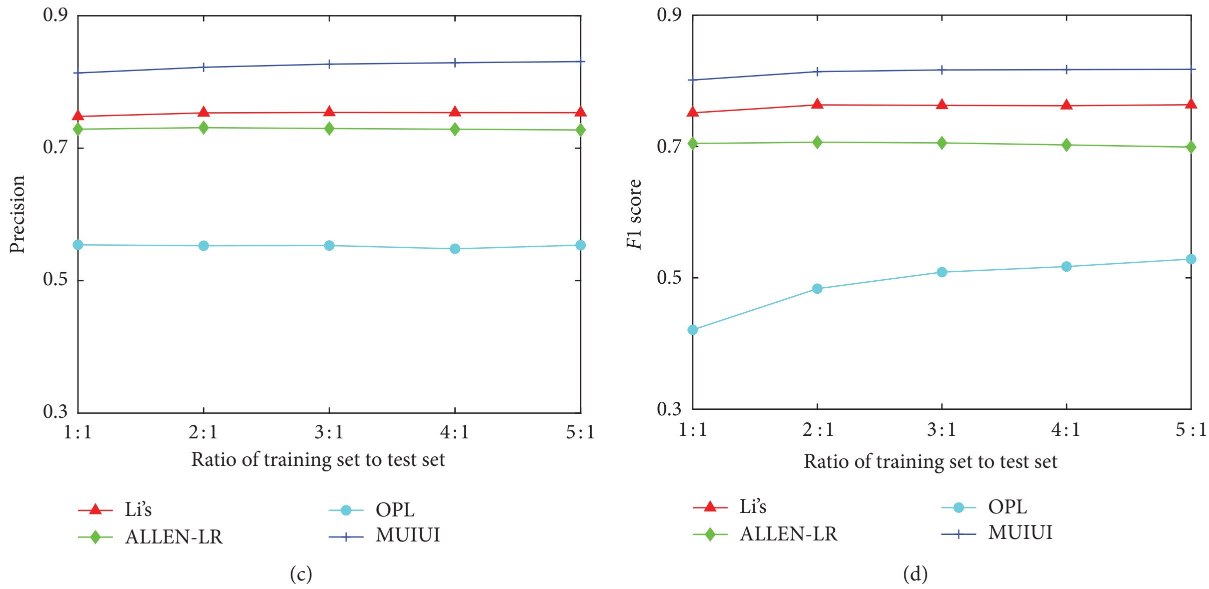


Figure 6: Results with different ratios of training set to the test set. (a) Accuracy. (b) Recall. (c) Precision. (d) *F*1 score.

1 : 1 to both ends, the number of samples increases, and the accuracy is getting higher and higher. The accuracy includes correctly predicted positive and negative samples. The more actual positive samples, the more positive samples are accurately predicted, and it is same for negative samples. The more the samples, the higher the accuracy. Therefore, the accuracy will first decrease and then increase. As shown in Figures 5(b)–5(d), when the proportion of positive samples decreased, the recall, precision, and F1 score also decreased. If training dataset has more positive samples, the classifier will learn more features of the positive samples and predict the positive samples more accurately. Leading to some negative samples are predicted to positive samples.

It can be seen from Figure 5 that the ALLEN-LR method has a higher recall than the method in this paper when positive samples are more than negative samples. However, when negative samples are more than positive samples, the performance of ALLEN-LR drops sharply. When the ratio of positive sample to negative sample is 1 : 4, 1 : 6, and 1 : 8, the recall, precision, and F1 score are almost zero. It shows that ALLEN-LR may judge some negative samples as positive samples. Based on this situation, the F1 score can evaluate the model better. According to Figure 5(d), MUIUI is stable and superior to other methods at different ratios. Because the cost of obtaining positive samples is too high, this paper chooses the ratio of 1 : 1 to construct the dataset.

5.3.3. The Ratio of the Training Set to the Test Set. To more fully illustrate the effectiveness of MUIUI, the following experiments are based on the ratio of the training set to the test set. Different ratio experiments are carried out 100 sampling verifications, and the average of 100 verification results are taken as the final results. According to the results, the accuracy, recall, precision, and *F*1 score of different frameworks are drawn.

Figures 6(a)–6(d) show that the MUIUI has higher indicators than the other three methods under different ratios. At the same time, it can be concluded that the larger the proportion of the training set is, the better the four methods perform.

Li's [11] method only extracts 14 features based on the display name, and there are no missing display names in the dataset. This is the only method without missing user data. The ALLEN-LR method [40] extracts features from the display name, locations in the profile of a user and his/her friends, and the multilayer relationships. It uses the LR classifier to perform user identification. Because the ALLEN-LR method relies heavily on relationships and needs locations in the profile of a user and his/her friends are relatively complete. However, the relationships in our dataset are incomplete, and the location in the profile is partially missing. When the data are partially missing or incomplete, the performance of ALLEN-LR is not ideal. Even if the proportion of the training set increases, it will not help the method. The OPL method [15] proposes methods to complete similarity of the display name, the similarity of profile photo, the similarity of location in profile, the similarity of text in profile, the similarity of URL in the profile, the popularity of the user, and the language user used. These seven features are used for user identification. Because profiles and relationships of some users are missing or incomplete in our dataset, the performance of OPL is also nonideal. It proves that MUIUI can reduce the impact of user data which are missing or incomplete.

6. Conclusion and Future Works

User identification has attracted extensive attention in academic circles, which can be used for friend recommendation, user privacy protection, and advertising recommendation. Due to user privacy settings and social network restrictions on data crawl, user data may be missing and incomplete in real social networks. The features extracted in previous research may be sparse. In order to solve these problems, we extracted effective features from public and available user data, which can reduce the impact of these problems. Firstly, we developed multi-process crawlers to obtain the latest user data of the dataset. Then, we used named entity recognition and entity linking to obtain and integrate locations and organizations from profiles and UGCs and extracted URLs from UGCs. We developed several algorithms to measure the similarity of the display name, all locations, all organizations, location in profile, all URLs, following organizations, and user ID, respectively. Finally, we proposed a fusion classifier machine learning-based user identification method. We verified the MUIUI framework on the dataset we crawled and the results indicate that the performance is better than that of existing representative works.

Popular social networks LinkedIn and Instagram also contain user data. Our work will be extended to these social networks in the future. We will introduce more effective features into the user identification method, such as user hotspot topics detection, trajectory analysis, and face perception of profile photos. These methods may improve the performance of user identification.

Data Availability

The data supporting this paper are from previously reported studies and datasets, which have been cited. The processed data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. U19A2081, 61802270 and 61802271). In addition, this work is also partially supported by Joint Research Fund of China Ministry of Education and China Mobile Company (no. CM20200409), and Fundamental Research Funds for the Central Universities (no. 2020SCUNG129).

References

[1] Statista, “Global social networks ranked by number of users 2020,” 2020, https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.

[2] R. Tang, S. Jiang, X. Chen, H. Wang, W. Wang, and W. Wang, “Interlayer link prediction in multiplex social networks: an iterative degree penalty algorithm,” *Knowledge-Based Systems*, vol. 194, p. 105598, 2020.

[3] J. Zhang, P. S. Yu, and Z.-H. Zhou, “Meta-path based multi-network collective link prediction,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1286–1295, New York, NY, USA, 2014.

[4] Y. Li, Z. Zhang, Y. Peng, H. Yin, and Q. Xu, “Matching user accounts based on user generated content across social networks,” *Future Generation Computer Systems*, vol. 83, pp. 104–115, 2018.

[5] S. Huang, J. Zhang, L. Wang, and X.-S. Hua, “Social friend recommendation based on multiple network correlation,” *IEEE Transactions on Multimedia*, vol. 18, no. 2, pp. 287–299, 2016.

[6] J. Zhang, P. S. Yu, Y. Lv, and Q. Zhan, “Information diffusion at workplace,” in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Association for Computing Machinery*, pp. 1673–1682, New York, NY, USA, 2016.

[7] Y. Qu, S. Yu, L. Gao, W. Zhou, and S. Peng, “A hybrid privacy protection scheme in cyber-physical social networks,” *IEEE Transactions on Computational Social Systems*, vol. 5, no. 3, pp. 773–784, 2018.

[8] Y. Qu, S. Yu, W. Zhou, and Y. Tian, “Gan-driven personalized spatial-temporal private data sharing in cyber-physical social systems,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2576–2586, 2020.

[9] Q. Zhan, J. Zhang, P. Yu, and J. Xie, “Community detection for emerging social networks,” *World Wide Web*, vol. 20, no. 6, pp. 1409–1441, 2017.

[10] Y. Li, Z. Su, J. Yang, and C. Gao, “Exploiting similarities of user friendship networks across social networks for user identification,” *Information Sciences*, vol. 506, pp. 78–98, 2020.

[11] Y. Li, Y. Peng, W. Ji, Z. Zhang, and Q. Xu, “User identification based on display names across online social networks,” *IEEE Access*, vol. 5, pp. 17342–17353, 2017.

[12] A. Agarwal and D. Toshniwal, “Smpft: social media based profile fusion technique for data enrichment,” *Computer Networks*, vol. 158, pp. 123–131, 2019.

[13] J. Liu, F. Zhang, X. Song, Y.-I. Song, C.-Y. Lin, and H.-W. Hon, “What’s in a name? an unsupervised approach to link users across communities,” in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pp. 495–504, Rome, Italy, 2013.

[14] D. Liu, Q. Wu, W. Han, and B. Zhou, “User identification across multiple websites based on username features,” *Chinese Journal of Computers*, vol. 38, pp. 2028–2040, 2015.

[15] H. Zhang, M.-Y. Kan, Y. Liu, and S. Ma, “Online social network profile linkage,” in *Asia Information Retrieval Symposium*Springer, Berlin, Germany, 2014.

[16] X. Zhou, X. Liang, H. Zhang, and Y. Ma, “Cross-platform identification of anonymous identical users in multiple social media networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 411–424, 2016.

[17] Q. Xuan and T. Wu, “Node matching between complex networks,” *Physical Review E*, vol. 80, Article ID 026103, 2009.

[18] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, “Cosnet: connecting heterogeneous social networks with local and global consistency,” in *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1485–1494, Sydney, Australia, 2015.

[19] F. Zhou, L. Liu, K. Zhang, G. Trajcevski, J. Wu, and T. Zhong, “Deeplink: a deep learning approach for user identity linkage,” in *Proceedings of the 37th IEEE Conference on Computer Communications*, pp. 1313–1321, Honolulu, HI, USA, 2018.

[20] S. Sajadmanesh, H. R. Rabiee, and A. Khodadadi, “Predicting anchor links between heterogeneous social networks,” in *Proceedings of the IEEE/ACM International Conference on*

Advances in Social Networks Analysis and Mining (ASONAM), pp. 158–163, Assam India, 2016.

[21] X. Kong, J. Zhang, and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pp. 179–188, San Francisco, CA, USA, 2013.

[22] J. Zhang, X. Kong, and P. S. Yu, "Transferring heterogeneous links across location-based social networks," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining, Association for Computing Machinery*, pp. 303–312, New York, NY, USA, 2014.

[23] J. Zhang and S. Y. Philip, "Integrated anchor and social link predictions across social networks," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 2215–2132, Buenos Aires, Argentina, 2015.

[24] J. Zhang, "Social network fusion and mining: a survey," 2018, https://arxiv.org/abs/1804.09874.

[25] L. Liu, W. K. Cheung, X. Li, and L. Liao, "Aligning users across social networks using network embedding," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 1774–1780, Palo Alto, CA,USA, 2016.

[26] R. Zafarani and H. Liu, "Connecting corresponding identities across communities," in *Proceedings of the Third International AAAI Conference on Weblogs and Social Media*, San Jose, CA, USA, 2009.

[27] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manils, "How unique and traceable are usernames?" in *Proceedings of the 11st International Symposium on Privacy Enhancing Technologies Symposium*, pp. 1–17, Waterloo, ON, Canada, 2011.

[28] Y. Li, Y. Peng, Z. Zhang, M. Wu, Q. Xu, and H. Yin, "A deep dive into user display names across social networks," *Information Sciences*, vol. 447, pp. 186–204, 2018.

[29] A. Acquisti, R. Gross, and F. D. Stutzman, "Face recognition and privacy in the age of augmented reality," *Journal of Privacy and Confidentiality*, vol. 6, pp. 1–20, 2014.

[30] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, "Linking users across domains with location data: theory and validation," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 707–719, Montreal, Canada, 2016.

[31] W. Chen, H. Yin, W. Wang, L. Zhao, W. Hua, and X. Zhou, "Exploiting spatio-temporal user behaviors for user linkage," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Association for Computing Machinery*, pp. 517–526, New York, NY, USA, 2017.

[32] X. Gao, W. Ji, Y. Li, Y. Deng, and W. Dong, "User identification with spatio-temporal awareness across social networks," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery*, pp. 1831–1834, New York, NY, USA, 2018.

[33] W. Chen, H. Yin, W. Wang, L. Zhao, and X. Zhou, "Effective and efficient user account linkage across location based social networks," in *Proceedings of the 34th IEEE International Conference on Data Engineering*, pp. 1085–1096, Paris, France, 2018.

[34] Y. Li, Y. Peng, Z. Zhang, H. Yin, and Q. Xu, "Matching user accounts across social networks based on username and display name," *World Wide Web*, vol. 22, no. 3, pp. 1075–1097, 2019.

[35] M. Motoyama and G. Varghese, "I seek you: searching and matching individuals in social networks," in *11th ACM International Workshop on Web Information and Data Management (WIDM 2009)*, pp. 67–75, Hong Kong, China, 2008.

[36] O. Goga, H. Lei, S. H. K. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira, "Exploiting innocuous activity for correlating users across sites," in *Proceedings of the 22nd International Conference on World Wide Web*, pp. 447–458.

[37] T. Man, H. Shen, S. Liu, X. Jin, and X. Cheng, "Predict anchor links across social networks via an embedding approach," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, vol. 16, pp. 1823–1829, Palo Alto, CA, USA, 2016.

[38] X. Zhou, X. Liang, X. Du, and J. Zhao, "Structure based user identification across social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1178–1191, 2018.

[39] J. Zhang and P. S. Yu, "Pct: Partial co-alignment of social networks," in *Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee*, pp. 749–759, Montreal, Canada, 2016.

[40] Y. Zhang, J. Fu, C. Yang, and C. Xiao, "A local expansion propagation algorithm for social link identification," *Knowledge and Information Systems*, vol. 60, no. 1, pp. 545–568, 2019.

[41] S. Peng, G. Wang, Y. Zhou et al., "An immunization framework for social networks through big data based influence modeling," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 6, pp. 984–995, 2019.

[42] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, "Hydra: Large-scale social identity linkage via heterogeneous behavior modeling," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 51–62, Snowbird, UT, USA, 2014.

[43] R. Zafarani and H. Liu, "Connecting users across social media sites: a behavioral-modeling approach," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 41–49, Chicago, IL, USA, 2013.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9934138, 15 pages
https://doi.org/10.1155/2021/9934138

Research Article

Trusted QoS Assurance Approach for Composite Service

Meng Cai,[1] **Ying Cui**,[2] **Yang Yu**,[1] **Ying Xue**,[1] **and Han Luo**[1]

[1] *School of Humanities and Social Sciences, Xi'an Jiaotong University, Xi'an 710049, China*
[2] *School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Meng Cai; mengcai@mail.xjtu.edu.cn

Received 18 March 2021; Revised 16 April 2021; Accepted 29 April 2021; Published 17 May 2021

Academic Editor: Hao Peng

The characteristics of service computing environment, such as service loose coupling, resource heterogeneity, and protocol independence, propose higher demand for the trustworthiness of service computing systems. Trust provisioning for composite services has become a hot research spot worldwide. In this paper, quality of service (QoS) planning techniques are introduced into service composition-oriented QoS provisioning architecture. By QoS planning, the overall QoS requirement of the composite service is decomposed into separate QoS requirement for every constituent atom service, the QoS level of which can subsequently be satisfied through well-designed service entity selection policies. For any single service entity, its QoS level is variable when the deployment environment or the load of service node changes. To mitigate the uncertainty, we put forward QoS preprocessing algorithms to estimate the future QoS levels of service entities with their history execution data. Then, based on the modeling of composite service and QoS planning, we design three algorithms, which include the time preference algorithm, cost/availability (C/A) preference algorithm, and Euclidean distance preference algorithm, to select suitable atomic services meeting the user's requirements. Finally, by combining genetic algorithm and local-search algorithm, we propose memetic algorithm to meet the QoS requirements of composite service. The effectiveness of the proposed methods by which the QoS requirements can be satisfied up to 90% is verified through experiments.

1. Introduction

With the enhancement of the interconnection, intercommunication, and interoperability of the information processing platform, as well as the improvement of the degree of integration, higher requirements are put forward for the reliable serviceability and service quality assurance of the service computing system [1]. The general information processing platform provides a consistent operating environment for all kinds of general information processing services and supports the intensive operation of all kinds of information services. At the same time, various information systems on the same platform, different services in the same information system, different service requests of different users, and different requests of the same user all have various requirements of service quality [2]. On the one hand, the unique characteristics of service computing, such as loose coupling, location transparency, and protocol independence, make it an ideal model for building various complex data processing and information service systems in the network environment. On the other hand, factors such as the looseness, independence, and heterogeneity of the computing environment also bring new technical challenges to the guarantee of nonfunctional attributes such as real-time performance, security, reliability, and availability of service computing. In the open and dynamic network environment, the composite service-oriented computing model needs to implement a reliable software architecture with high reliability, adaptability, and self-management capability [3].

Trusted computing integrates real-time, safe, reliable, and available computing theories in the traditional category and studies the interaction of system behaviors under nonfunctional attributes in a unified way, so that the behavior state can be monitored, the behavior results can be evaluated, and the abnormal behaviors can be controlled [4]. Quality of service (QoS) is a comprehensive index describing the nonfunctional characteristics of computer systems and communication networks, which is used to measure the satisfaction degree of using a service [5]. Both trusted computing and QoS aware computing focus on the multidimensional nonfunctional characteristics of the system, so it can be considered to represent the trusted attribute as the multidimensional QoS demand of the system, to provide credibility guarantee for the network service. It has become an important direction in the field of trusted computing to use the service quality theory to provide a guarantee for the trusted demand of service [6].

As the number of Web services shared over the network increases dramatically, the traditional choice of user services is based on QoS values published by vendors. In reality, the network environment shared by Web services is characterized by openness, instability influenced by the external environment, and unreliability of QoS value of services influenced by business competition. The incredibility of the QoS value of Web services and the uncertainty of the invocation of Web services make users unable to obtain Web service combinations according to actual requirements. Therefore, in the case of complex network environment and massive Web services, how to efficiently select a group of Web services that can meet users' functional requirements and have credibility from a large number of candidate services is a key scientific and technical problem to be solved in the field of service computing.

In this paper, we proposed an approach composed of QoS modeling, QoS prediction, QoS planning, and service selection, to guarantee the QoS of composite service. Firstly, we built the models to describe the trusted QoS of atomic service and trusted QoS of composite service, respectively. Secondly, we introduced the exponential smoothing method to evaluate the abilities of atomic services and found that the prediction of QoS information of atomic service by the exponential smoothing method was accurate and feasible. Based on the QoS estimation, we modeled QoS calculation rules of composite service. In addition, we proposed a series of service selection algorithms and used computational simulation to demonstrate that the Euclidean distance preference algorithm had a more balanced and excellent characteristic. Finally, we compared the memetic algorithm with random selection to meet the multidimensional QoS requirement of composite service.

2. Related Works

With the popularization and development of Web service technology, on the one hand, more and more users begin to complete various business processes by combining Web service, and the business process pattern is being promoted to the broader users. On the other hand, users no longer need specific Web service as the executors of tasks in the process owing to the appearance of a large number of Web services with the same or similar functions rather than automatically selecting Web service according to the needs and constraints of users.

To meet the specific application background and requirements, we usually need to combine multiple Web services according to certain granularity to form a service process with a specific structure and realize complete business logic in dealing with complex business. Facing this situation makes it important to evaluate and ensure the process of QoS.

When users select Web service, in addition to functional attributes, they can also set up constraints on some nonfunctional attributes, that is, by constraining QoS attributes to be selected. QoS can include attributes such as price, response time, availability, and reliability [7]. Providers of Web service also tend to give a range of QoS attribute values of the service, thus providing a reference for potential users.

In QoS sensitive business processes, the QoS of the entire process is also a feature of user concern except requiring the process to complete predefined tasks [8]. Therefore, how to make an effective choice in the alternative Web service to meet the requirements of users, so that the selected Web service can not only complete the task assigned by the process and meet the local constraints, but also cooperate with the Web service which completes other tasks in the process, is an urgent issue that needs to be solved and it is often referred to as QoS sensitive Web service composite issue.

At present, many scholars have studied the trusted computing technology and QoS technology in the complex distributed computing environment and achieved some results [9]. Existing research is often based on the idea of QoS guarantee, from the perspective of reliability, security, and other nonfunctional attributes of service computing research, especially through the combination of atomic services to maximize the needs of users [10, 11]. For Web service composition in the study, the researcher proposed the service selection method based on mathematical programming; service selection method based on genetic algorithm (GA); service selection method based on particle swarm optimization (PSO) algorithm, based on swarm intelligence computing; service selection method based on global QoS constraint decomposition; service selection method based on artificial intelligence theory; and many other kinds of service composition methods [12–14].

The literature [15] carries out a survey and research review in the field of Web services technology, clarifies the research motivation of service credibility evaluation, discusses the importance of atomic Web services credibility in the construction of service composition strategy, and classifies the credibility evaluation methods. Wang et al. [16] divided the process of service composition into three stages: service planning, service selection, and service binding, to reasonably and effectively evaluate the credibility of services, and investigated the credibility of services in each stage. Ardagna and Pernici [17] designed a service composition method based on mixed-integer programming when considering the possible local and global QoS constraints in service selection. Sun and Zhao [18] decomposed global QoS constraints into local QoS constraints through a mixed-integer programming method, thus transforming the service composition problem belonging to the global optimization problem into the local optimal service selection problem and reducing the complexity of the problem. Surianarayanan et al. [19] proposed a service composition method of constraint decomposition method to calculate the utility of composite services through the utility of component services and the utility constraints of component services of composite service constraints. The literature [20] decomposed service composition into two stages: constraint decomposition and service selection. Another study [21] proposed a service optimization combination method based on correlation graph and 0-1 linear programming. The service optimization combination method based on mathematical programming can effectively solve the problem of service optimization combination to a certain extent. However, when the problem scale is large, the calculation time of such a method is relatively long, and it cannot meet the real-time requirements of service composition.

Few researchers in the above studies distinguish the concept of service and service entity from the perspective of trusted QoS guarantee architecture. When making QoS planning, planners are faced with the underlying atomic service entity that provides specific service functions. In this system view, the requirement of trusted QoS for composite services can only be met through QoS planning based on the choice of atomic service entities. Moreover, the QoS embodied by the service entity at run time is often inconsistent with the QoS declared by the service entity, which makes this QoS planning method limited in guaranteeing the trusted QoS of the composite service at run time.

3. Trusted QoS for Service Modeling

In this paper, we refer to the composite service with trusted constraints of QoS described jointly by BPEL language and state diagrams as processes. We focus more on service entities than service for the process; what directly determines the process QoS is not the service entity of QoS, but the QoS; the planning of QoS is also based on the service (abstract service) of QoS, not the service entity of QoS.

Service composition is to combine some relatively simple services according to certain business process logic into complex composite services, thus providing more powerful and complete business functions. Based on the loose service combination, the customization of business processes can be realized, and personalized services can be provided to users, so that the business can adapt to changes [22].

Entities are defined as people, units, various computer hardware and software modules, or their multiple sets. Multiple sets are linear sum of integral coefficients of various elements.

Atomic service/service entity is defined as one entity that works or operates for another entity called service. Here, the work is for the service provider; the operation (computer and information receiving, sending, transforming, etc.) is for the computer hardware and software module that provides the service.

Composite service is defined as merging multiple small granularity services into one large granularity service. In the network architecture, the service combination is a combination of service providers.

Abstract service is defined as service functions provided by service entities. It can also be understood as an abstract proxy composed of atomic service entities with the same service function (but the nonfunctional capabilities of QoS will be different), which is used to represent all atomic services with this function.

With this concept, in the planning process of QoS, we can put forward reasonable requirements of QoS for each abstract service in the composite service, because the abstract service represents the operation status and QoS capability of all atomic service entities with the same function. In this paper, unless otherwise specified, we call abstract service "servic", which is different from atomic service (service entity).

When the business process changes, it is only necessary to reorganize the services that make up the business process or dynamically adjust the component services that make up the business process to quickly respond to the changes of the business process [23]. In the business process realized by service composition, both developers and end-users can combine the existing services to obtain new and complex services and form new or even dynamic business processes. This promotes the utilization of service, improves the reusability of service, and accelerates the development of application projects [24].

3.1. Trusted QoS for Atomic Service Modeling. There are many methods for modeling the trusted properties of atomic services. Generally speaking, it is divided into real time, availability, reliability, cost, and so on. In this paper, we use the following methods to model:

(i) Real time: it is used to measure the time characteristics of a service. A service call time t defined as follows: on the service caller side, the time difference between t_1 moment and t_2 moment from sending service request to receiving returned result is $|t_2 - t_1|$. t includes network transmission time. The real-time performance of service will be described by ternary < minimum. Minimum, average, and maximum are all elements of $R+$. Average represents the average service call time to access the service in history, and maximum represents the minimum service call time to access the service in history. The average access time can be calculated using the following formula: $T(s) = (\sum_{i=1}^{n} T_i(s)/n)$, where n is the number of times the service was observed in history, and $T_i(s)$ is the time at which the service was called in history.

The reason why this arithmetic average method has a predictive function is that it can always eliminate to a certain extent the influence of random changes caused by accidental interference on the QoS capability of atomic service. However, this method also conceals the fluctuating trend in the development of the atomic service itself.

Aiming at the limitation of this modeling method, we will use a weighted average method named exponential smoothing in the planning process of QoS. It means that the recent QoS characteristics of the atomic service have a greater impact on the future and the more up-to-date information reflected; the importance of historical data which predict the QoS capability of the atomic service should decrease from the close to the distant in time.

(ii) Availability: the availability of a service describes the probability that the service can be accessed correctly. The availability of service S, namely $A(S)$, is defined as a real number on $[0, 1]$, $A(S) \in [0, 1]$. Use the following method to measure the availability of services: $A(S) = T_{\theta}(S)/\theta$, where θ is a time constant in seconds and $T_{\theta}(S)$ is the time at which the service can be accessed correctly in the past θ seconds. In practical application, different θ values are selected for observation according to the access frequency of different services. For services with high frequency of access, we choose a smaller period of time for investigation. For services with low frequency of access, we choose a larger period of time for investigation.

(iii) Cost: it is the expense of using a service. The trusted cost of a service is described as a positive integer. The cost of atomic services is defined as an element on I+. In practical applications, the cost of a service is given by the provider of the service (or the cost is determined according to the time spent by the service user to calculate the resources and the use of the resources).

3.2. Trusted QoS for Composite Service Modeling. The trusted nature of the process has the same trusted aspect and description domain as the trusted nature of the service. The credibility of the composite service is determined by the credibility of its subservices and its composition mode. The operation function of composite service is composed of multiple subtasks in a certain way, and there are control dependency and data dependency between subtasks. For the execution process of composite services S, we can decompose it with reference to the method in [10] and describe the control dependence in the execution process by using state diagram. On this basis, we can use the method of [8] to generate a directed acyclic graph to describe its execution process. The node represents the subtask, and the direction of the edge describes the execution order of the task. Different subtasks may be provided by different subtasks. The execution plan of a service S is defined as $\langle G, P\rangle$, where G is a directed acyclic graph representing its control dependency during execution.$P = \{\langle t_1, s_1\rangle, \langle t_2, s_2\rangle, \ldots, \langle t_n, s_n\rangle\}$ describes the corresponding relationship between subtasks and subservices, and $\langle t_i, s_i\rangle$ indicates that t_i is completed by the service s_i.

Figure 1 is a typical implementation plan diagram. The process of service S can be broken down into multiple subtasks. The control dependency between tasks is described by the directed acyclic graph. The starting state is t_i, two parallel subtasks t_1 and t_2 are triggered simultaneously, the concurrent activities are synchronized at t_4, then task t_5 is started, and finally t_f is finished. The functions of these subtasks are provided by several services. $\langle t_i, s_i\rangle$ describes the corresponding relationship between subtasks and services. Among them, $<t_i, t_1, t_2, t_3, t_4, t_f>$ is a critical path (shown by the green squares in Figure 1 in the execution process), and the execution time of the critical path determines the execution time of the whole operation.

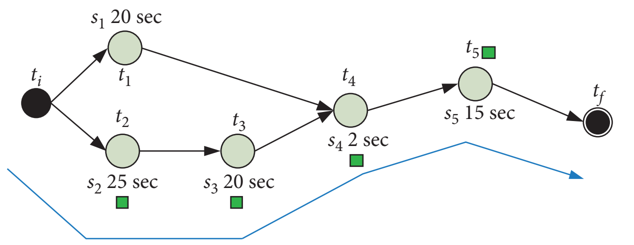


Figure 1: A schematic diagram of the typical implementation plan.

For a more complex business process of Web service, we can decompose it into recursive combinations of several basic institutions. Table 1 gives the QoS calculation rules for the flow of several basic structures. In this paper, we will explain and verify the proposed method with three common QoS attributes: time, cost, and availability.

Sequence structure (Sequence): the service process of sequence structure **W** is composed of service by m $(s_1, s_2, s_i, \ldots, s_m)$ in a certain order.

Time (T): the response time $T(W)$ of the service process **W** is composed of the sum of the response time of each service that constitutes the service process.

$$T(W) = \sum_{i=1}^{m} T(s_i). \quad (1)$$

Cost (C): the cost $C(W)$ of service process **W** is the sum of the costs of each service that constitutes the service process.

$$C(W) = \sum_{i=1}^{m} C(s_i). \quad (2)$$

Availability (A): the availability $A(W)$ of service process **W** is composed of the availability product of each service that constitutes the service process.

$$A(W) = \prod_{i=1}^{m} A(s_i). \quad (3)$$

Choice structure (Choice): the service process of choice structure **W** is composed of service by m $(s_1, s_2, s_i, \ldots, s_m)$ in a certain order.

For the selection structure, each selection branch is marked with the probability of being selected. For example, a process selection structure has two branches, its prices c_1 and c_2, respectively. The probability of being selected is p and q, respectively. The cost of the whole structure is calculated as follows: $pc_1 + qc_2$, s.t. $p + q = 1$.

Table 1: QoS calculation methods for composite service.

QoS dimension	Workflow			
	Sequence	Choice	Parallel	Loop
Time (T)	$\sum_{i=1}^{m} T\,(s_i)$	$\sum_{i=1}^{m} P_i * T\,(s_i)$	$\mathrm{Max}\{T\,(s_i)\}$, $i \in (1, 2, \ldots, m)$	$m * T\,(s)$
Availability (A)	$\sum_{i=1}^{m} A\,(s_i)$	$\sum_{i=1}^{m} P_i * A\,(s_i)$	$\prod_{i=1}^{m} A\,(s_i)$	$A\,(s)^m$
Cost (C)	$\sum_{i=1}^{m} A\,(s_i)$	$\sum_{i=1}^{m} P_i * C\,(s_i)$	$\sum_{i=1}^{m} C\,(s_i)$	$m * C\,(s)$

The initialization value of probability can be determined by the designer of business process, and then the information obtained by monitoring the process execution process is updated continuously. Accordingly, for the selection structure 1, . . ., *m with N* of branches, the probability of each branch is $\{p_1, p_2, \ldots, p_i, \ldots, p_m\}$, of which $\sum_{i=1}^{m} p_i = 1$. Usually, the QoS of this choice structure is the attribute value of each task and its corresponding branch probability, and then the sum.

Time (T): the response time $T(W)$ of the service process **W** is composed of the weighted sum of the response time probabilities of the service that constitutes the service process.

$$T\,(W) = \sum_{i=1}^{m} P_i * T\,(s_i). \tag{4}$$

Cost (C): the cost $C(W)$ of service process **W** is composed of the weighted sum of the cost probabilities of the service that constitutes the service process.

$$C\,(W) = \sum_{i=1}^{m} P_i * C\,(s_i). \tag{5}$$

Availability (A): the availability $A(W)$ of service process **W** is composed of the weighted sum of the cost probabilities of the service that constitutes the service process.

$$A\,(W) = \sum_{i=1}^{m} P_i * A\,(s_i). \tag{6}$$

Parallel structure (Parallel): the service process of parallel structure **W** is composed of service by m $(s_1, s_2, s_i, \ldots, s_m)$ in synchronization.

Time (T): the response time $T(W)$ of the service process **W** is composed of the maximum value of the response time that constitutes the service process. For a concurrent structure, the time response of the process is limited by the branch with the largest time response.

$$T\,(W) = \mathrm{Max}\{T\,(s_i)\}, \quad i \in (1, 2, \ldots, m). \tag{7}$$

Cost (C): the cost $C(W)$ of service process **W** is composed of the sum of the costs of the services that constitute the service process.

$$C\,(W) = \sum_{i=1}^{m} C\,(s_i). \tag{8}$$

Availability (A): the availability $A(W)$ of service process **W** is composed of the availability product of each service that constitutes the service process.

$$A\,(W) = \prod_{i=1}^{m} A\,(s_i). \tag{9}$$

Loop structure (Loop): the service process W of loop structure is composed of m times repetition of loop body service s. For a given service s and the number of cycles m, it can also be equivalent to a sequential structure which is composed of m of the same service s. If a loop trip costs c_1, the estimated total cost of the cyclic structure is $m * c_1$. Compared with the method of expanding the loop, this method can calculate the QoS of the whole process more quickly and accurately.

Time (T): the response time $T(W)$ of the service process **W** is m times the response time of the loop body service s.

$$T\,(W) = m * T\,(s). \tag{10}$$

Cost (C): the cost $C(W)$ of service process W is m times the cost of loop service s.

$$C\,(W) = m * C\,(s). \tag{11}$$

Availability (A): the availability $A(W)$ of service process W is the *m-th* power of s availability for the loop body.

$$A\,(W) = A\,(s)^m. \tag{12}$$

4. Data Preprocessing for Atomic Service

Before service selection, the QoS planning module must know the QoS information of the atomic services in the service portfolio, which should conform to the atomic services trust modeling approach described in Section 3.1. For an atomic service, its QoS capabilities may be manifested at different levels over time. Simply put, multiple executions of an atomic service will necessarily reflect different QoS, just as a Web service will not have the same response time every time. In the face of such a large amount of historical information, it is very important for the accuracy of QoS planning to make a reasonable analysis and estimate the QoS capability of the atomic service in the next execution.

For each atomic service with the same functionality, its ability to provide services is different, which is not only restricted by its characteristics but also shows different QoS characteristics due to the environment it is deployed in and the resources it allocates. Therefore, for each atomic service, it is necessary to estimate the QoS capability of the services it currently provides. This process is called "QoS prediction" of atomic services. The main purpose of QoS prediction of atomic services is to meet the demand after QoS planning by selecting appropriate atomic services as far as possible, that is, to meet the user's QoS demand at the same time, to avoid the adjustment of the resource layer. A forecast is an estimate of the current QoS capabilities of atomic services that, if accurate, may avoid the additional overhead of resource layer control. If there is no special explanation, this paper takes the QoS prediction of the time dimension as an example for illustration.

4.1. Exponential Smoothing Prediction. The exponential smoothing method requires less data, so it is a simple and practical forecasting method for QoS prediction [25]. The method of exponential smoothing takes the weighted average of a sequence of historical events as a forecast of the future. It is a special case of the weighted moving average method, where we select only one weight, the weight of the nearest observed value. The weights of other data can be calculated automatically and will get smaller over time. The basic model of the exponential smoothing method is as follows.

$$F_{t+1} = \alpha Y_t + (1-\alpha)F_t. \tag{13}$$

In (13), F_{t+1} is the predicted value of time series in the period $t+1$, Y_t is the actual value of time series in the period t, F_t is the predicted value of time series in the period t, and α is the smoothing constant $(0 \le \alpha \le 1)$.

Formula (13) shows that the predicted value for the period $t+1$ is the weighted average of the actual value for the period t and the predicted value for the period t. In fact, we can show that the predicted value for the exponential smoothing method for any period is also the weighted average of all the historical actual data for the time series, as in $F_4 = \alpha Y_3 + \alpha(1-\alpha)Y_2 + (1-\alpha)^2 Y_1$.

Although the exponential smoothing method provides a weighted average of all historical observations, we do not need to store all historical data on the computer for the next period. Once the smoothing constant α is selected, we need only two items of information to calculate the predicted value. The formula shows that if α is given, we only need to know the actual value and the predicted value of the time series in the period of t, namely, Y_t and F_t; then, we can calculate the predicted value in the period of $t+1$.

Forecast accuracy. Although any value of α up to 0 to 1 is acceptable, some values of α produce more accurate predictions than others. We rewrite (13) in the following form:

$$F_{t+1} = F_t + \alpha\left(Y_t - F_t\right). \tag{14}$$

Therefore, the new forecast (F_{t+1}) is equal to the historical forecast (F_t) plus an adjustment, which is equal to α times the most recent forecast error $(Y_t - F_t)$. In other words, by adjusting the predicted value of the t period and some prediction errors, we can get the predicted value of the $t+1$ period. If the time series contains a large number of random variations, we tend to use a smaller smoothing constant. The reason for this choice is that many forecast errors are due to random variation, and we do not want to overreact to forecasts and adjust them too quickly. For the time series with small random variation, a larger smoothing exponential constant can be selected. The advantage of this method is that the condition can be changed quickly to adjust the error when the prediction error occurs. For the selection of the most appropriate α, the Mean Square Error (MSE) analysis should be carried out through the analysis of historical data or experiments.

4.2. QoS Prediction Experiment for Random Changes. Figure 2 shows the QoS capability estimation diagram of atomic service based on the exponential smoothing method, where the red broken line on the vertical axis represents the response time of a Web service executed for 50 consecutive times, and the blue asterisk represents the response time of these 50 atomic services. The abscissa represents the number of tests based on the time estimate from the exponential smoothing method. Besides, we use $T(Y_i)$ to represent the actual measured value of the i-th Web service response time, and $T(F_i)$ is used to represent the exponential smoothing estimate of the i-th Web service response time, where $i \in (1, 2, \ldots, 50)$.

As can be seen from the figure, the exponential smoothing method has a good prediction effect on the estimation of QoS capability of atomic service and smooths the capability deviation displayed by the service during its execution, which can eliminate interference errors to a certain extent. These errors can be caused by inaccuracies in measurement or by the environment in which the atomic service itself is deployed. The exponential smoothing method can be used to describe to a considerable extent the true level of service quality of the atomic service itself, which is an essential characteristic of the atomic service, as opposed to the performance characteristics that are affected by the environment in which it is deployed and the allocation of resources. Besides, the QoS estimation of exponential smoothing cannot guarantee the accurate prediction of QoS level for the next execution of the atomic service. In essence, it is the weighted average of the historical value of the service on the QoS indicator, which is an improvement of the time average of the trusted QoS modeling method of the atomic service in Section 3.1.

What we need to notice is that the value of each blue asterisk $T(F_i)$ is given by the weighted average of $T(Y_j)$ $[j \in (1, 2, \ldots, i-1)]$, which is less than i in the time coordinate. Of course, in practical application, to reduce the storage capacity, the actual value stored in the data center is the actual measured value and the estimated value at the last execution of the Web service which is calculated by (14), rather than recording all measurements of the historical execution time of the atomic service.

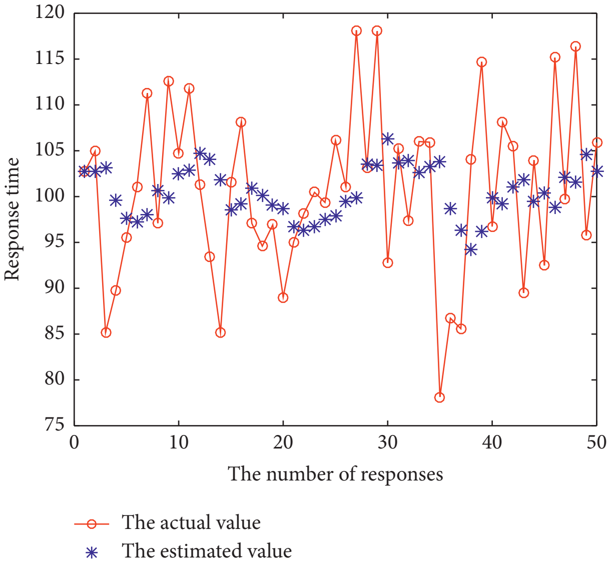


Figure 2: QoS estimation for random variation.

Linear fitting was conducted between the actual and estimated values of 50 measurements in this experiment, as shown in Figure 3, where the abscissa represents the actual value, the ordinate represents the estimated value, and $(T(Y_j), T(F_i))$ constitutes the coordinate points in the figure. Figure 3(a) represents the original data, and Figure 3(b) represents the results of actual and estimated values sorted from small to large (to reflect the fluctuation range of QoS value). The results show that both the actual value and the estimated value can reflect the response time of the Web service around 100, while the estimated value has a smaller standard variance. The fitting equation in Figure 3(b) is $T(F) = 0.29T(Y) + 71$, and the correlation coefficient $R^2 = 0.97$. R^2 is a real number between 0 and 1, which is used to evaluate the fitting degree of the regression equation. The greater the value, the better the fitting degree.

Figure 4 shows the residual analysis between the 50 measurement results and the estimated results. All the values are evenly distributed between them, belonging to the standard normal shape, which further confirms the correctness of linear fitting in Figure 3. In this experiment, the prediction effect of QoS is not good enough because QoS varies randomly. Nevertheless, the evaluation of atomic service QoS capability by using the exponential smoothing method is relatively accurate. It is essentially a dynamic estimation of the average value of the atomic service QoS capability and a weighted average of the historical measurement value. It explores the essential characteristics of the service. The QoS ability of atomic service will show random fluctuations with the white noise interference. The results of experiments show that the standard deviation of the estimated value 2.75 is significantly smaller than that of the actual value 9.19 (shown in Figures 2 and 3). The estimated value is embedded into small volatilities around real the QoS ability value 100. This means that the estimated value based on exponential smoothing method could reflect the nature of the QoS ability of atomic service. Therefore, this approach is consistent with the QoS modeling for atomic services in Section 3.1.

4.3. Prediction of QoS with a Linear Trend.

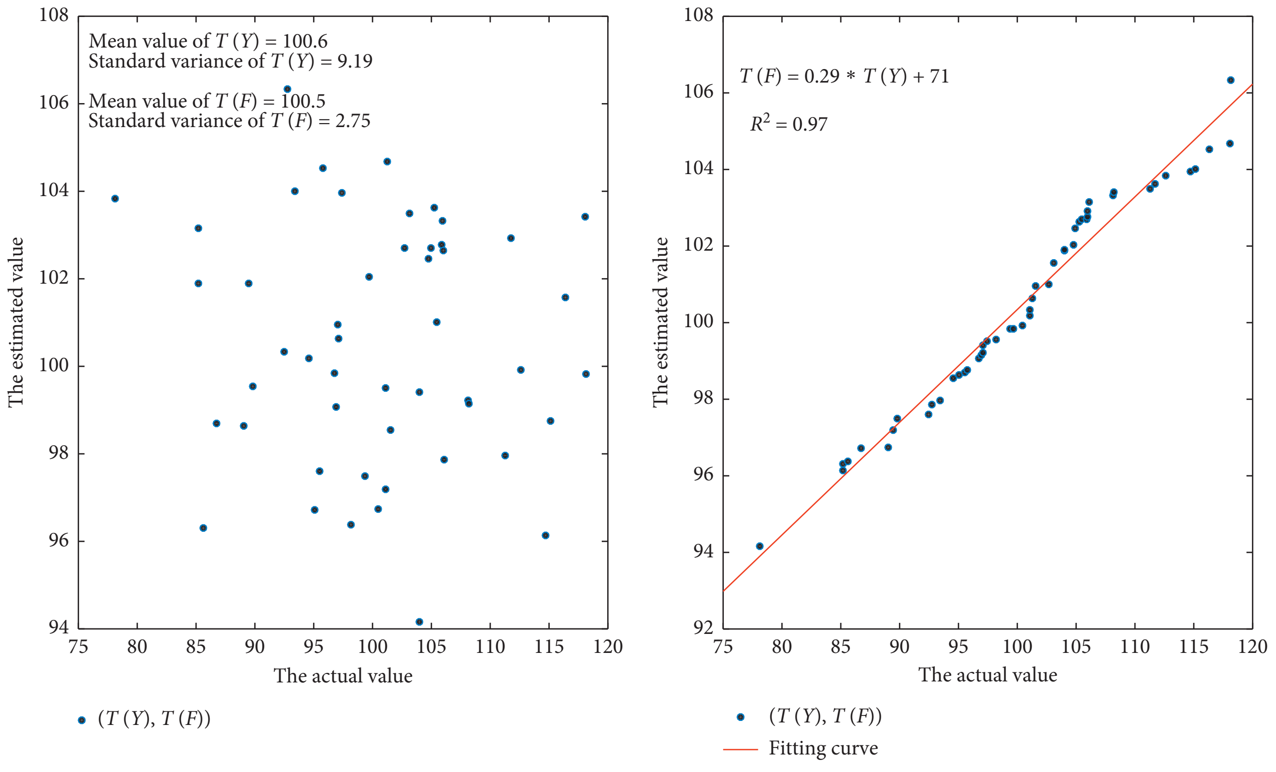
Studies have shown that the QoS supply level of a service entity is related to the load of the service node where the service entity is located. For example, in [26], through a large number of experimental tests on database services, it is found that the average service response time of the service is correlated with the CPU load of the service nodes. This relationship can be intuitively understood as follows: in general, a lower load on the service node means that the service node has fewer service invocation requests in the local service queue, and the service request of the service entity will be processed faster. Furthermore, when a load of service nodes is low, the task scheduling overhead caused by service process switching and file association and the interaction between processes of different service entities will be reduced, which will also improve the response time and availability of service entities.

The time response of an atomic service may change with the node load, and this change has a linear trend. In other words, in a certain time series, the response time of an atomic service will increase with the increase of node load and decrease with the decrease of node load. Although there is not necessarily a strictly linear relationship between them, such a monotonous trend does exist.

Based on the theory of the relationship between response time and node load, we conducted relevant tests again, as shown in Figure 5. For QoS estimation with a rising trend, the exponential smoothing method can also have a good tracking effect, and the estimation of the actual measured value is accurate, which fully reflects the time response characteristics of the atomic service. However, for a real system, the response time that increases as the node load increases should be a value that varies more slowly and is closer to that shown in Section 4.2.

Figure 6 shows the fitting between the actual and estimated values of QoS with an upward trend. Figure 6(a) shows the raw data, and Figure 6(b) shows the actual and estimated values sorted from smallest to largest. The results show that the QoS estimation with the linear trend is more accurate than that with the random change, and can reflect the changing trend of QoS. The R^2 of the linear equation fitted in Figures 6(a) and 6(b) is greater than 0.95, and the correlation coefficient is greater than 0.9. The RMSE value and SSE value of fitting results in Figure 6(a) are 14.76 and 10460, respectively, while the RMSE value and SSE value of fitting results in Figure 6(b) are 7.875 and 2977, respectively. This indicates that there is a strong correlation between the actual value and the estimated value and that the ratio between $T(F)$ and $T(Y)$ is close to 1, indicating that the estimated value is very accurate in predicting the actual measured value.

The above two experiments fully prove that the prediction of atomic service QoS information by the exponential smoothing method is accurate and feasible. The estimated data obtained by the exponential smoothing



(a)

(b)

Figure 3: QoS fitting of random variation. (a) The actual value. (b) The actual value.

method is a group of QoS information with relatively slow changes, which can fully reflect the reliable QoS level of the atomic service. Moreover, the algorithm is simple and easy to operate, and the consumption of system resources is small. Therefore, it can be fully applied to the preprocessing of QoS data of atomic service for reliable QoS planning.

5. Service Selection Based on QoS Planning

By preprocessing the history data of atomic service, the multidimensional trusted QoS information of each atomic service could be obtained. The typical information would be applied to trusted QoS planning, which is equivalent to decomposing the process QoS requirements of the application into QoS requirements for each service in this composite service.

QoS planning is the decomposition of the QoS requirements of the application process proposed by the user into the QoS requirements that should be satisfied by each service that makes up the process (the process is shown in Figure 7). Based on the results of QoS planning, the system selects the atomic services for each service that can guarantee the QoS requirements of that service according to a certain service selection strategy. The selection of appropriate atomic services to form a composite service is the key to meeting the QoS requirements of users. For verification, this paper uses three dimensions of the QoS metrics for illustration: time (T), cost (C), and availability (A). Based on the results of QoS planning, three service selection strategies are proposed by the author.

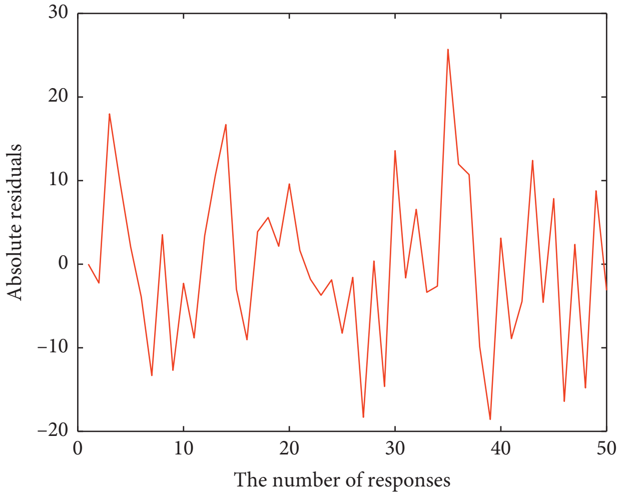


Figure 4: Residual analysis diagram.

5.1. *Time Preference Algorithm.*

For each service in the composite service, the atomic service with the shortest response time (the value after data preprocessing) is selected to complete the service according to the requirements of the time dimension in the results of QoS planning (average time complexity is $O(n^2)$). This method maximally satisfies the user's requirements for the QoS time dimension, because for each atomic service, even if the QoS capability is estimated using data preprocessing, there is still no precise guarantee that the QoS characteristics of the atomic service will be the

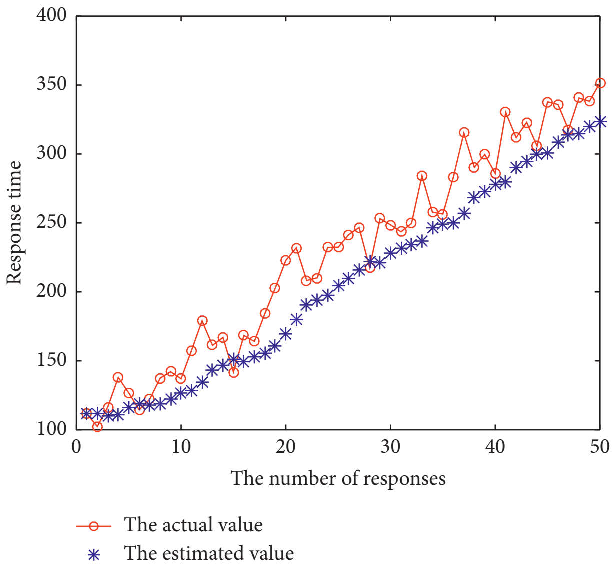


Figure 5: Estimates of QoS with an upward trend.

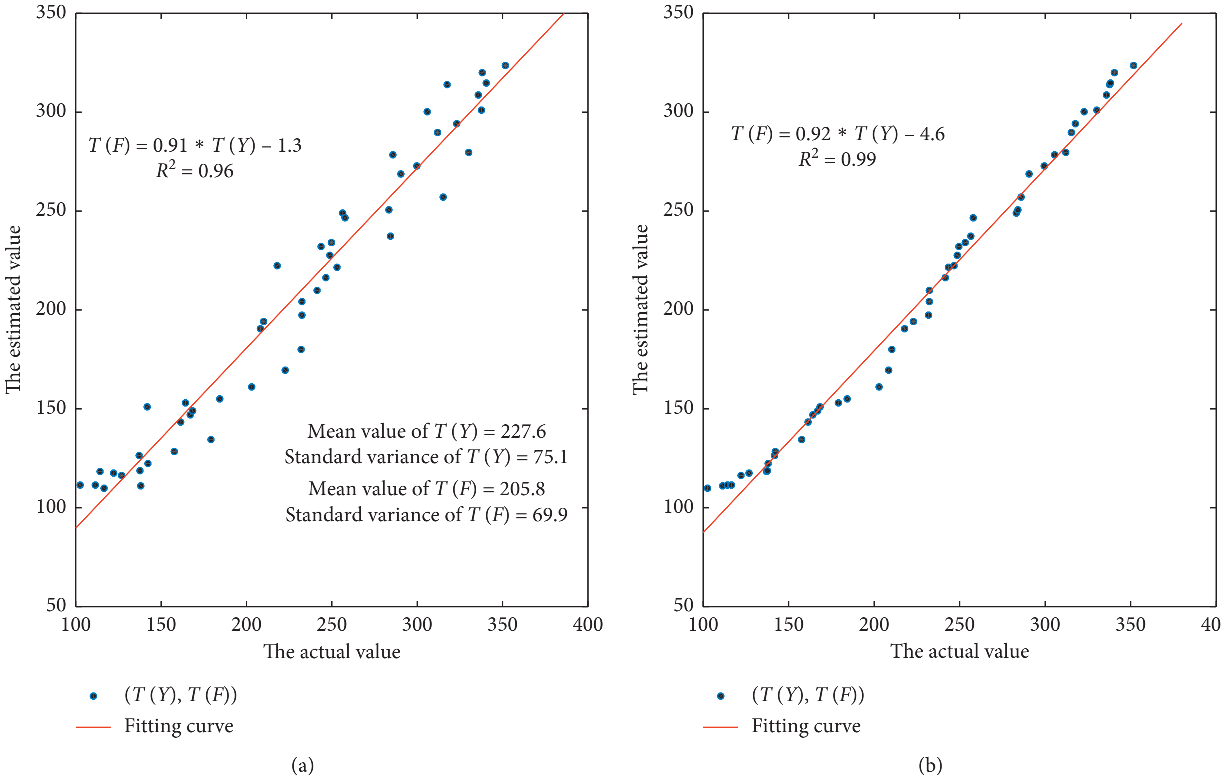


Figure 6: QoS fitting with an upward trend. (a) The actual value. (b) The actual value.

estimated value at the next execution. In the case when the time characteristics of the atomic service are significantly different from the estimated value, the user's requirements for the time dimension could be satisfied to the maximum extent with the time preference policy, because it leaves the maximum amount of redundancy in time.

5.2. Cost/Availability (C/A) Preference Algorithm. The cost/availability (C/A) preference algorithm is a service selection algorithm based on QoS planning. The basic idea, shown as follows, is to satisfy the user's demand in the time dimension at the lowest possible cost. (Algorithm 1).

Algorithm description:

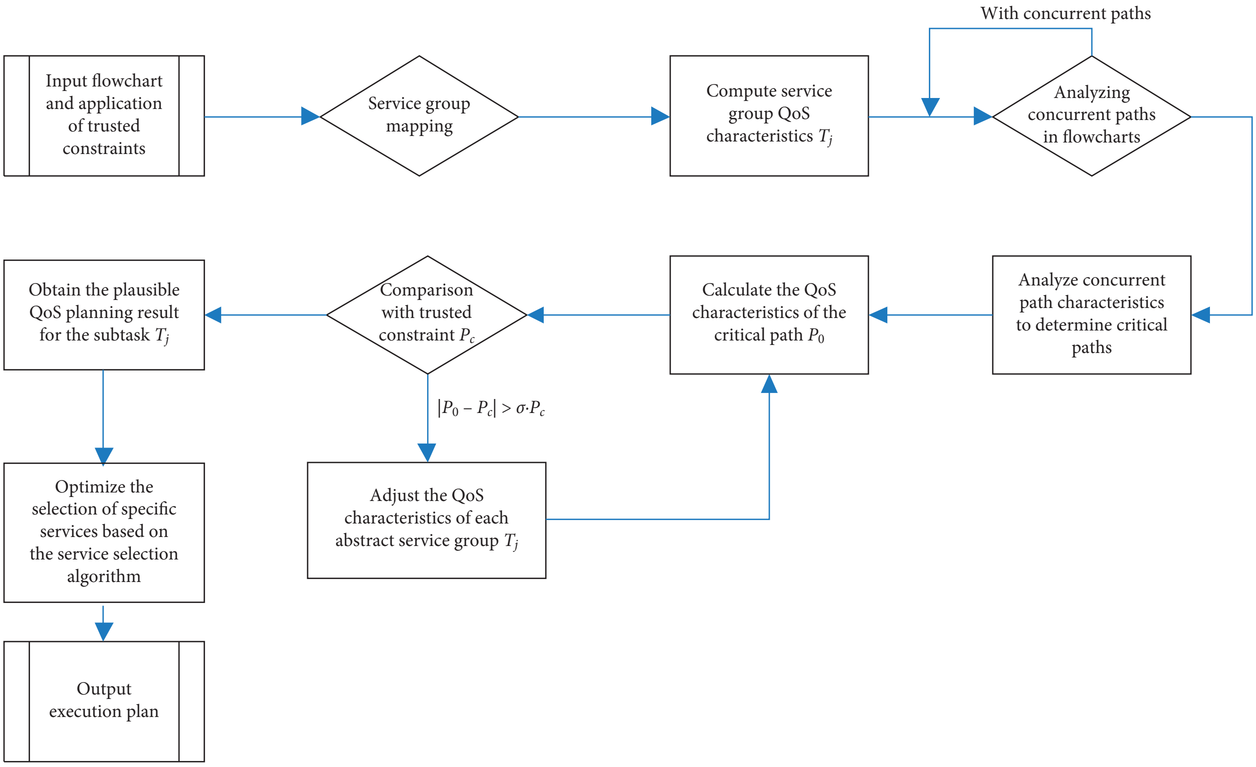

With concurrent paths
Input flowchart and application of trusted constraints
Service group mapping
Compute service group QoS characteristics T_j
Analyzing concurrent paths in flowcharts
Obtain the plausible QoS planning result for the subtask T_j
Comparison with trusted constraint P_c
Calculate the QoS characteristics of the critical path P_0
Analyze concurrent path characteristics to determine critical paths
$|P_0 - P_c| > \sigma \cdot P_c$
Optimize the selection of specific services based on the service selection algorithm
Adjust the QoS characteristics of each abstract service group T_j
Output execution plan

Figure 7: Flowchart of the QoS planning algorithm.

(1) Based on the results of QoS planning, for each service, the atomic services that satisfy its real-time requirements T_{pj} and form a sequence A_j are selected. Note that the atomic services in the sequence are out of order at this point.

(2) For each atomic service in the sequence A_j, it is backed up to form a service group until the availability meets the QoS planning requirements A_{pj} (if the availability of the atomic service already meets the requirements, no backup is needed); the cost of the atomic service group is $C_{ji}{}^{*}k$, and k is the number of atomic services in the service group.

(3) Sort all the atomic service groups in increasing order of cost/availability (C/A), with the services of different functions in one column, to obtain the order A_j.

(4) Select the first service group from the sequence A_j, i.e., the service S_j^1 with the smallest C/A, to form the service flow.

This service selection strategy allows users to achieve the required QoS requirements for each dimension with a minimal cost. The average time complexity is $O(\sum s_{ji} + B \cdot N_A + N_A^2)$, where B is the average backup times of A_j and N_A is number of atomic services in A_j. However, for atomic services whose real-time QoS estimates deviate significantly from the actual measurements, the real-time performance may not be satisfied.

5.3. Euclidean Distance Preference Algorithm. The composite service selected by the Euclidean distance preference algorithm is the closest to the comprehensive quality of service proposed by the user [27]. Besides, the required quality of service for each service after QoS planning is denoted as $\mathrm{QoS}(s) = (T, C, A)$, which represents the conditions that the service should satisfy in three dimensions: real time, cost, and availability.

For example, there are n atomic services s_1, s_2, s_3, $\mathrm{QoS}(s_1) = (T_1, C_1, A_1)$, $\mathrm{QoS}(s_2) = (T_2, C_2, A_2)$, $\mathrm{QoS}(s_3) = (T_3, C_3, A_3)$, so that the services of the same function could be described as a matrix:

$$\mathrm{Q} = \begin{pmatrix} T_1 & C_1 & A_1 \\ T_2 & C_2 & A_2 \\ \cdots & \cdots & \cdots \\ T_n & C_n & A_n \end{pmatrix}. \tag{15}$$

In terms of service quality criteria, there are quality standards where higher values indicate better quality, while there are quality standards where lower values indicate better quality [28]. The former are called positive-quality criteria, such as availability, and the latter are called negative-quality criteria, such as execution time. In addition, in order to prevent the value of a quality criterion from being exceedingly large to influence the final result, it is necessary to concentrate on the values of all quality criteria between [0, 1].

Input:
(1) Flowchart
(2) The flowchart contains m tasks, denoted as t_j, $(j \in 1, 2, \ldots, m)$.
(3) Each task has n_j service with the same function to complete the task, denoted as s_{ji}, $(i \in 1, 2, \ldots, n_j)$.
(4) Each atomic service has three QoS properties: T (real time), A (availability), and C (cost). Each atomic service has its own real-time interval $[T_{ji1}, T_{ji2}]$ and estimated execution time $\overline{T_{ji}}$.
(5) Each service would have real-time requirements T_{pj} and availability requirements A_{pj} after QoS planning, $j \in 1, 2, \ldots, m$.

Algorithm 1

For negative service quality, we use (16) for processing; for positive service quality, (17) is used for processing.

$$V_{i,j} = \begin{cases} \dfrac{Q_j^{\max} - Q_{i,j}}{Q_j^{\max} - Q_j^{\min}}, & \text{if } Q_j^{\max} - Q_j^{\min} \neq 0, \\ 1, & \text{if } Q_j^{\max} - Q_j^{\min} = 0. \end{cases} \tag{16}$$

$$V_{i,j} = \begin{cases} \dfrac{Q_{i,j} - Q_j^{\max}}{Q_j^{\max} - Q_j^{\min}}, & \text{if } Q_j^{\max} - Q_j^{\min} \neq 0, \\ 1, & \text{if } Q_j^{\max} - Q_j^{\min} = 0. \end{cases} \tag{17}$$

In the above equation, atomic services s_{ij} with the same function constitute service s_j, $Q_j^{\max}$ denotes the maximum value of service s_j on dimension Q, $Q_j^{\min}$ denotes the minimum value of all atomic services s_j on dimension Q, and $Q_{i,j}$ denotes the value of atomic services on dimension Q, where Q denotes the dimension of service quality, e.g., real time T, cost C, and availability A.

The matrix in (15) is processed according to (16) and (17) to obtain

$$Q^* = \begin{pmatrix} V_{11} & V_{12} & V_{13} \\ V_{21} & V_{22} & V_{23} \\ \cdots & \cdots & \cdots \\ V_{n1} & V_{n2} & V_{n3} \end{pmatrix} = \begin{pmatrix} Q_1 \\ Q_2 \\ \cdots \\ Q_n \end{pmatrix}. \tag{18}$$

We calculate the demand $\text{QoS}(s) = (T, C, A)$ after QoS planning using the above formula also for the comprehensive quality to obtain the comprehensive quality value q. q and Q_1 are considered as two points, and then the Euclidean distance is used to calculate the deviation between q and Q_1. The value of each service quality of q after processing could be set as (q_1, q_2, q_3), and then the deviation between q and Q_j is

$$d(q, Q_j) = \sqrt{\sum_{i=1}^{3} (q_i - V_{ji})^2}. \tag{19}$$

Then, the smallest one in $d(q, Q_j)$ is the service that is closest to the comprehensive service quality proposed by the user. The average time complexity is $O(n \cdot (m + n + 1))$, where n is the number of atomic services and m is the number of QoS dimensions.

5.4. Simulation Experiment. Figure 8 shows the graph of the results after using trusted QoS planning. In the following, the three algorithms are compared in terms of the total time, total cost, and Euclidean distance metrics, respectively. The horizontal axis indicates the number of simulation experiments and the vertical axis indicates the indicator we examined.

The three algorithms are examined in terms of total time, which is shown in Figure 9.

From the figure, it could be seen that the time (T) preference algorithm has the shortest total time, which is due to the fact that the algorithm uses only the time metric as the basis for service selection, which maximizes the satisfaction of the time requirement. The random selection algorithm and the cost/availability (C/A) preference algorithm exhibit random fluctuations around the expected value (1400) because they do not consider the time metric. The Euclidean distance preference algorithm performs better than the random selection algorithm and the cost/availability (C/A) preference algorithm in terms of time metrics because it considers time, cost, and availability together, but not as well as the time (T) preference algorithm.

The three algorithms are examined in terms of total cost, which is as shown in Figure 10.

From Figure 10, it could be seen that the cost/availability (C/A) preference algorithm is the least costly. This is due to the fact that the algorithm uses cost (C) as a metric for measuring service selection and therefore is able to maximize the cost satisfaction.

The random selection algorithm and the time (T) preference algorithm exhibit random fluctuations around the expected value (250) because the metric of cost is random in both. The Euclidean distance preference algorithm performs better than the random selection algorithm and the time (T) preference algorithm in terms of cost metrics because it considers time, cost, and availability together, but not as well as the cost/availability (C/A) preference algorithm.

The three algorithms are examined in terms of similarity to the expected value (Euclidean distance), as shown in Figure 11.

As can be seen from Figure 11, the Euclidean distance preference algorithm has the best performance in this metric. The algorithm considers a combination of time (T), cost (C), and availability (A). The random selection algorithm has the worst performance in all three metrics because they are chosen randomly. The time (T) preference algorithm performs better than the random selection algorithm

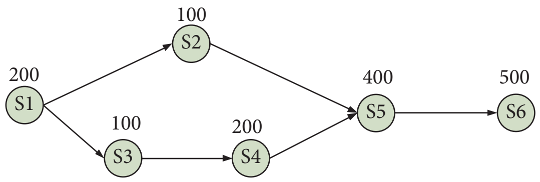


Figure 8: Planning result for process QoS.

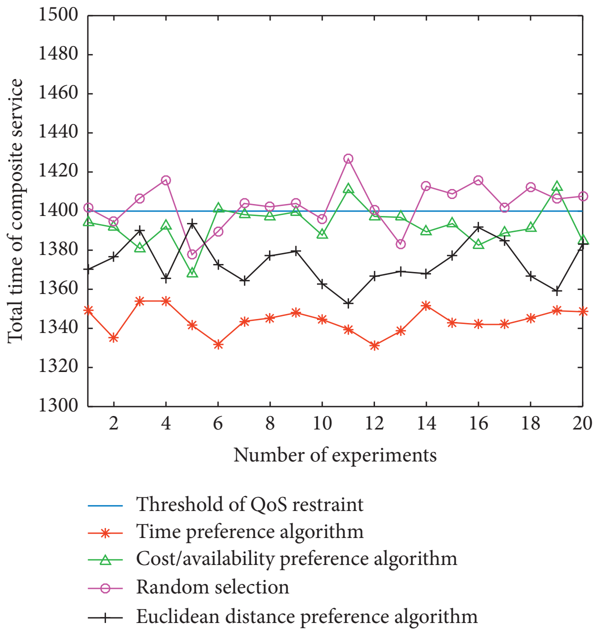


Figure 9: Comparison of the total time of the four algorithms.

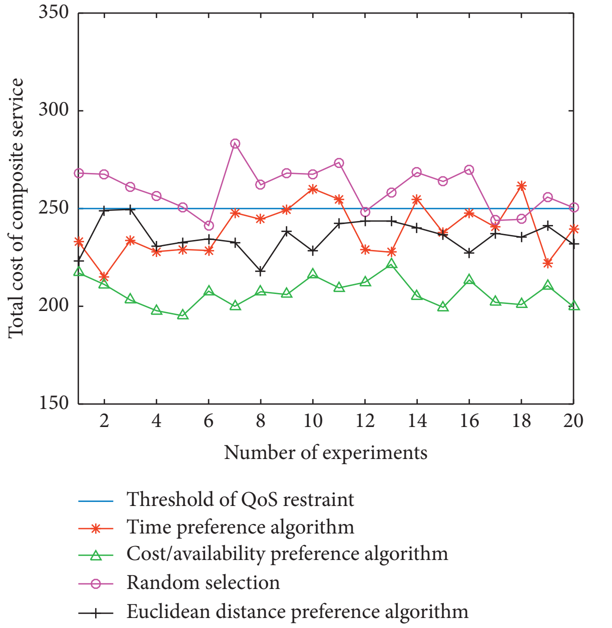


Figure 10: Comparison of the total cost of the four algorithms.

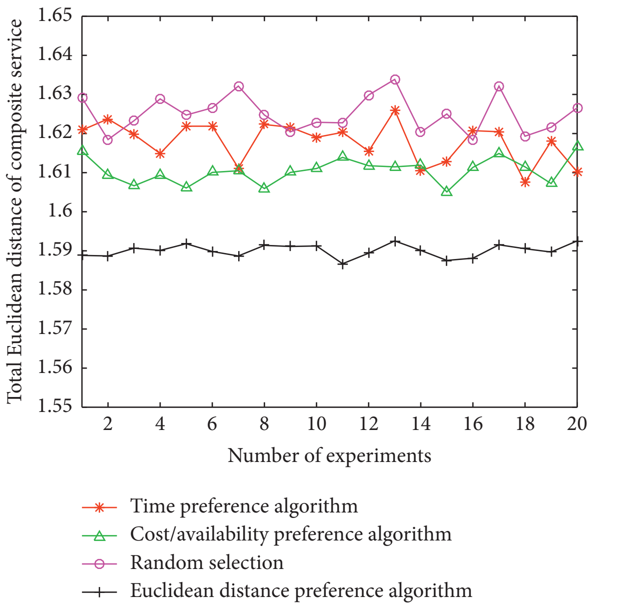


Figure 11: Comparison of the four algorithms with the expected value of the Euclidean distance.

in terms of the Euclidean distance of the expected value because it takes time (T) into account. The cost/availability (C/A) preference algorithm outperforms the time (T) preference algorithm in terms of Euclidean distance because it considers both cost (C) and availability (A), but it is inferior to the Euclidean distance preference algorithm.

In summary, the Euclidean distance preference algorithm integrates the three indicators into consideration and has a more balanced and excellent characteristic. In practice, different algorithms can be chosen according to different needs.

5.5. Service Selection Based on Heuristic Algorithms. When the computational power is sufficient, service selection can be achieved using heuristic algorithms without going through QoS planning. Service selection for composite service is an NP-hard problem, and this paper propose the memetic algorithm (MA) by combining genetic algorithm (GA) with local search to solve this class of problems [29, 30].

The algorithmic framework of this paper is listed in Algorithm 2. Firstly, various parameters of the genetic algorithm, the flow of the composite service, and the QoS of each atomic service are required to be input. Then the population P is generated by the initialization function InitialPopulation(), and next it enters a loop until the number of iterations reaches the set maximum number of iterations. In the loop, firstly, Selection() is used to select the parent population to participate in crossover and mutation in a roulette manner, then GeneticOperation() is used to perform crossover and mutation operation. LocalSearch() is a further search after the crossover and mutation operation so as to find the local optimum, and then UpdatePopulation() is used to update the population and get a better

```
**Input:** The maximum number of iterations: $N_{\mathrm{Imax}}$. The size of population: $S_{\mathrm{P}}$. The size of the mating pool: $S_{\mathrm{MP}}$. The size of tournament: $S_{\mathrm{T}}$. The probability of crossover: $P_{\mathrm{C}}$. The probability of mutation: $P_{\mathrm{M}}$. The process of composite service and the QoS of each atomic service.
**Output:** The selected atomic services and the QoS of composite service.
**Steps:**
Initialize $P$←InitialPopulation ($S_P$);
**for** $n = 1$; $n < \mathrm{N}_{\mathrm{Imax}} + 1$; $n$++
    $P_{\mathrm{parent}} \leftarrow$ Selection ($P$, SMP, ST);
    $P_{\mathrm{child}} \leftarrow$ GeneticOperation ($P_{\mathrm{parent}}$, $P_C$, $P_M$);
    $P_{\mathrm{newchild}} \leftarrow$ LocalSearch ($P_{\mathrm{child}}$);
    $P \leftarrow$ UpdatePopulation ($P$, $P_{\mathrm{newchild}}$);
end for
return $P$, QoSmax
```

Algorithm 2: MA of service selection.

```
**Input:** The chromosome $P_{\mathrm{child}}$. The process of composite service and the QoS of each atomic service.
**Output:** New chromosome $P_{\mathrm{newchild}}$.
Steps:
Initialize $P_{\mathrm{newchild}} = P_{\mathrm{child}}$;
**for** $i = 1$; $i < N_{\mathrm{AbstS}} + 1$; $i$++
    **for** $j = 1$; $j < N_{\mathrm{AtomS}}(i) + 1$; $j$++
        $P_{\mathrm{child}}\ (i) = S(j)$;//(A atomic service differs from $P_{\mathrm{Nchild}}$' gene)
        **if** $QoS(P_{\mathrm{child}}) > QoS(P_{\mathrm{newchild}})$//(find an atomic service that improve QoS)
            $P_{\mathrm{newchild}} = P_{\mathrm{child}}$;
        **end if**
    **end for**
    $P_{\mathrm{child}}\ (i) = P_{\mathrm{newchild}}(i)$;
**end for**
**return** $P_{\mathrm{newchild}}$
```

Algorithm 3: Algorithm of local search.

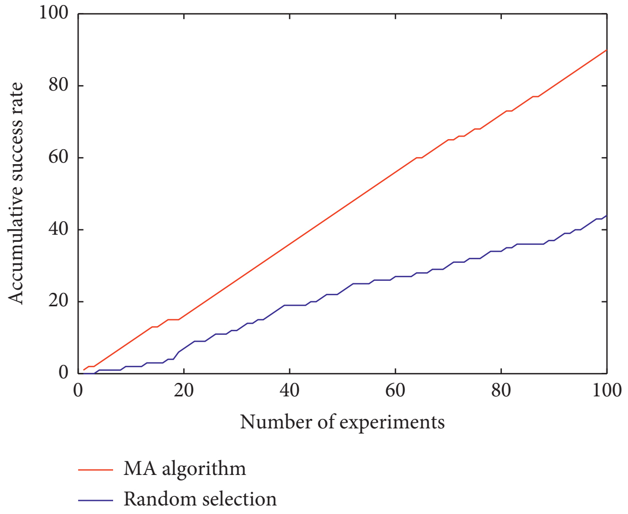


Figure 12: Algorithm comparison based on success rate of QoS assurance.

chromosome population. Finally, the result of the calculation is output.

The local-search strategy is given by Algorithm 3 in order to accelerate the convergence. N_{AbstS} in Algorithm 3 denotes the number of abstract services in the composite service, and $N_{\mathrm{AtomS}}(i)$ denotes the number of atomic services in abstract service i. We traverse each gene of the chromosome and determine whether replacing it with another atomic service improves the QoS. If changing the atomic service of a gene can improve the QoS, the new gene is accepted so that a local optimum is achieved. If QoS could be improved by changing atomic service of the typical gene, this gene could be accepted in order to reach optimal results. The average time complexity of MA algorithm is $O(N_{\text{Imax}}\sum_{N_{\text{AbstS}}}N_{\text{AtomS}}(i))$.

The simulation experiments are performed on the composite service flow in Figure 8, using random selection of atomic services and MA selection of atomic services for each experiment, for a total of 100 tests. We set $N_{\text{Imax}} = 1000$, $S_{\text{P}} = 100$, $S_{\text{MP}} = 100$, $S_{\text{T}} = 2$, $P_{\text{C}} = 0.1$, and $P_{\text{M}} = 0.9$. Figure 12 shows the number of successes of the above two methods in meeting the QoS requirements proposed by the application in 100 experiments, respectively, where the horizontal coordinate represents the number of experiments n, $n \in 1, 2, \ldots, 100$, and the vertical coordinate represents the number of times m the QoS requirements of the composite service are satisfied in n experiments, $m \in 1, 2, \ldots, n$. As can be seen from the figure, the number of successes after MA selection of atomic services is significantly greater than the number of successes for random selection of atomic services. Among the 100 experiments conducted, the user's requirements can be satisfied up to 90% because of the QoS of the composite service after MA-selected atomic service, while the randomly selected service has less than 50% chance of success in terms of the QoS requirements proposed by the user, although the user's functional requirements could be satisfied.

6. Conclusion and Future Work

Web composite service technology aims at solving the problem of effective integration of functionally diverse Web service resources on the Internet, and users' multifaceted application requirements could be satisfied by constructing functionally complex and superior composite services. The huge number of candidate services with similar functional attributes and different nonfunctional attributes will increase the complexity of composite service and lead to the problem of Web composite service as NP-hard problem. In the complex network environment, it is difficult for the traditional QoS-based composite service approach to guarantee that the constructed combination solution can meet the user requirements because it cannot measure the trustworthiness of Web services. In this paper, we mainly focus on how to solve the problem of trustworthy QoS guarantee for composite service in the highly complex, dynamic, and untrustworthy Internet environment. We design an operational mechanism to guarantee the trustworthiness requirements of the composite services. The various task segments in the composite services are completed by virtual services, and the quality of the composite services is ensured by QoS planning of the upper layer applications, prediction of the QoS capability of atomic services, and implementation of various service selection algorithms to finally meet the abstract service trustworthiness requirements. Future work will optimize (1) the selection of services with identical functions but differing interfaces and (2) the impact of interatomic service correlations on composite services.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (71501153), the Innovation Capability Support Project of Shaanxi Province of China (2021KRM135), the Research Fund of Grand Theory and Practical Problem in Philosophy and Social Science of Shaanxi Province of China (2021ND0221), the Research Fund of the Education Department of Shaanxi Province of China (20JG020), and the Natural Science Foundation of Shaanxi Province of China (2019JM-572).

References

[1] A. Muñoz and E. B. Fernandez, "TPM: a pattern for an architecture for trusted computing," in *Proceedings of the European Conference on Pattern Languages of Programs*, pp. 1–8, Kloster Irsee, Bavaria, Germany, July 2020.

[2] A. Nageswaran, A. Revathi, and R. Kaladevi, "Adaptive video streaming with multidimensional quality of service," *European Journal of Molecular & Clinical Medicine*, vol. 7, no. 8, pp. 2098–2105, 2020.

[3] C. Wang, H. Ma, G. Chen, S. Hartmann, and J. Branke, "Robustness estimation and optimisation for semantic web service composition with stochastic service failures," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 99, pp. 1–16, 2020.

[4] F. Ibrahim and E. E. Hemayed, "Trusted cloud computing architectures for infrastructure as a service: survey and systematic literature review," *Computers & Security*, vol. 82, pp. 196–226, 2019.

[5] E. Gelenbe, J. Domanska, P. Fröhlich, M. P. Nowak, and S. Nowak, "Self-aware networks that optimize security, QoS, and energy," *Proceedings of the IEEE*, vol. 108, no. 7, pp. 1150–1167, 2020.

[6] L. Chuang, W. Yuanzhuo, and T. Liqin, "Development of trusted network and challenges it faces," *ZTE Communications*, vol. 6, no. 1, pp. 13–17, 2020.

[7] H. Gao, D. Chu, Y. Duan, and Y. Yin, "Probabilistic model checking-based service selection method for business process modeling," *International Journal of Software Engineering and Knowledge Engineering*, vol. 27, no. 6, pp. 897–923, 2017.

[8] L. Wang, "Architecture-based reliability-sensitive criticality measure for fault-tolerance cloud applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 11, pp. 2408–2421, 2019.

[9] M. Al-Khafajiy, T. Baker, M. Asim et al., "COMITMENT: a fog computing trust management approach," *Journal of Parallel and Distributed Computing*, vol. 137, pp. 1–16, 2020.

[10] L. Zeng, B. Benatallah, A. Ngu et al., "QoS-aware middleware for web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.

[11] H. Issa, C. Assi, M. Debbabi, and S. Ray, "QoS-aware middleware for web services composition: a qualitative approach," *Enterprise Information Systems*, vol. 3, no. 4, pp. 449–470, 2009.

[12] R. Alhajj and J. Rokne, *Encyclopedia of Social Network Analysis and Mining*, Springer Publishing Company, USA, 2014.

[13] M. Moghaddam and J. G. Davis, *Service Selection in Web Service Composition: A Comparative Review of Existing Approaches*, Web Services Foundations, New York, NY, USA, 2014.

[14] W. Zhang, Y. Yang, S. Zhang, D. Yu, and Y. Li, "Correlation-aware manufacturing service composition model using an extended flower pollination algorithm," *International Journal of Production Research*, vol. 56, no. 14, pp. 4676–4691, 2018.

[15] D. Artz and Y. Gil, "A survey of trust in computer science and the semantic web," *Journal of Web Semantics*, vol. 5, no. 2, pp. 58–71, 2007.

[16] S. Wang, Q. Sun, H. Zou, and F. Yang, "Reputation measure approach of web service for service selection," *IET Software*, vol. 5, no. 5, pp. 466–473, 2011.

[17] D. Ardagna and B. Pernici, "Global and local QoS guarantee in web service selection," in *Proceedings of the international Conference on Business Process Management*, pp. 32–46, Nancy, France, May 2005.

[18] S. X. Sun and J. Zhao, "A decomposition-based approach for service composition with global QoS guarantees," *Information Sciences*, vol. 199, pp. 138–153, 2012.

[19] C. Surianarayanan, G. Ganapathy, and M. S. Ramasamy, "An approach for selecting best available services through a new method of decomposing QoS constraints," *Service Oriented Computing and Applications*, vol. 9, no. 2, pp. 107–138, 2015.

[20] V. Gabrel, M. Manouvrier, and C. Murat, "Optimal and automatic transactional web service composition with dependency graph and 0-1 linear programming," in *Proceedings of the International Conference on Service-Oriented Computing*, pp. 108–122, Paris, France, November 2014.

[21] A. J. S. Cardoso, "Quality of service and semantic composition of workflows," Doctoral Dissertation, University of Georgia, Athens, GA, USA, 2002.

[22] H. Elshaafi and D. Botvich, "Optimisation-based collaborative determination of component trustworthiness in service compositions," *Security and Communication Networks*, vol. 9, no. 6, pp. 513–527, 2016.

[23] B. Benatallah, M. Dumas, Q. Shen, and A. H. Ngu, "Declarative composition and peer-to-peer provisioning of dynamic web services," in *Proceedings of the 18th International Conference on Data Engineering*, pp. 297–308, San Jose, CA, USA, February 2002.

[24] M. P. Papazoglou and D. Georgakopoulos, "Introduction," *Communications of the ACM*, vol. 46, no. 10, pp. 24–28, 2003.

[25] S. Mahajan, L.-J. Chen, and T.-C. Tsai, "Short-term PM2.5 forecasting using exponential smoothing method: a comparative analysis," *Sensors*, vol. 18, no. 10, p. 3223, 2018.

[26] S. Ranjan and E. Knightly, "High-performance resource allocation and request redirection algorithms for web clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1186–1200, 2008.

[27] S. P. Patel and S. H. Upadhyay, "Euclidean distance based feature ranking and subset selection for bearing fault diagnosis," *Expert Systems with Applications*, vol. 154, Article ID 113400, 2020.

[28] Y. Liu, A. H. Ngu, and L. Z. Zeng, "QoS computation and policing in dynamic web service selection," in *Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers & Posters*, pp. 66–73, New York, NY, USA, May 2004.

[29] X. Fu, P. Pace, G. Aloi, L. Yang, and G. Fortino, "Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm," *Computer Networks*, vol. 177, Article ID 107327, 2020.

[30] B. Cao, W. Zhang, X. Wang, J. Zhao, Y. Gu, and Y. Zhang, "A memetic algorithm based on two_Arch2 for multi-depot heterogeneous-vehicle capacitated Arc routing problem," *Swarm and Evolutionary Computation*, vol. 63, Article ID 100864, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5518528, 12 pages
https://doi.org/10.1155/2021/5518528

Research Article

A Malicious URL Detection Model Based on Convolutional Neural Network

Zhiqiang Wang,[1,2,3] Xiaorui Ren,[1] Shuhao Li,[1] Bingyan Wang,[1] Jianyi Zhang ,[1] and Tao Yang[3]

[1] *Beijing Electronic Science and Technology Institute, Beijing 100071, China*
[2] *State Information Center, Beijing 100032, China*
[3] *Key Lab of Information Network Security, Ministry of Public Security, Shanghai 200000, China*

Correspondence should be addressed to Jianyi Zhang; nese@163.com

Received 18 January 2021; Accepted 30 April 2021; Published 15 May 2021

Academic Editor: Hao Peng

With the development of Internet technology, network security is under diverse threats. In particular, attackers can spread malicious uniform resource locators (URL) to carry out attacks such as phishing and spam. The research on malicious URL detection is significant for defending against these attacks. However, there are still some problems in the current research. For instance, malicious features cannot be extracted efficiently. Some existing detection methods are easy to evade by attackers. We design a malicious URL detection model based on a dynamic convolutional neural network (DCNN) to solve these problems. A new folding layer is added to the original multilayer convolution network. It replaces the pooling layer with the k-max-pooling layer. In the dynamic convolution algorithm, the width of feature mapping in the middle layer depends on the vector input dimension. Moreover, the pooling layer parameters are dynamically adjusted according to the length of the URL input and the depth of the current convolution layer, which is beneficial to extracting more in-depth features in a wider range. In this paper, we propose a new embedding method in which word embedding based on character embedding is leveraged to learn the vector representation of a URL. Meanwhile, we conduct two groups of comparative experiments. First, we conduct three contrast experiments, which adopt the same network structure and different embedding methods. The results prove that word embedding based on character embedding can achieve higher accuracy. We then conduct the other three experiences, which use the same embedding method proposed in this paper and use different network structures to determine which network is most suitable for our model. We verify that the model designed in this paper has the highest accuracy (98%) in detecting malicious URL through these experiences.

1. Introduction

Hackers often use spam and phishing [1, 2] to trick users into clicking malicious URL, the Trojans will be implanted into the victims' computers, or the victims' sensitive information will be leaked. The technology of malicious URL detection can help users identify malicious URL and prevent users from being attacked by malicious URL. Traditionally, research on malicious URL detection adopts blacklist-based methods to detect malicious URL. This method has some unique advantages. It has high speed, has low false-positive rate, and is easy to realize. However, nowadays, the domain generation algorithm (DGA) can generate tens of thousands of different malicious domain names every day, which cannot be detected effectively by the traditional blacklist-based methods.

Researchers have been using a machine learning technique to identify malicious URL. However, these methods often need to extract the features manually, and attackers can design these features to avoid being identified. Faced with today's complex network environment, designing a more effective malicious URL detection model becomes a research focus.

This paper proposes a malicious URL detection model based on a DCNN. It adopts word embedding based on the character embedding method to extract features automatically and learn the URL's expression. Meanwhile, we verify the validity of the model through a series of contrast experiments.

In this study, our innovations and contributions are as follows:

(1) This paper proposes a malicious URL detection model based on a DCNN. The dynamic convolution algorithm adds a new folding layer to the original multilayer convolution structure. It replaces the pooling layer with the k-max-pooling layer. In the dynamic convolution algorithm, the width of feature mapping in the middle layer depends on the vector input dimension. Moreover, the pooling layer parameters are dynamically adjusted according to the length of the URL input and the depth of the current convolution layer, which helps extract more in-depth features in a wider range.

(2) In the stage of feature extraction and representation, the features are extracted from the URL sequence. The extracted features are integrated into a vector, and the vector is processed directly by the convolutional neural network to learn the classification model. This method not only simplifies the process of feature extraction, it does not depend on extracting features manually, but also combines the advantages of character embedding and word embedding. Word embedding can obtain word sequence information, which cannot be obtained by character embedding. Character embedding can process special characters and unfamiliar words in the URL. The dictionary and vector dimension are also not too big. The combination can save memory space and express the URL more effectively, which will help extract information from the URL.

(3) To prove the feasibility of the model proposed in this paper, we did a lot of comparative experiments. As for the embedding method, we conduct three contrast experiments to verify that word embedding based on character embedding achieves higher accuracy than word embedding and character embedding. We also perform three contrast experiments and prove that leveraging the network structure consisting of a DCNN and different fields extracted from the URL can achieve a better effect.

The rest of this paper is organized as follows. In Section 2, we introduce the research status of malicious URL detection methods. In Section 3, we present the malicious URL detection model and its main modules. In Section 4, we conduct experiments on the malicious URL detection model and test the embedding methods. Finally, we offer a brief discussion in Section 5.

1.1. Related Work. At present, the methods [3–5] of detecting malicious URL can be roughly divided into traditional detection methods based on blacklist and detection methods based on machine learning. Literatures [6, 7] introduce the detection method based on a blacklist. Although this method is simple and efficient, it cannot detect the newly generated malicious URL, which has severe limitations. Literature [8] points out that attackers can generate various malicious domain names through a random seed to effectively evade the traditional detection method based on a blacklist.

In literatures [9–11], researchers have applied machine learning technology to detect malicious URL. Machine learning learns the prediction model based on statistical properties and classifies a URL as a malicious URL or a benign URL. This method attempts to analyze URL and their relevant websites or web page information to extract the features. The features extracted by this method are often divided into two types, static features and dynamic features. Literature [12] obtains lexical information in URL strings, information about hosts, and sometimes HTML and JavaScript content. Literature [13] extracts a series of network traffic-related features from URL, and the support vector machine (SVM) is adopted for detection. Literature [14] proposes three methods of feature processing to optimize the classification effect. While the above methods have shown good performance, there are still some limitations. Traditional detection methods based on machine learning often require extract features manually [15, 16]. Attackers can avoid being detected by these detection methods by designing these features, making it very difficult to maintain the detection system based on traditional machine learning. Additionally, in large-scale malicious URL detection, a trained model may lose some useful information from URL.

Referring to the idea of text classification [17, 18], many researchers have proposed a variety of methods based on deep learning [19] models to detect malicious URL and judge whether a URL is malicious only according to the strings contained in the URL. These methods can automatically extract valid information in the URL. For example, literature [20] uses the cyclic neural network model at the character level to classify URL generated by DGA. Literature [21] proposes the method of extreme machine learning to detect malicious URL. Combining n-gram model with deep learning, literature [22] takes the advantages of character-level semantic features to detect whether DGA generates the URL. A variety of deep learning architectures for malicious URL detection are listed in the literature [23], including the structure of single-layer long short term memory(LSTM) [24], the structure of bidirectional LSTM [25], the combined structure of CNN and LSTM [26, 27], and the deep convolution structure [28]. On this basis, literature [29] designs a keyword-based malicious URL detection model, which combines word embedding and GRU model. Literature [30] analyzes the structures and features of different URL, extracts more features, and proposes a semisupervised training model for URL's multiclassification. Literature [31] extracts URL domain name features and instantaneous features of redirection attacks and optimizes the neural network structure to improve detection accuracy. In recent years, it has become a new research direction to detect malicious URL directly. Literature [32] takes the original URL as the input of malicious URL detection system, transforms URL into feature vectors by character embedding technology, and then uses the convolutional neural network for training, which significantly reduces the dimension of data and the amount of computation. Additionally, it can help achieve a good classification effect. In literature [30], word embedding is used to transform URL in each message into vector-matrix, which is then inputted into the convolutional neural network for analysis. Literature [33] improves the detection algorithm and adds a convolution branch on the CNN structure to extract in-depth character-level features. The disadvantage of this method is that it adopts character embedding or word embedding alone, and it is difficult to extract features in both characters and phrases. Literatures [34, 35] adopt a parallel convolutional neural network to detect malicious URL. They combine character embedding and word embedding, improve the word embedding method in the vector embedding stage, and extract the URL's character and phrase features, which improve the detection effect. However, one of their disadvantages is that the fixed CNN structure is used to detect URL. The model parameters cannot be adjusted according to the input vector's dimension, so it is difficult to extract the in-depth features in a wide range.

Based on DCNN, this paper designs a malicious URL detection model to solve the abovementioned problem. We will detail this in the following chapters. The materials and methods section should contain sufficient detail so that all procedures can be repeated. It may be divided into headed subsections if several methods are described.

2. Malicious URL Detection Model

Our paper proposes a malicious URL detection model based on convolutional neural networks. The construction of the model is shown in Figure 1. The model mainly includes three modules: vector embedding module, dynamic convolution module, and block extraction module. In the following, we will introduce each module and the detection process in detail.

3. Vector Embedding Module

In our model, a URL is inputted into the embedding layer, and we use word embedding based on character embedding to transform the URL into a vector expression. Moreover, the URL will be input into the DCNN for feature extraction.

The vector embedding module represents the input URL sequence as a suitable vector to facilitate the subsequent process. In the beginning, URL vector representation is initialized randomly. It is then inputted into the embedding layer used in the subsequent training and the most appropriate URL vector expression is obtained during the training process. This module uses an advanced word embedding method based on character embedding. The module extracts the phase information from the URL and the character-level information from the word. The information extracted will be used in subsequent training to obtain the most appropriate vector expression of the URL, and then the vector expression is inputted into the subsequent convolutional layer.

An example of a word embedding method based on character embedding is shown in Figure 2, where k represents the embedding dimension of characters and words, L_2 represents the number of words contained in a URL string, and L_3 represents the number of characters contained in each word in the URL string. In the example, we convert each URL to an id sequence of word and character, respectively. Then, the embedding layer obtains word embedding matrix EM_w and character embedding matrix EM_c during the subsequent training. The word ID sequence uses the word embedding matrix EM_w to obtain the matrix expression URL_w at the word level. The character ID sequence uses the character embedding matrix EM_c to obtain multiple word matrix expressions based on character embedding. The multiple word matrices will be merged and compressed into a matrix representation URL_{cw} of the URL at the word level. The matrix representation URL_w and the matrix representation URL_{cw} are added. We will get the final representation of the URL.

3.1. Dynamic Convolution Module.

3.1. Dynamic Convolution Module. The dynamic convolution module is to extract features automatically from the input data. The processing procedure of data includes 1D convolution, folding, and dynamic pooling. The DCNN can adjust parameters and extract features in a wider range based on the input length and the current convolutional layer.

When the DCNN is training, the network's upper layer's output is inputted into the network's next layer. The URL is inputted into the input layer, and it is converted to a suitable vector expression in the embedding layer. Then, the first convolution layer starts to extract features. After the data are outputted from the convolutional layer, the data tensor dimension is compressed by the folding layer and then inputted to the pooling layer for dynamic pooling. After several rounds of convolution-folding-pooling, the data are finally inputted into the fully connected layer for training, and the result is finally outputted from the output layer.

3.2. Block Extraction Module. The block extraction module extracts different fields such as subdomain name, domain name, and domain name suffix from URL and encodes them as the second data branch of the detection model. In the embedding layer, the URL is converted to an appropriate vector. After passing through the embedding layer, the second data branch is merged with the first data branch, and the merged result is inputted to the fully connected layer for training. When the block extraction module extracts different fields, it can separate the top-level domain name or national domain name from the URL string.

The block extraction module can distinguish between generic top-level domains and national top-level domains. Therefore, the model can make full use of essential

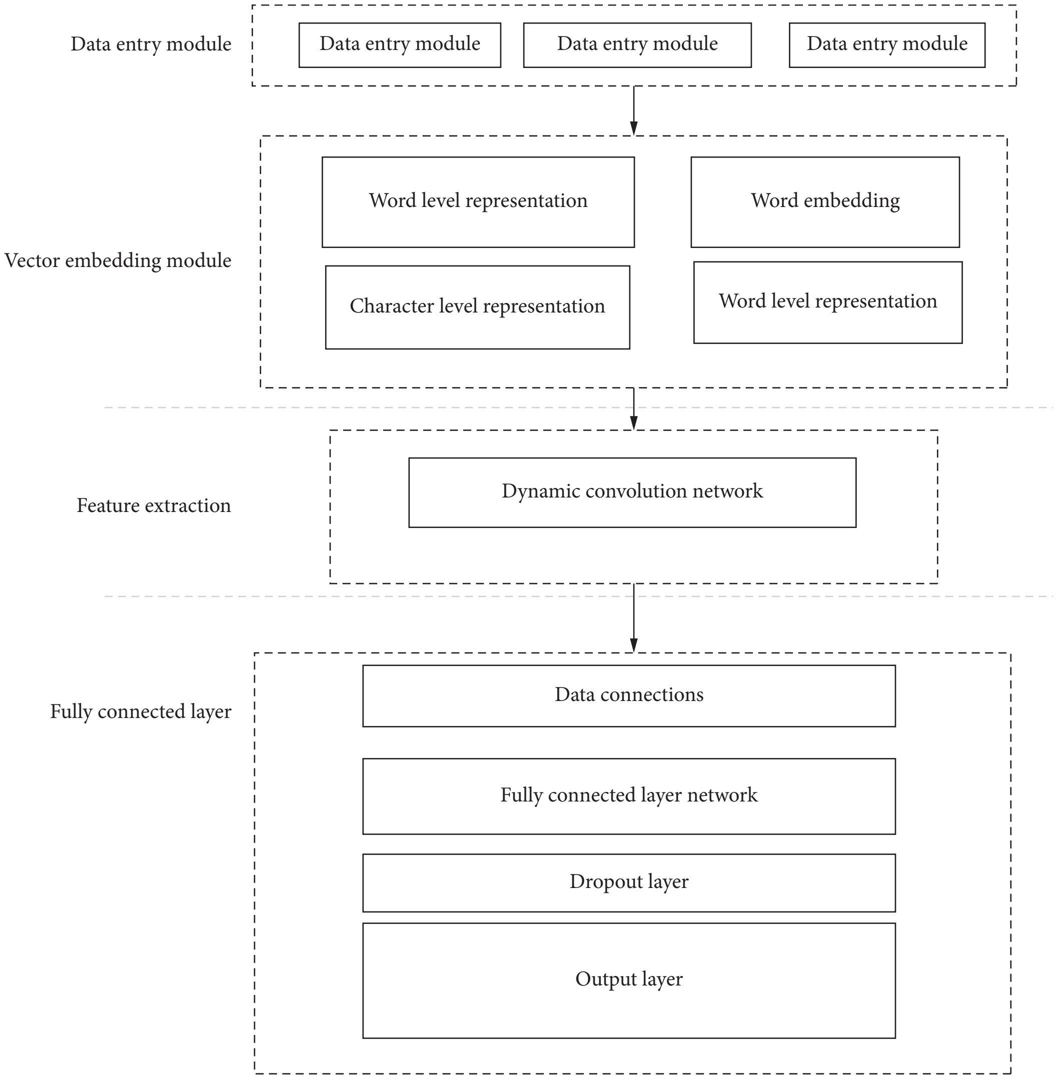


Figure 1: The detection model.

information. It takes the different fields in the URL as different features, which enriches the fully connected layer's input.

3.3. Detection Process. The detection process is as follows. First, domain name, subdomain name, and domain name suffix are sequentially extracted from URL. In the first branch of the detection model, we pad each URL to a fixed length, of which every word is marked with a specific number. The entire URL is represented as a sequence of numbers. Then, the sequences are inputted to the embedding layer and trained together with other layers. These sequences will learn the appropriate vector expression during the training process. The data stream outputted from the embedding layer is subsequently inputted into a DCNN. The output passes through a convolution layer, a folding layer, and a pooling layer in two successive rounds. In the flatten layer, the data stream is flattened. It then waits for connecting with data from the other branch, where domain name, subdomain name, and domain name suffix are marked. The different main domain name, subdomain name, or domain name suffix in each field are encoded as an independent expression. Then, the marked data are inputted into the three newly added embedding layers and obtain the appropriate vector expression. After that, the information is transformed into a suitable shape in the reshape layer and merged with the first branch's data. The two branches' outputs are combined and jointly inputted to the fully connected layer for training. We use the DCNN to extract features automatically and use different fields in URL as different features to detect malicious URL jointly. After the dropout layer, the result is outputted into the output layer.

This model can fully obtain the information carried by the different fields in the URL string and enrich the input of the fully connected layer, which improves the detection effect.

In summary, the branch of processing data is added for expanding the input of the detection model. When training in the fully connected layer, the features are extracted

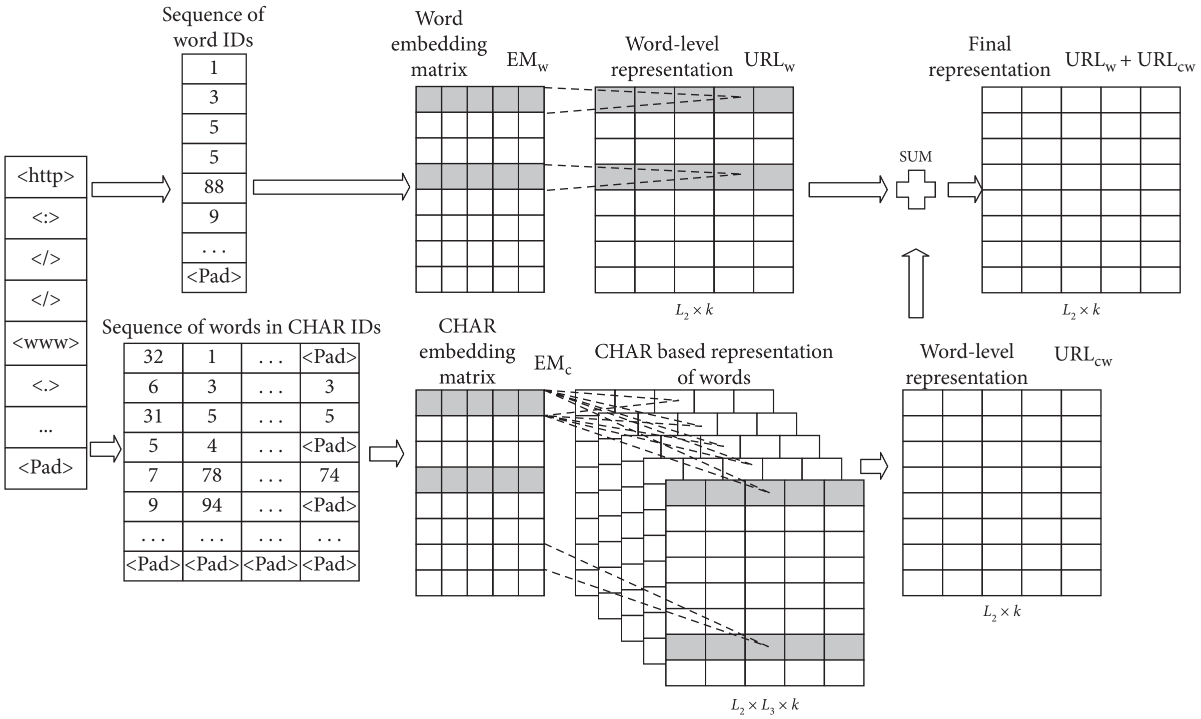


Figure 2: Example of a word embedding method based on character embedding.

automatically by the convolutional neural network and extracted artificially from the URL field. The detection model can effectively utilize critical information in the URL, such as top-level domain names and national domain names, to achieve higher accuracy and recall. Accuracy is vital, especially for detecting models, because if the accuracy is low, normal web pages may be classified as malicious websites and will be blocked.

4. Experiment and Evaluation

4.1. Experimental Environment. The experimental environment is based on the Windows operating system. The processor is i5-7500, the memory is 8 GB, the programming language is Python 3.6, and the deep learning framework is TensorFlow.

4.2. Comparative Experiments between Different Embedding Methods. Our paper adopts the word embedding method based on character embedding, which considered the advantages and disadvantages of word embedding and character embedding. The advantages of word embedding based on character embedding are as follows:

(1) This method can effectively express rare words. It takes full use of character-level and word-level information. Therefore, this method can accurately represent rare words in URL.

(2) This method can reduce the scale of the embedded matrix and reduce memory space. Meanwhile, it helps to convert a URL to a more accurate expression.

(3) This method can convert new words that do not exist in training set into more accurate vectors, thereby extracting character information.

Based on different embedding methods, we conducted three comparative experiments. Experiment 1 adopted character embedding. Experiment 2 utilized word embedding as an embedding method. Experiment 3 used word embedding based on character embedding as an embedding method. These experiments used malicious DGA URL as the training set, and the network structure was stacked CNN. We measured the accuracy, F1-score, precision, and recall ratio to evaluate the results.

Attackers can communicate with the control center through a malicious DGA domain name. A malicious DGA domain name can be treated as a string during detection. However, compared with the real malicious URL, the malicious DGA domain name contains fewer characters.

The experimental result is shown in Table 1. We find that word embedding based on character embedding achieves the highest accuracy among the above two embedding methods through these experiments. The accuracy of word embedding based on character embedding is 0.958; the accuracy of character embedding and word embedding are only 0.923 and 0.954, respectively. The recall of word embedding based on character embedding achieves 0.976, which is higher than character embedding and word embedding. We also draw ROC curves and AUC curves among the three experiments. It is shown in Figure 3.

Table 1: Experiment results.

No.	Embedding method	Network structure	Accuracy	F1-score	Recall	Precision
1	Character embedding	Stacked CNN	0.923	0.926	0.964	0.890
2	Word embedding	Stacked CNN	0.954	0.953	0.931	0.977
3	Word embedding based on character embedding	Stacked CNN	**0.958**	**0.959**	**0.976**	**0.942**

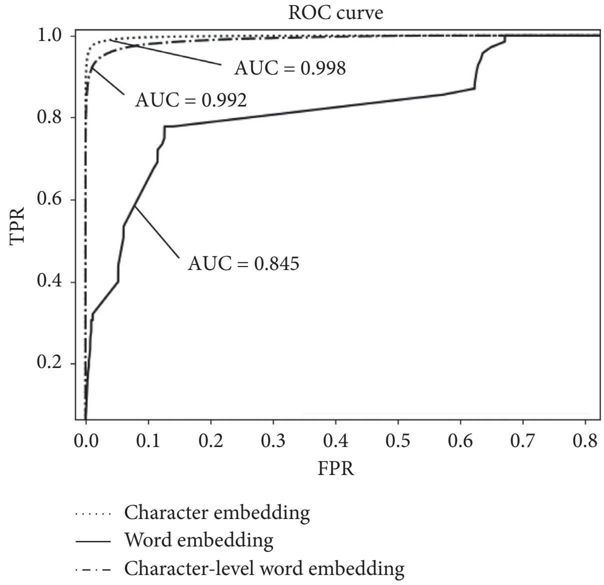


Figure 3: Comparison of ROC curves and AUC values.

4.2.1. Network Structure Experiment

(i) Experimental Dataset. We collect a large amount of URL data from github.com, uci.edu, and kaggle.com to verify the model's validity and feasibility. Besides, we select the top 1 million domain names, published on the Alexa website on May 17, 2019, as standard URL. Alexa ranks URL in terms of website traffic, including the site's top-level domains and subdomains. The dataset division is shown in Table 2; the distributions of the dataset are shown in Figures 4–6.

(ii) Experiment Setting. We designed three comparative experiments. We tried to use different network structures to determine the best solution for our model. Experiment 1 utilized the stacked CNN. Experiment 2 only leverages DCNN. Experiment 3 adopted DCNN, and it extracted different fields from the URL to participate in training. We set the same experimental parameters for these three experiments. Besides, each URL's length was set to 200 words, and the vector embedding dimension was 32. The DCNN included two convolutional layers. The number of convolution kernels was set to 128. The DCNN was finally trained by one fully connected layer, and it adopted the Adam optimization algorithm. The learning rate was 0.001, and the drop rate of the dropout layer was 0.5. In the process of the experiment, we adopted batch training, and each batch contains 100 URL.

Table 2: Dataset.

	Training set	Validation set	Test set
Malicious URL	200 k	25 k	25 k
Normal URL	200 k	25 k	25 k
Total	400 k	50 k	50 k

(iii) Experimental Results and Analysis. We measured the accuracy, F1-score, precision, and recall ratio to evaluate the results. The final results of the detection model in this paper are shown in Table 3. The accuracy reaches 0.987, the precision reaches 0.993, the F1-score reaches 0.987, and the recall ratio is 0.981. The accuracy and loss are shown in Figures 7 and 8. As the number of iterations increases, the training accuracy increases continuously, and the fitting degree of the model is ideal. At the same time, the loss continues to decrease.

We also list the experimental results of other comparative experiments, as shown in Table 4; the network structure DCNN + extracting fields obtained a better effect than the other two network structures. The accuracy reaches 0.987, and the precision is 0.993. The ROC curves are drawn, and

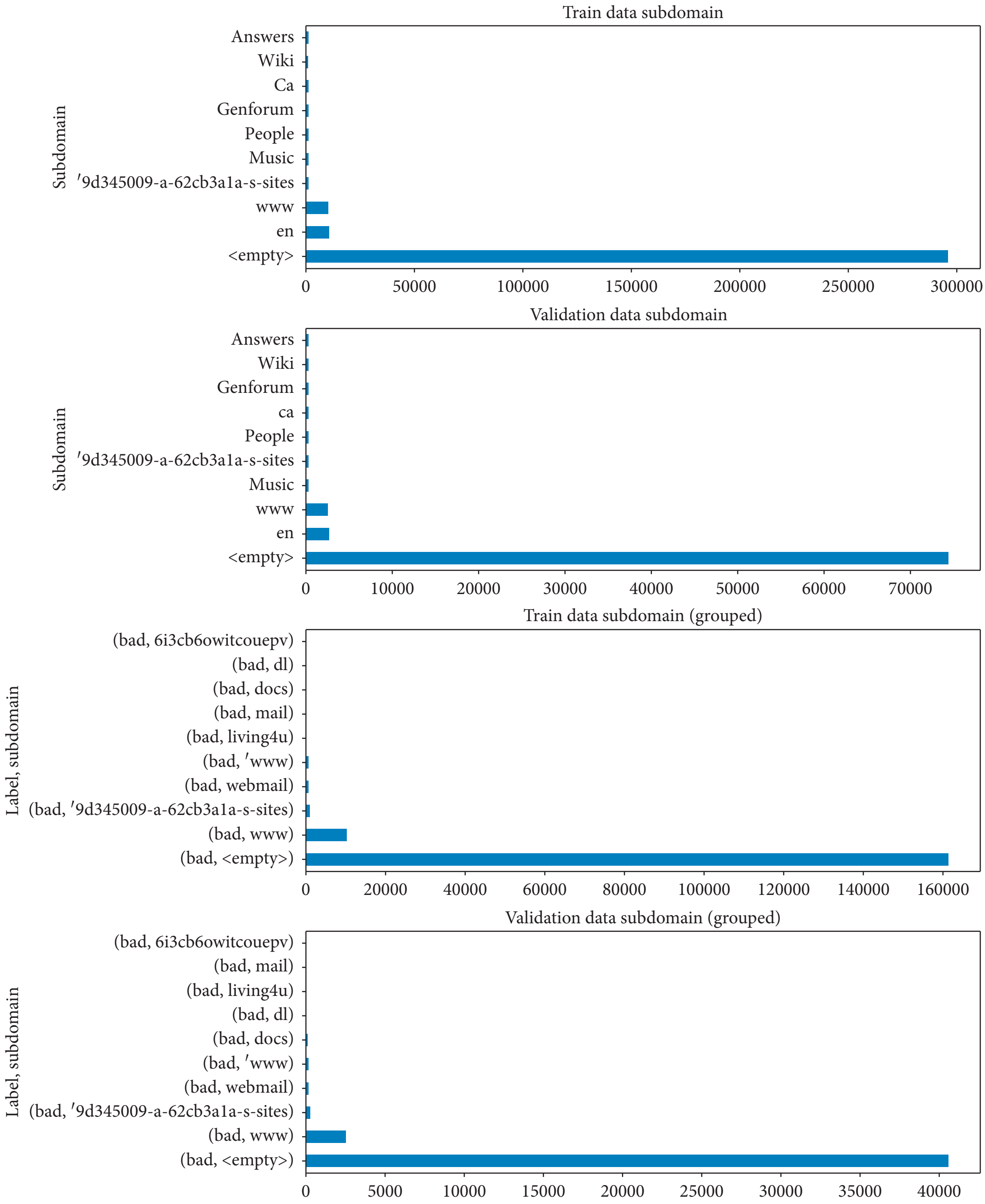


Figure 4: Distribution of data subdomains.

the AUC value is calculated to facilitate comparison and analysis. They are shown in Figure 9. We can see the TPR of DCNN + extraction is the first to reach 0.9993.

From the results, it can be seen that the best detection effect is obtained in Experiment 3. The high accuracy indicates that benign samples are less likely to be misjudged as malicious samples. Experiments show that the URL can be adequately expressed, critical features can be extracted to obtain better detection results using DCNN and the word embedding based on character embedding. Extracting different fields from the URL can make full use of keywords in the URL to improve detection accuracy and precision. The abovementioned experiments verify the validity of the detection model in this paper.

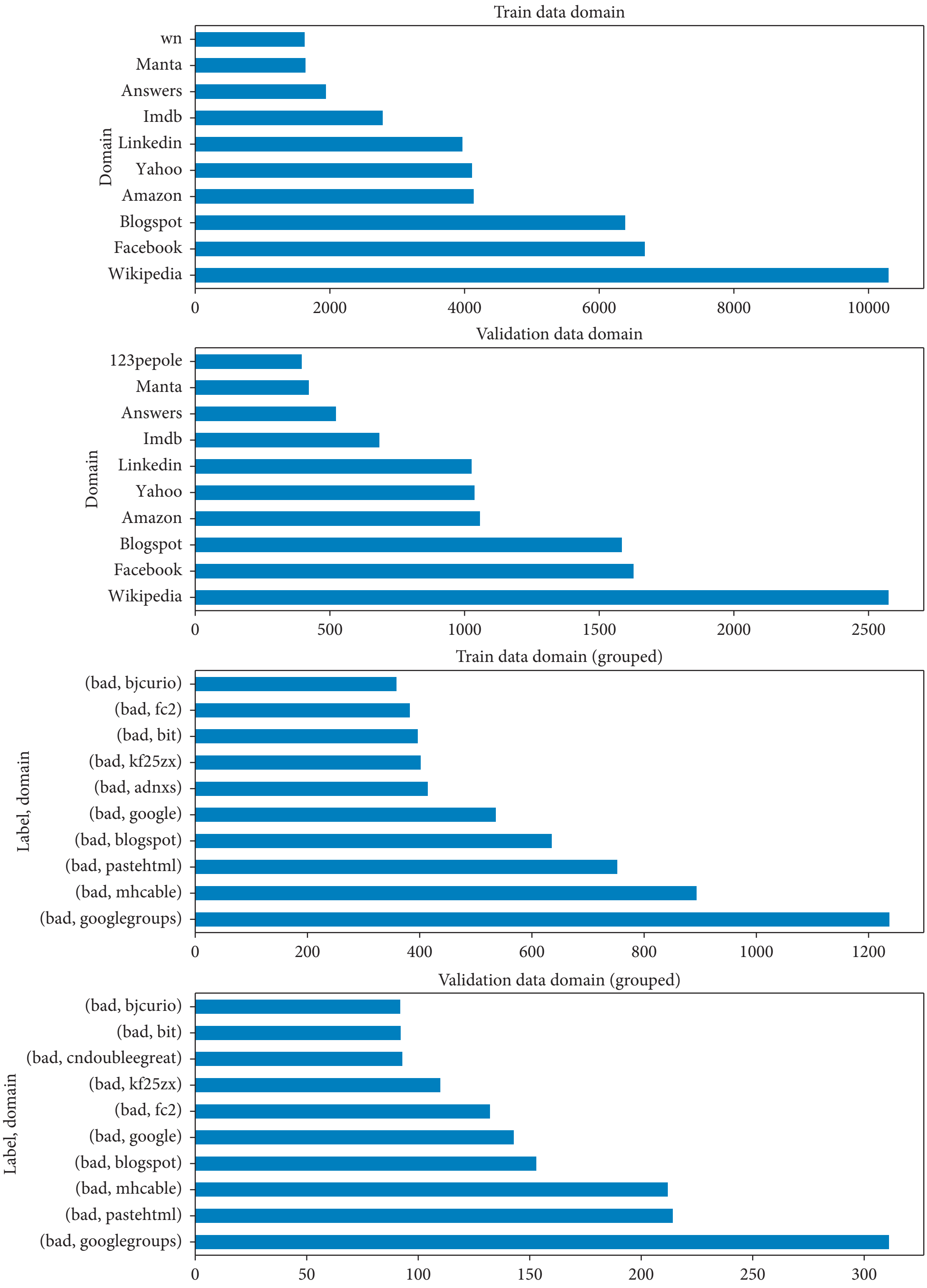


Figure 5: Distribution of data domain.

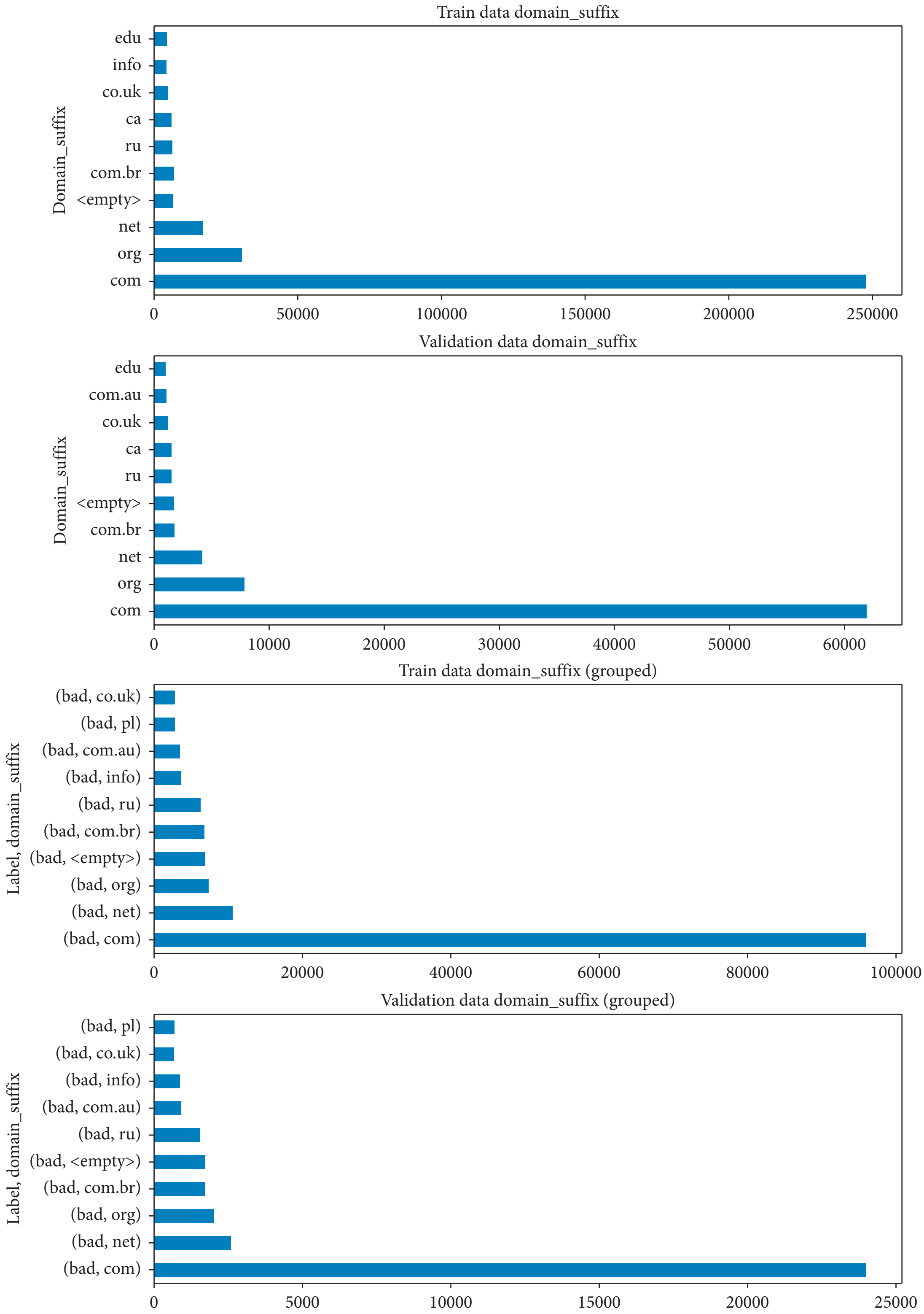


Figure 6: Distribution of data domain_suffix.

Table 3: Experimental results.

Detection indicator	Meaning	Value
Accuracy	((TP + TN)/P + N)	0.987
F1-score	(2TP/(2TP + FP + FN))	0.987
Recall	(TP/(TP + FN))	0.981
Precision	(TP/(TP + FP))	0.993

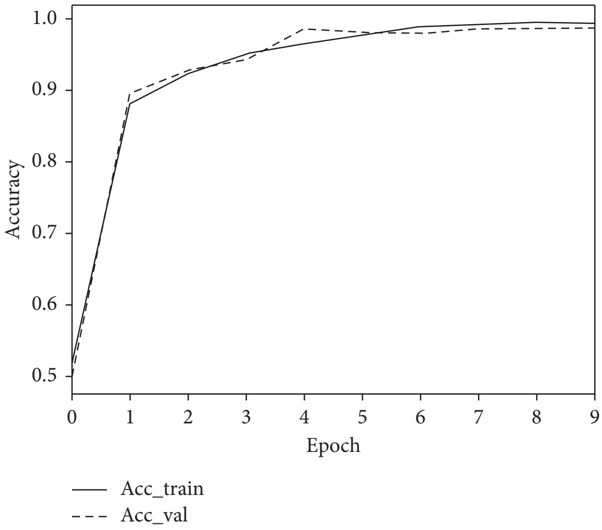


Figure 7: Training accuracy and verification accuracy.

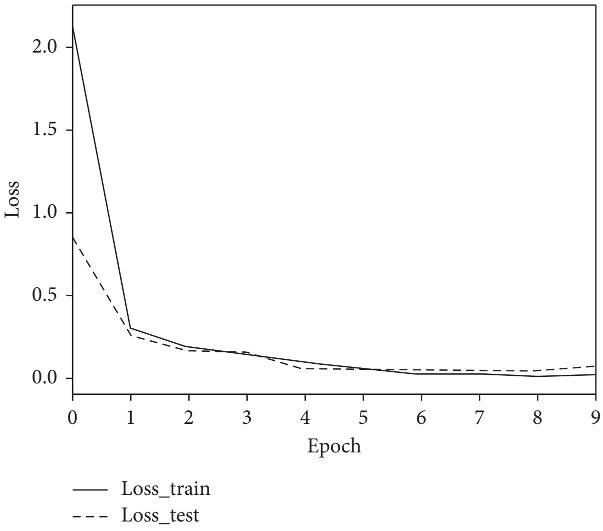


Figure 8: Training loss and validation loss.

Table 4: Experiment results.

No.	Embedding method	Network structure	Accuracy	F1-score	Recall	Precision
1	Word embedding based on character embedding	Stacked CNN	0.958	0.959	0.976	0.942
2	Word embedding based on character embedding	DCNN	0.961	0.960	0.936	0.984
3	Word embedding based on character embedding	DCNN + extracting fields	**0.987**	**0.987**	**0.981**	**0.993**

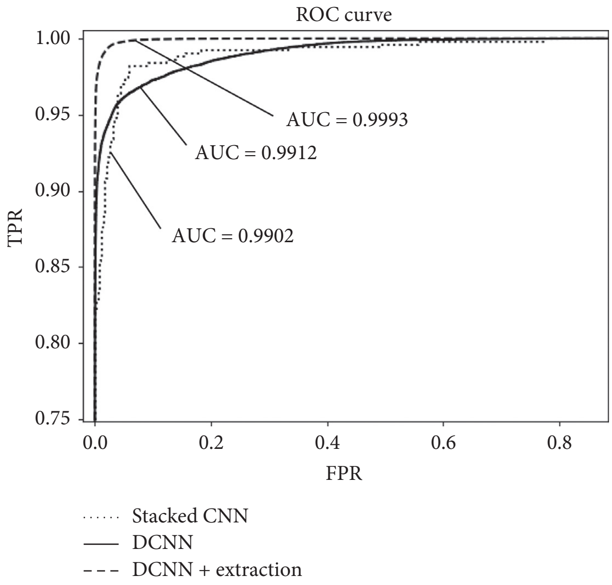


Figure 9: Comparison of ROC curves and AUC values in three experiments.

5. Conclusions

This paper aims to design a new malicious URL detection model based on deep learning. We designed a word embedding method based on character embedding, and the vector expression of a URL is learned automatically by combining character embedding with word embedding. DCNN for malicious URL detection is designed. According to the length of the input vector and the depth of the current convolution layer, the pooling layer parameters are dynamically adjusted to extract features in a wider range. We coordinated the relationship between different modules and adjusted network parameters. Besides, the real URL and malicious DGA URL in the real network are collected, and a series of experiments are designed and conducted. The results and various indicators are compared and analyzed to demonstrate the validity of the detection model.

The detection model achieves the expected effect in experiments. However, considering that the network traffic in the test environment and the real network are different, and with the development of the Internet, types of malicious URL are more diverse. It is necessary to timely update the model in the actual scenario. Therefore, to better adapt to the requirements of various complex application scenarios, we plan to study how to simplify the detection model's architecture and shorten the training time while keeping the detection performance unchanged in the future.

Data Availability

The URL data used to support the findings of this study have been deposited in the malicious and benign URL dataset: https://gitee.com/blackwall/UrlDetect/blob/master/urldetect/%E5%AE%9E%E9%AA%8C%E4%BB%A3%E7%A0%81.zip.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge the Key Laboratory of Network Assessment Technology of the Institute of Information Engineering. This research was financially supported by the Key Lab of Information Network Security, Ministry of Public Security (C19614), China Postdoctoral Science Foundation (2019M650606), and First-Class Discipline Construction Project of Beijing Electronic Science and Technology Institute (3201012).

References

[1] D. J. Lemay, R. B. Basnet, and T. Doleck, "Examining the relationship between threat and coping appraisal in phishing detection among college students," *Journal of Internet Services and Information Security*, vol. 10, no. 1, pp. 38–49, 2020.

[2] H. Kim, "5G core network security issues and attack classification from network protocol perspective," *Journal of Internet Services and Information Security*, vol. 10, no. 2, pp. 1–15, 2020.

[3] K. Aram and J. O. SoK, "A systematic review of insider threat detection," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 10, no. 4, pp. 46–67, 2019.

[4] R. B. Basnet and R. Shash, “Towards detecting and classifying network intrusion traffic using deep learning frameworks,” *Journal of Internet Services and Information Security*, vol. 9, no. 4, pp. 1–17, 2019.

[5] F. Valenza and M. Cheminod, “An optimized firewall anomaly resolution,” *Journal of Internet Services and Information Security*, vol. 10, pp. 22–37, 2020.

[6] D. R. Patil and J. B. Patil, “Survey on malicious web pages detection techniques,” *International Journal of U- and E-Service, Science and Technology*, vol. 8, no. 5, pp. 195–206, 2015.

[7] B. Yu, J. Pan, J. Hu, A. Nascimento, and M. De Cock, “Character level based detection of DGA domain names,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Rio de Janeiro, Brazil, July 2018.

[8] C. Choudhary, R. Sivaguru, M. Pereira, B. Yu, A. Nascimento, and M. De Cock, “Algorithmically generated domain detection and malware family classification,” in *Proceedings of the International Symposium on Security in Computing and Communication*, pp. 640–655, Springer, Singapore, September 2018.

[9] S. Garera, N. Provos, and M. Chew, “A framework for detection and measurement of phishing attacks,” in *Proceedings of the ACM Workshop on Recurring Malcode*, ACM, New York, NY, USA, November 2007.

[10] D. K. M. M. Gupta, “Behind phishing: an examination of phisher modi operandi,” in *Proceedings of the Usenix Workshop on Large-scale Exploits & Emergent Threats*, DBLP, San Francisco, CA, USA, April 2008.

[11] J. Ma, L. K. Saul, and S. Savage, “Beyond blacklists: learning to detect malicious Web sites from suspicious URL,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Paris, France, July 2009.

[12] H. Choi, B. B. Zhu, and H. Lee, “Detecting malicious web links and identifying their attack types,” in *Proceedings of the 2nd USENIX conference on Web application development*, Boston, MA, USA, June 2011.

[13] K. Bartos, M. Sofka, and V. Franc, “Optimized invariant representation of network traffic for detecting unseen malware variants,” in *Proceedings of the USENIX Security Symposium*, pp. 807–822, Austin, TX, USA, August 2016.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, The People’s Posts and Telecommunications Press, Beijing, China, 2016.

[15] Z. Feng, C. Shuo, and W. Xiaochuan, “Classification for DGA-based malicious domain names with deep learning architectures,” in *Proceedings of the Second International Conference on Applied Mathematics and information technology*, p. 5, Shanghai, China, October 2017.

[16] C. Xu, J. Shen, and X. Du, “Detection method of domain names generated by DGAs based on semantic representation and deep neural network,” *Computers & Security*, vol. 85, pp. 77–88, 2019.

[17] K. Yoon, “Convolutional neural networks for sentence classification,” 2014, https://arxiv.org/abs/1408.5882.

[18] X. Zhang, J. Zhao, and Y. Lecun, “Character-level convolutional networks for text classification,” 2015, https://arxiv.org/abs/1509.01626.

[19] J. Clayton and K. Bishal, “Towards detecting and classifying malicious url using deep learning,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 11, no. 4, pp. 31–48, 2020.

[20] Z. U. O. Wen, *Research and Design of Malicious URL Detection Algorithm on Deep Learning*, Beijing University of Posts and Telecommunications, Beijing, China, 2019.

[21] J. Yang, P. Yang, X. Jin, and Q. Ma, “Multi-classification for malicious URL based on improved semi-supervised algorithm,” in *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 143–150, IEEE, Guangzhou, China, July 2017.

[22] T. Shibahara, K. Yamanishi, and Y. Takata, “Malicious URL sequence detection using event de-noising convolutional neural network,” in *Proceeings of the Communications (ICC), 2017 IEEE International Conference*, pp. 1–7, IEEE, Paris, France, May 2017.

[23] J. Woodbridge, H. S. Anderson, and A. Ahuja, “Predicting domain generation algorithms with long short-term memory networks,” 2016, https://arxiv.org/abs/1611.00791.

[24] B. Dhingra, Z. Zhou, and D. Fitzpatrick, “Tweet2Vec: character-based distributed representations for social media,” 2016, https://arxiv.org/abs/1605.03481.

[25] S. Vosoughi, P. Vijayaraghavan, and D. Roy, “Tweet2vec: learning tweet embeddings using character-level CNN-LSTM encoder-decoder,” in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 1041–1044, ACM, Pisa, Italy, July 2016.

[26] Y. Liu, K. J. Zhao, S. Ge lian, and Y. Liu, “A fast DGA domain algorithm based on deep learning,” *Journal of Shandong University(Natural Science)*, vol. 54, no. 7, pp. 106–112, 2019.

[27] Z. Ming, B. Xu, and B. Shuai, “A deep learning method to detect web attacks using a specially designed CNN,” in *Proceedings of the International Conference on Neural Information Processing*, Bangkok, Thailand, November 2017.

[28] M. Antonakakis, R. Perdisci, and Y. Nadji, “From throw-away traffic to bots: detecting the rise of DGA-based malware,” in *Proceedings of the 21th USENIX Security Symposium*, Bellevue, WA, USA, August 2012.

[29] Y. Lu, G. Liu, Z. Jiang tao, W. Liu, B. h. wen, and Y. Dai, “Improved algorithm for detecting the malicious domain name based on the convolutional neural network,” *Journal of Xidian University*, vol. 1, no. 12, pp. 1–8, 2019.

[30] J. Saxe and K. Berlin, “eXpose: a character-level convolutional neural network with embeddings for detecting malicious url, file paths, and registry keys,” 2017, https://arxiv.org/abs/1702.08568.

[31] S. Schiavoni, F. Maggi, and L. Cavallaro, “Phoenix: DGA-based botnet tracking and intelligence,” in *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, Cham, Switzerland, March 2014.

[32] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modeling sentences,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 655–665, ACL, Baltimore, MD, USA, June 2014.

[33] L. Hung, Q. Pham, D. Sahoo, C. Steven, and H. Hoi, “URLNet: learning a URL representation with deep learning for malicious URL,” 2018, https://arxiv.org/abs/1802.03162.

[34] Y. Shi, G. Chen, and J. Li, “Malicious domain name detection based on extreme machine learning,” *Neural Processing Letters*, vol. 48, no. 3, pp. 1347–1357, 2018.

[35] Z. Wang, S. Li, B. Wang, X. Ren, and T. Yang, “A malicious URL detection model based on convolutional neural network,” in *Proceedings of the International Symposium on Security and Privacy in Social Networks and Big Data*, pp. 34–40, Tianjin, China, September 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5513590, 7 pages
https://doi.org/10.1155/2021/5513590

Research Article

A Multimodality Information Synchronization Scheme for a Multisource Information System in the Electric Grid

Pan Zhang,[1] Fenggang Lai,[1] Jing Du,[1] Wenda Lu,[2] and Xiao Yu[3]

[1]*State Grid Information & Telecommunication Co., Ltd, Beijing 100761, China*
[2]*Information & Telecommunication Branch, State Grid Zhejiang Electric Power Corporation, Hangzhou 310000, China*
[3]*Department of Computer Science and Technology, Shandong University of Technology, Zibo 255022, China*

Correspondence should be addressed to Xiao Yu; yuxiao8907118@163.com

Received 5 February 2021; Accepted 23 April 2021; Published 6 May 2021

Academic Editor: Zhe-Li Liu

A multisource and heterogeneous database is an important problem that disturbs the use of the electric power information system. The existing database synchronization scheme has some problems in practical applications, such as high resource loss and poor portability. This paper presents a high-efficiency database synchronization scheme for the electric power information system. The database is monitored, and its changes are captured by the shadow table and trigger method. Thus, data could be exchanged in trusted networks and nontrusted networks. In addition, a predetermined strategy is used to avoid data conflicts and ensure consistency and reliability of data synchronization. The above method is applied in the protection system of power networks. The results show that the synchronization scheme can effectively ensure the security of the system and has higher synchronization efficiency.

1. Introduction

The scale of the data network of the electric power information system increases rapidly with the development of the smart grid. Data scheduling is filled with dispatch centers, power plants, users, and so on. The security problems need to be researched and solved for services, especially for data security issues that are important and restrict the application in the electric power information system.

The telemechanical system controls the power station and substation through the telemetry data of the system [1–7]. Its security and reliability directly affect the safety and reliability of the whole power control system. Because the telemechanical system is interconnected with the local MIS (management information system) or connected to the internet, the power system design and construction, and lack of data network protection, the telemechanical system is easy to be attacked.

Network attacks occur frequently. Trojan horses, worms, and ransomware emerge one after another on the internet, which pose a serious threat to the safety in the production of the smart grid. Therefore, just like government networks, military networks, and other classified networks, the security of the power information system, especially the production-related real-time data networks and dispatching data networks, should carefully formulate targeted strategies and set up a strong security guarantee system [3].

The traditional data backup method is to ensure the data security by the database maintainers under the premise of a continuously open network. However, in some networks where the degree of secrecy is relatively high, database synchronization should be implemented in a secure and isolated environment. Therefore, the new technique of database synchronization between isolated networks is worth further study.

The general network isolation scheme is implemented by installing a network isolation device between trusted intranet and nontrusted internet [8–19]. The principle is based on the idea of access control and physical isolation and defines relevant constraints and rules to guarantee the security strength of the network.

Our inspiration comes from the object-oriented database system and distributed client-server mechanism. On the basis of the isolated network, we put forward a new efficient data synchronization scheme. In the scheme, the shadow table and trigger method are used to capture the data changes in the database. In this case, data between intranet and internet are synchronized without any disruption to system security. We implement two synchronous techniques to evaluate the scheme in the protection system of the information system of the smart grid. Experiment results show that our method could ensure system safety and achieve excellent performance.

The rest of the paper is organized as follows. We describe the overall framework of the isolation system in Section 2. Section 3 describes database synchronization in network isolation. We discuss our implementation in Section 4 and analyze numerical results in Section 5. Finally, we conclude our paper in Section 6.

2. Overall Structure of Network Isolation

Network isolation is a physical isolation method by a special isolation device between inner intranet and outer internet [19–28]. Network isolation technology ensures that the internal information of the trusted network is not leaked and uses shared storage to complete the safe exchange of data between networks [20–23].

In general, isolation devices are not connected to either trusted intranet or nontrusted internet. If there are requests for information exchange between them, the isolation device tries to connect to one of the two networks. Figure 1 shows the isolation structure used in the electric power information system.

The network isolation system abandons common network protocols such as TCP/IP and adopts a new type of proprietary security protocol to exchange data. The system blocks the TCP/IP connection, makes the internal network and external network completely lose the connection, and completely eliminates the hidden danger of the TCP/IP network attack. In addition, the system can effectively reduce the attack threat by using the vulnerability of the OS with the support of isolated hardware and software.

3. Principal of Database Synchronization

3.1. Description of Database Synchronization. Due to the physical isolation characteristics of the network isolation environment, the database distribution in two isolated networks includes the following three synchronization steps: capture changes of the source data, distribute the data, and update the data to the target database [23–28]. Figure 2 shows the structure of data synchronization. The process of data synchronization can be divided into several processes as follows: change capture, data distribution, network transmission, synchronization monitoring, and data update [29–32].

In data synchronization, the first step is to capture the data changes in the database. Thus, a combination of triggers and shadow tables is used. This approach uses an XML file as an intermediary to log changes' information such that every table of the source database corresponds to an XML file.

Data distribution adopts the server-client mode, in which each network device acts as a client to actively connect to the server. A monitor module that is continuously running is used to record the update of the source database and target database, respectively. As is mentioned in the graph, the data update module is used to update the data to the target database [33]. When the update operation is executed, the communication module starts to send (or receive) data according to the synchronous configuration policy. Thus, an entire synchronization procedure is completed.

3.2. Process of Database Synchronization. Data synchronization is the process of synchronizing data from the source database to the target database. Maintaining consistency of multiple copies of replicated objects and considering the synchronization efficiency can effectively reduce network overhead and shorten response time, thus improving the availability and reliability of the whole system. The synchronization approach takes the following three steps.

3.2.1. Change Capture. Change capture is the basis of synchronization, and it directly determines how the database is updated and how time is selected for synchronization. Changing sequence information is essential when synchronizing the target database. In addition, a large amount of control information needs to be synchronized as well [1, 34]. Due to different characteristics of the trigger method and shadow table method, the combined method of trigger and shadow table is adopted to ensure the performance of the source database and minimize the impact on the system operation.

(i) Trigger method: this method is especially suitable for a large amount of data and often needs incremental synchronization. Trace triggers are created for data operations such as Insert, Delete, Modify, and Update in the source data table. When one of the above operations occurs on the source table, newer field data, action type, and action sequence number are stored in the log table, which provides synchronous updates to the source table.

(ii) Shadow table method: this method is generally used in scenes with a small amount of data and low requirement of the real-time performance. The advantage of this method is that it has little impact on the business system and easy to deploy. When the source table needs to synchronize, a supporting shadow table is created to record the change tracking table. After that, the shadow table and source table are compared to extract the changing information. In this case, the shadow table is synchronized.

In the above combined synchronization method, data distribution could obtain synchronization information by the source table and change tracking table. Thus, the work of change capture is integrated in distributed modules and then encapsulated into the database layer [2, 23].

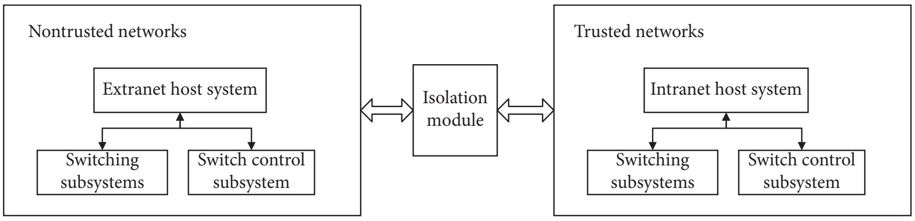


Figure 1: Structure of the network isolation system.

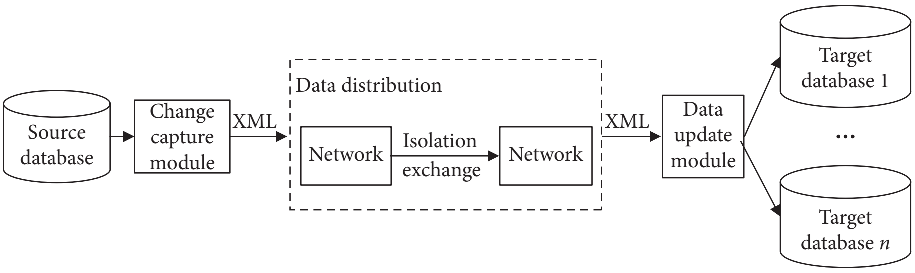


Figure 2: Overall structure of database synchronization.

Change capture is the process of capturing the sequence of changes in the source table. Based on the configuring parameters of the synchronization mode, the system automatically creates the tracking table and shadow table for the source table's change capture. The main idea of data synchronization technology to capture changes is to create a change tracking table for multiple related source tables, perhaps a single source table, or all tables of an entire database. Source table's field information is recorded, such as the type of operation, the sequence number of operations, the operation time, and the change's key field information.

3.2.2. Monitoring the Synchronization. Because of the independence of the JDBC platform, it has become one of the most popular methods to access the database. In this case, JDBC is used to connect to the database in the synchronous monitoring module. If there are changes in the source database, the synchronization system will start the communication module according to the synchronization mode.

3.2.3. Data Distribution. The primary purpose of data distribution is to implement change information from the source table to the corresponding target table. Based on the above change capture method in synchronization, the data distribution module obtains the corresponding SQL statements in sequence according to the sequence number in the change tracking table. The dispatcher then executes the SQL statement on the target server and applies the changes in the source table to the target table. After successful execution, the record corresponding to the sequence number in the change tracking table is deleted. The dispatcher is a coordinator and is also responsible for passing control information and mediation if replication conflicts are found.

3.2.4. Data Update. Data updates occur on network target nodes. When the XML file is sent to the network node, the SQL statement is immediately extracted. If the instruction contains one of the Insert, Delete, or Update operations, the SQL statement is executed directly, and the target table is updated accordingly. In particular, if it is a Create operation, the target database will create a new synchronization table and initialize it.

3.3. Conflict Detection and Resolution. Replication operations incurred by data synchronization can cause inconsistencies and conflicts between different copies [35, 36]. It is necessary to determine the cause and location of the conflict and resolve the conflict in accordance with the predetermined strategy. In addition, the granularity of conflict detection, record level, field level, and so on, will affect the performance of the system. Therefore, a control information table was set up in the prototype to help resolve the conflict. The structure of the control information list is shown in Figure 3.

As shown in Figure 3, all pieces of control information are organized into a list, which is indexed by metadata. Each information item consists of the conflicting data item, conflict position, operator, and timestamp. The main function of each of these information items is as follows:

(1) Conflicting data item: it provides a change copy of the conflicting data items, which can reflect relationships between the changes of data items

(2) Conflict position: it identifies the subject causing the conflict

(3) Operator: it indicates the location that caused the conflict, which may contain multiple server nodes

(4) Timestamp: it provides the exact time when the conflict occurred

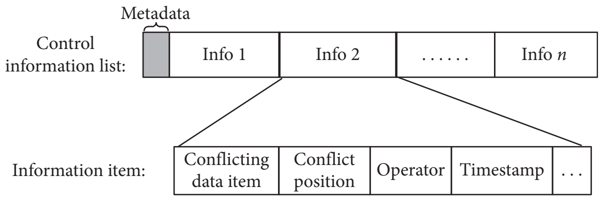


Figure 3: Structure of the control information list.

4. Network Isolation Deployment in the Electric Power Information System

4.1. Network Isolation Device. In addition to using basic firewalls and proxy servers for security, the power information system also uses network isolators as gateways. The isolation device adopts isolation technology that the intranet and internet disconnect to the isolation device.

The isolated transmission mechanism uses dedicated hardware and security protocols to achieve data exchange between internal and external networks. The data monitoring module has powerful control and management function by security mechanisms, such as access control, identity authentication, and encryption.

4.2. Isolation Scheme in the Electric Power Information System. The whole topology of the telemechanical network and MIS network of the smart grid is shown in Figure 4. Figure 4 shows that the telemechanical system could connect directly to the dispatch center and production site without network isolation. Furthermore, as the middle system of the MIS and internet connection, once the telemechanical system is deliberately destroyed by a virus or hacker, the whole power network will face serious risks.

To guarantee the reliability of the telemechanical system and intranet, network isolation should be implemented between the telemechanical system and MIS networks [29], and the original industry network will be protected. The connection between internet and remote database will also be blocked. In the MIS network, an image database of the remote database is created to replace the original remote database to provide services to the system.

In the opposite case, the image database should be synchronously updated to the original telemechanical database. The telemechanical web system is deployed on intranet so that the telemechanical engineer can browse the telemechanical data. Figure 4 shows the working scene that the original telemechanical database is the source database, and the mirror database is the target database.

4.3. Deployment and Configuration of Synchronization. The synchronization system consists of the sending end and receiving end. Typically, the sending end is deployed on a remote database server, and the receiving end is deployed on a mirrored database server to save resources [37, 38].

In the initial state of the synchronization, most of the tables need to be synchronized first, and then the tables need to be synchronized selectively. Because there are too many tables in the remote database to synchronize every table in real time, tables are categorized as follows:

(1) Nonsynchronous tables: these tables are always used in the external network, such as parameter tables for data acquisition, evaluation parameter settings, and intermediate evaluation calculation tables.

(2) Synchronous tables: these tables are always involved in synchronous operations. Depending on the size of these tables and how often they are updated, appropriate synchronization method needs to be selected.

5. Experiment Results and Analysis

5.1. Experiment Environment. For our experiments, we used Dell PowerEdge Server which has Intel Xeon processors running at 3.20 GHz and 32 GB RAM. The OS is Windows Server 2016, and the test DBMS includes MySQL, Oracle, and SQL Server. The inner and outer networks are separated by special electric power information system isolation devices. During the prototype system operation, unnecessary programs and processes are closed as much as possible to minimize resource usage.

5.2. Operational Approach. After the prototype was implemented, data replication function was tested. On the image database server, an image database is created to test the performance of synchronization. Next, a synchronous receiver program is run to monitor the network at port 9098 connecting the database successfully.

On the source database server, the receiver is always running. After synchronization table configuration is completed, a series of actions including change tracking table, change tracking trigger, shadow table, and incremental change analysis are performed over and over again.

In general, system synchronization will read update data from source tables incrementally on the basis of the preconfiguration and then update to external internet. All of the above processes need no manual interventions.

5.3. Performance Analysis. There are approximately 3.5 million records in the test database, each of which is about 100 bytes long. We test the time efficiency and space efficiency of the system by randomly selecting some of record items. Figures 5 and 6 show the experiment results, respectively.

Figure 5 shows that time efficiency of the data synchronization technique increases linearly with the increase of the record number. Generally, time consumption of every record is smaller when there are more records. On the contrary, when few records are recorded, more time is spent. A reason for this could be that data synchronization adopts the method of copying by reading control information afterwards, and it brings limitations on synchronization efficiency. In terms of space efficiency, as shown in Figure 6, it remains a significant constant that indicates the system has been stable and reliable.

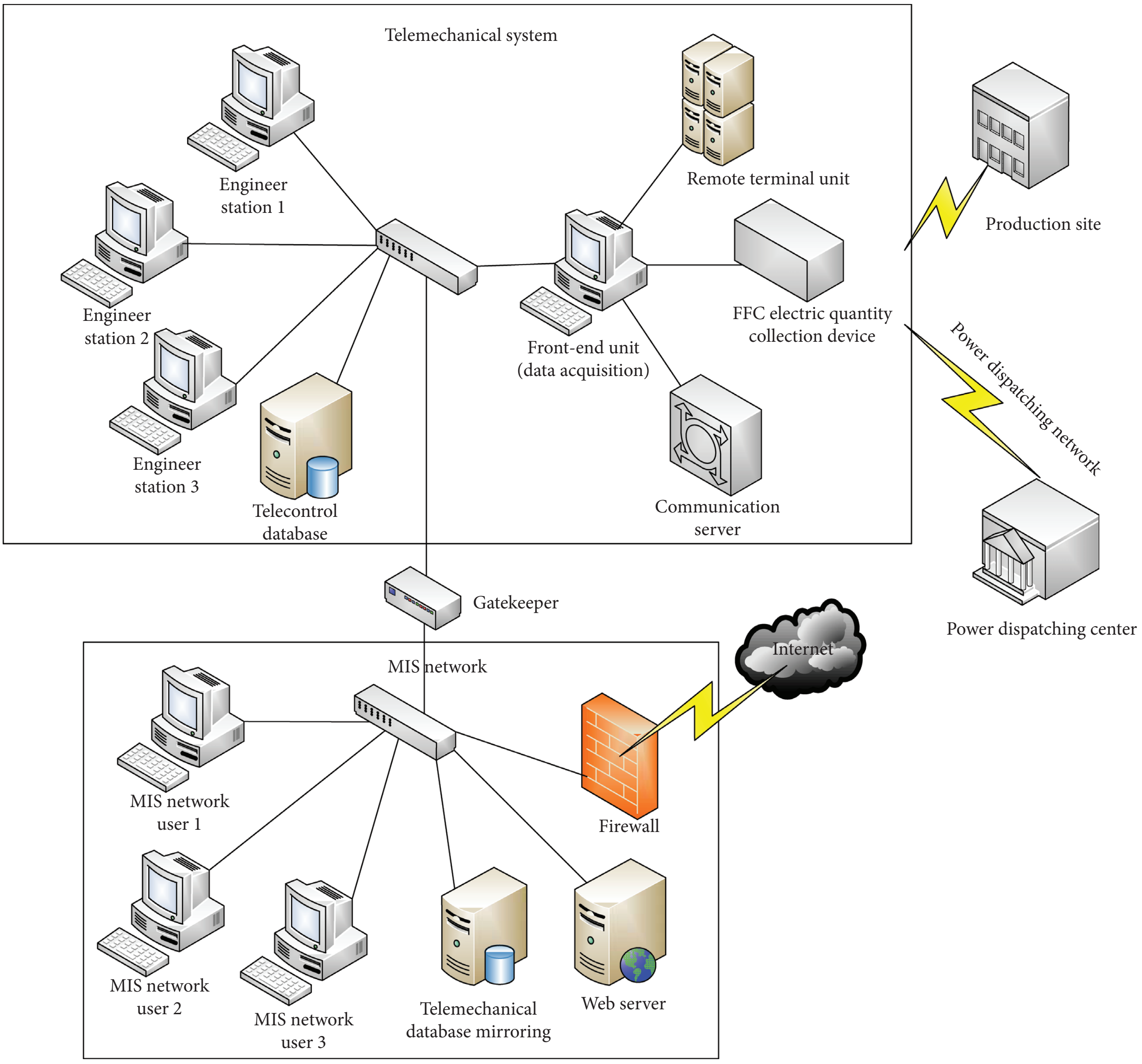


Figure 4: Network isolation topology.

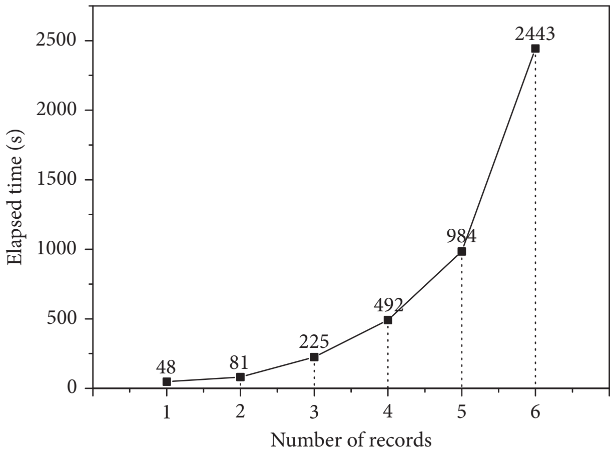


Figure 5: Test result of time efficiency.

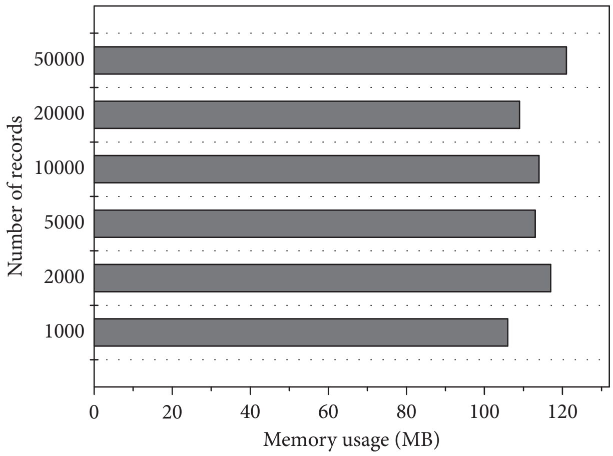


Figure 6: Test result of space efficiency.

Table 1: Time efficiency and resource efficiency of initialization/real-time synchronization.

Work type	CPU utilization (%)	Memory usage (MB)	Response time (ms)
Initialization synchronization	50	170	0.75
Real-time synchronization	≤10	110	6.0

We suppose the isolation network environment is an insecure and vulnerable network that is easy to be attacked by viruses and hackers. When the external database system is damaged, the restoration of the external server can quickly complete the reconstruction of the mirror database and normal operation of synchronization.

When the system performs operations of initializing the target table, the speed of synchronization can reach 80,000 per minute. So, the target database could complete initialization in 1 hour. In the initialization process, the receiver also needs to update the target database constantly, so the CPU utilization rate is about 50% that is still relatively higher. Memory consumption is about 170 MB.

When the target table of the database has been initialized, the remaining operations are incremental updates. Since then, the speed of synchronization is about 10,000 records per minute. In this case, the CPU utilization is at a relatively low level, less than 10%, and the memory consumption is about 110 MB. The main working parameters of initialization synchronization and real-time synchronization are given in Table 1.

It is worth mentioning that the system has been stable after entering the working state. Data in the mirror database could be updated immediately, and end users can barely feel the corruption of the system. With the same method, the system can also easily recover the corrupted data of external internet.

6. Conclusion

The hybrid isolation synchronization scheme for the electric power information system realizes secure and high-performance data exchange between trust and nontrust networks. This scheme is not only suitable for the isomorphic databases but also suitable for heterogeneous databases. With the data exchange captured by the hybrid method, the synchronization scheme could record and query all of the synchronization data.

Data Availability

The data are not available because of the supported funding privacy policy.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was sponsored by State Grid Technology Project "Research and Application of Key Technologies in Enterprise-Level Data Center Platform based on Full Service Unified Data Center" (Grant no. 5211XT190033) and Zhejiang Lab (2020LE0AB02). This work was supported in part by the National Natural Science Foundation of China (nos. 61876019 and 62072037).

References

[1] H. Cui, J. Feng, W. Ma, Q. Zhang, K. Pang, and K. Yu, "Research and implementation of heterogeneous database synchronization technology based on isolation gateway," *Software Engineering*, vol. 19, no. 2, pp. 10–13, 2016.

[2] S. P. Kumar, "Adaptive consistency protocols for replicated data in modern storage systems with a high degree of elasticity," *in Document and Text Processing*, Conservatoire National Desarts et Metiers (CNAM), Paris, France, 2016.

[3] B. Kemme and G. Alonso, "A suite of database replication protocols based on group communication," in *Proceedings of the 18th International Conference on Distributed Computing Systems (ICDCS)*, pp. 156–163, Amsterdam, Netherlands, May 1998.

[4] S. K. Wong and M. Y. Siu, "Location spoofing attack detection with pre-installed sensors in mobile devices," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 11, no. 4, pp. 16–30, 2020.

[5] Z. Guan, X. Liu, L. Wu et al., "Cross-lingual multi-keyword rank search with semantic extension over encrypted data," *Information Sciences*, vol. 514, pp. 523–540, 2020.

[6] F. Yan, X. Yang, J. Liu, H. L. Tang, Y.-A. Tan, and Y. Z. Li, "Optimizing the restoration performance of deduplication systems through an energy-saving data layout," *Annals of Telecommunications*, vol. 74, no. 7-8, pp. 461–471, 2019.

[7] Y. Li, S. Yao, K. Yang, Y.-A. Tan, and Q. Zhang, "A high-imperceptibility and histogram-shifting data hiding scheme for JPEG images," *IEEE Access*, vol. 7, pp. 73573–73582, 2019.

[8] Y. Li, X. Zhang, X. Xu, and Y.-A. Tan, "A robust packet-dropout covert channel over wireless networks," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 60–65, 2020.

[9] M. Park, S. Kim, and J. Kim, "Research on note-taking apps with security features," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 11, pp. 63–76, 2020.

[10] L. Zhang, S. Hao, and Q. Zhang, "Recovering SQLite data from fragmented flash pages," *Annals of Telecommunications*, vol. 74, no. 7-8, pp. 451–460, 2019.

[11] Y.-A. Tan, X. Xu, C. Liang, X. Zhang, Q. Zhang, and Y. Li, "An end-to-end covert channel via packet dropout for mobile networks," *International Journal of Distributed Sensor Networks*, vol. 14, no. 5, 2018.

[12] R. Zhu, B. Zhang, J. Mao, Q. Zhang, and Y.-A. Tan, "A methodology for determining the image base of ARM-based industrial control system firmware," *International Journal of Critical Infrastructure Protection*, vol. 16, pp. 26–35, 2017.

[13] W. Wang, Y. Shang, Y. He, Y. Li, and J. Liu, "BotMark: automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors," *Information Sciences*, vol. 511, pp. 284–296, 2020.

[14] W. Wang, X. Wang, D. Feng, J. Liu, Z. Han, and X. Zhang, "Exploring permission-induced risk in android applications

for malicious application detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1869–1882, 2014.

[15] W. Meng, W. Li, and L.-F. Kwok, "EFM: enhancing the performance of signature-based network intrusion detection systems using enhanced filter mechanism," *Computers & Security*, vol. 43, pp. 189–204, 2014.

[16] W. Li, W. Meng, Z. Tan, and Y. Xiang, "Design of multi-view based email classification for IoT systems via semi-supervised learning," *Journal of Network and Computer Applications*, vol. 128, pp. 56–63, 2019.

[17] Y. Yuan, L. Huo, Z. Wang, and D. Hogrefe, "Secure apit localization scheme against sybil attacks in distributed wireless sensor networks," *IEEE Access*, vol. 6, pp. 27629–27636, 2018.

[18] D. Berbecaru, A. Lioy, and C. Cameroni, "Supporting authorize-then-authenticate for wi-fi access based on an electronic identity infrastructure," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 11, pp. 34–54, 2020.

[19] D. M.-E. Francesc, E. A.-I. José, and S. Melissa, "Evaluation of database replication techniques for cloud systems," *Computing and Informatics*, vol. 34, no. 5, pp. 973–995, 2015.

[20] Q. Fan, M. Tan, and Y. Luo, "Security setting and security hardening of computer equipment in LAN," *Information Security and Communication Confidentiality*, vol. 17, no. 7, p. 31, 2008.

[21] B. Dong, "Physical isolation technology for network security," *Computer and Automation*, vol. 10, no. 2, pp. 108–110, 2004.

[22] I. Kholod, A. Shorov, and S. Gorlatch, "Efficient distribution and processing of data for parallelizing data mining in mobile clouds," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 11, pp. 2–17, 2020.

[23] W. Liu and L. Tong, "Design of change capture scheme in integration of heterogeneous databases," *Workflow of Database Synchronization Sending End Computer Application Research*, vol. 23, no. 7, pp. 213–215, 2005.

[24] R. Guerraoui and A. Schiper, "Software-based replication for fault tolerance," *Computer*, vol. 30, no. 4, pp. 68–74, 1997.

[25] H. Anada, "Decentralized multi-authority anonymous authentication for global identities with non-interactive proofs," in *Proceedings of the 2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, vol. 10, no. 4, pp. 23–37, IEEE, Washington, DC, USA, June 2019.

[26] F. Guo, Q. Zhao, X. Li et al., "Detecting adversarial examples via prediction difference for deep neural networks," *Information Sciences*, vol. 501, pp. 182–192, 2019.

[27] Q. Zhang, Y. Zhao, Y. Wang, T. Baker, J. Zhang, and J. Hu, "Towards cross-task universal perturbation against black-box object detectors in autonomous driving," *Computer Networks*, vol. 180, Article ID 107388, 2020.

[28] A. Kumar and A. Segev, "Cost and availability tradeoffs in replicated data concurrency control," *ACM Transactions on Database Systems*, vol. 18, no. 1, pp. 102–131, 1993.

[29] J. B. Lim and A. R. Hurson, "Data duplication and consistency in a mobile, multidatabase enviroment," in *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 50–58, IEEE, Tainan, Taiwan, December 1998.

[30] A. Paul, S. Lihua Wang, D. S. Sharmila, and C. P. Rangan, "Non-transferability in proxy re-encryption revisited," *Journal of Internet Services and Information Security*, vol. 10, no. 3, pp. 1–30, 2020.

[31] M. Nicola and M. Jarke, "Performance modeling of distributed and replicated databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 4, pp. 645–672, 2000.

[32] J. Wu and N. Li, "Design and application of safety isolation software for intelligent hydropower plant," *Hydropower Plant Automation*, vol. 37, no. 4, pp. 67–69, 2016.

[33] H. Yu, Y. Ding, X. Gao, N. Geng, X. Huang, and Y. Lu, "Data security transmission technology of water conservancy project standardization management based on one-way isolation gate," *Zhejiang Water Science and Technology*, vol. 44, no. 5, pp. 48–50, 2016.

[34] D. Filipović, D. Sokač, and R. Picek, "Bidirectional database synchronization to the cloud computing platform," in *Proceedings of the ICISS 2020: 2020 The 3rd International Conference on Information Science and System*, London, UK, March 2020.

[35] S. H. Phatak and B. R. Badrinath, "Conflict resolution and reconciliation in disconnected databases," in *Proceedings of the 10th International Workshop on Database & Expert Systems Applications (DEXA)*, pp. 76–81, IEEE, Florence, Italy, September 1999.

[36] R. K. Cassar, J. Vella, and J. Ellul, "A conflict resolution abstraction layer for eventually consistent databases," in *Proceedings of the 2016 International Conference on Engineering & MIS (ICEMIS)*, pp. 1–5, IEEE, Agadir, Morocco, September 2016.

[37] A. Marotta, D. Cassioli, M. Tornatore, Y. Hirota, Y. Awaji, and B. Mukherjee, "Reliable slicing with isolation in optical metro-aggregation networks," in *Proceedings of the 2020 Optical Fiber Communication Conference and Exhibition*, IEEE, San Diego, CA, USA, March 2020.

[38] J. Gray and A. Reuter, *Transaction Processing: Concepts and Techniques*, Morgan Kaufmann, San Mateo, CA, USA, 1993.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5518220, 11 pages
https://doi.org/10.1155/2021/5518220

Research Article

An Intelligent Detection Method of Personal Privacy Disclosure for Social Networks

Haiyan Kang,[1] **Yanhang Xiao**,[1,2] **and Jie Yin**[1,3]

[1] *Department of Information Security, Beijing Information Science and Technology University, Beijing 100192, China*
[2] *Faculty of Engineering, University of New South Wales, Sydney 2052, Australia*
[3] *School of Computer Science, University of Sydney, Sydney 2006, Australia*

Correspondence should be addressed to Haiyan Kang; kanghaiyan@126.com

Received 5 February 2021; Revised 15 March 2021; Accepted 31 March 2021; Published 24 April 2021

Academic Editor: Hao Peng

With the increase of the number of users in the current social network platform (taking WeChat as an example), personal privacy security issues are important. This paper proposes an intelligent detection method for personal privacy disclosure in social networks. Firstly, we propose and construct the eigenvalue in social platform. Secondly, by calculating the value of user account assets, we can obtain the eigenvalue to calculate the possibility of threat occurrence and the impact of threat. Thirdly, we analyse the situation that the user may leak the privacy information and make a score. Finally, SVM algorithm is used to classify the results, and some suggestions for warning and modification are put forward. Experiments show that this intelligent detection method can effectively analyse the privacy leakage of individual users.

1. Introduction

Today's society is developing rapidly, and with the popularity of smartphones, the amount of private information they generate is increasing. With the occurrence of "PRISM," Facebook user personal information leaked, and other incidents, the issue of private security has begun to attract people's attention.

In recent years, WeChat is one of the most popular apps in China, and as of June 2019, the number of monthly active accounts on WeChat reached 1.13 billion. The huge number of users will contain a large amount of user privacy information. However, most users do not have relevant expertise or neglect information management. Therefore, when using WeChat, they do not pay attention to the protection of private information. After investigation and analysis, disclosure of account passwords, the addition of friend settings, location information, etc. during the chat process will pose a threat to user data if they are leaked and may cause economic loss or even personal harm. Because of the above problems, there have been related studies. Reference [1] conducted a comprehensive evaluation of the apps in the mobile app market but did not consider the risk of social platforms. Reference [2] aimed at Facebook to collect a large amount of user data and analyse them using a questionnaire, but the number of users on WeChat in China far exceeds Facebook. Reference [3] summarized the abnormal account detection scheme based on the four aspects of behaviour characteristics, content, graphs, and unsupervised learning. The detection scheme is rich, but the feature values involved are relatively small, making the data analysis insufficient and not targeting personal social accounts to make appropriate adjustments. Reference [4] fully considered the issue of trajectory privacy leakage and protected it with a prefix tree, but it was not enough to consider all aspects of personal social account security. Reference [5] made a formal description of the malicious use of the address book matching function and made corresponding protection measures, but it was also considered incomplete. In [6], a lot of malicious programs and risk programs on the Android side were analysed in depth, but only the leakage of information through phone calls and text messages was analysed. It did not analyse the social platform and failed to give a more intuitive evaluation system. In [7], a high scientific and rigorous static analysis, dynamic analysis, and network data model were used for multidimensional analysis. However, only 30 apps were tested, and it was concluded that the current application market software leaks user privacy. No risk assessment is performed for each user. In [8], the risk analysis was based on association rules and game theory, but the selected feature values were few and were not targeted at social platforms. Moreover, it is learned that the privacy leak detection systems in the current environment are oriented to companies and enterprises and are not suitable for analysing individual users.

The innovations proposed in this article are as follows: (1) Through the investigation and analysis of WeChat, eight characteristic items in social networks are proposed and constructed, which are account passwords during chats, WeChat wallet consumption records (not friends), and WeChat wallet transfer records (friends), Moments settings of strangers, settings for nearby people, settings for adding friends, Moments settings of friends, and information acquisition of mini-program. The intelligent detection system uses these filtered feature items to calculate the risk value more efficiently and accurately. (2) Use the operations of asset identification, threat identification, and vulnerability analysis to calculate the comprehensive threat value. (3) An intelligent detection method based on SVM (Support Vector Machine) is proposed to divide the data more accurately. (4) After investigation, most of the detection software with similar functions today is oriented to enterprises, and this system is a rare intelligent detection system for individual users on the market.

2. Principles of Intelligent Detection System

The intelligent detection method of personal privacy leakage for social networks proposed in this article is always for users. There is a risk of personal privacy leakage. By obtaining user WeChat settings, asset identification, threat identification, and vulnerability analysis are performed, and the matrix is compared to obtain security. For event risk value, calculate information leakage risk coefficient according to weight. Reference [9–12] pointed out that machine learning has been widely applied in the fields of healthcare, cybersecurity, etc. due to its powerful data mining capabilities, where SVM is one of the most popular machine learning algorithms; therefore use SVM algorithm to divide information leakage risk coefficient and get a final evaluation.

2.1. Risk: Risk Is the Effect of Uncertainty on a Goal. The risks explored in this article refer to the risks of information security breaches, human or natural threats, and the use of vulnerabilities in information systems and their management systems to cause security incidents and their impact on organizations. In the current environment of high information transparency, private information cannot be in a state of zero risks [8].

2.2. Asset Identification

2.2.1. Assets and Their Value. Assets refer to any information or resources that are valuable to the unit. The value of assets does not refer to the economic value of the information system but is closely related to the business work of the organization. Asset value is the importance and sensitivity of assets and the main content of asset identification.

2.2.2. Asset Identification. Asset identification includes two steps: "asset classification" and "asset assignment." This article explores the classification of application software. Based on asset classification, further semiqualitative and semiquantitative analysis of assets is performed; that is, asset valuation is performed, to have a scientific and rational understanding of asset value. Assets are broken down into three security attribute assignments: "confidentiality assignment," "integrity assignment," and "availability assignment."

2.2.3. Confidentiality. It is the feature that prevents the information from being leaked to unauthorized individuals, entities, processes, or makes it useless.

2.2.4. Integrity. It protects the accuracy and completeness of information and processing methods.

2.2.5. Usability. It is a feature that can be accessed and used by authorized entities once they are needed [13].

2.3. Threat Identification

(1) Threat: Potential cause of an accident that may cause damage to assets or units.

(2) Threat identification: Referring to the process of analysing the potential cause of an accident. Threat identification is divided into "threat classification" and "threat assignment" [13].

2.4. Vulnerability Analysis

(1) Vulnerability: Weakness in assets or assets that can be threatened. Compared with threats, threats are the external cause of risk, and vulnerability is the internal cause of risk. The two together form a risk.

(2) Vulnerability identification: Referring to the process of analysing and measuring the weak links of assets that may be threatened to use [13].

2.5. Basic Introduction of SVM Algorithm. SVM refers to support vector machine, which is a common method of discrimination. In the field of machine learning, it is a supervised learning model, which is usually used for pattern recognition, classification, and regression analysis.

The main idea of SVM can be summarized as two points:

(1) It analyses linearly separable cases. For linearly inseparable cases, by using a nonlinear mapping algorithm, a linearly inseparable sample from a low-dimensional input space is transformed into a high-dimensional feature space to make it linearly separable. It is possible to perform a linear analysis of the nonlinear features of the sample using a linear algorithm in the feature space.

(2) It constructs the optimal hyperplane in the feature space based on the structural risk minimization theory, so that the learner is globally optimized, and the expectations in the entire sample space meet a certain upper bound with a certain probability [14].

3. Intelligent Detection Model Design

3.1. Basic Architecture of Intelligent Detection Model. This intelligent detection model is divided into a data source layer, an analysis layer, and a calculation layer, as shown in Figure 1. Among them, after the user source of WeChat data is obtained by the data source layer, eight characteristic values are selected for analysis and calculation; the analysis layer performs asset identification in turn for the characteristic values, and threat calculation and vulnerability analysis, respectively, obtain calculation tables. Asset identification selects three security attributes of asset confidentiality, integrity, and availability, calculates the asset value, and divides the asset value into five levels to obtain a quantitative asset value table. Threat identification is to classify threats into five levels based on the frequency of threats to obtain a table of the frequency of threats. Vulnerability analysis is to calculate the fragility property calculation table by calculating the basic measurement group, time measurement group, and environmental measurement group in turn; at the calculation layer, the three calculation tables in the analysis phase are combined with the security event to compare the two-dimensional matrix table to obtain from each eigenvalue's data the risk value of the security event.

Then the sum of the weight values of each risk value is used to obtain the risk value, and the risk value is brought into the corresponding SVM classifier to obtain the final result.

3.2. Eigenvalue Construction. Based on the investigation and analysis of WeChat, we selected the following conditions as the eigenvalues. The intelligent detection system uses these filtered feature items to calculate the risk value more efficiently and accurately:

3.2.1. Account Password in the Chat Process. The account and password are directly mentioned during the chat. If the chat history is stolen, the account and password information is leaked, and the entire account will be lost, with more illegal acts.

3.2.2. WeChat Wallet Consumption Records (Non-Friends). They require money to communicate with each other without knowing too much about the identity of the other party, have lack of security protection, and may cause economic losses.

3.2.3. WeChat Wallet Transfer Records (Friends). The transfer security between friends is higher than the transfer between non-friends, but if the identity of the friend is impersonated, the identity of the transfer counterparty is unknown, so even the transfers between friends will be at risk.

3.2.4. Setting up a Circle of Strangers. The setting of a circle of strangers is divided into invisible to strangers, ten photos visible to strangers, and unlimited. If the attacker continuously obtains the user circle information for a long time, the stranger can see that the ten photos are not much different from unlimited, which will cause a large amount of information leakage for the user.

3.2.5. Settings for Nearby People. If the nearby people are not closed, the real-time location of the user will be exposed and used by criminals.

3.2.6. Add Friend Settings. The related settings include whether you need to verify when adding as a friend. The way to search for users is divided into WeChat, mobile phone number, and QQ number, in addition to business card. Too many permissions in this regard will increase the possibility of being disturbed by strangers.

(7) Location of Moments: The attacker can further commit a crime based on the obtained positioning information, causing the user's personal safety to be threatened

(8) Mini-Program Information Acquisition: Mini-programs usually obtain user information. If the mini-programs are used by criminals, arbitrating user information will lead to user information leakage.

3.3. Asset Identification. Assets have security attributes such as confidentiality, integrity, and availability, which reflect the characteristics of the asset in different aspects. By quantifying the three security attributes, one can calculate a value that reflects the asset [15].

$$\text{AssetValue} = \text{INT}\left[\log_2\left(2\text{Conf} + 2\text{Int} + 2\text{Avail}\right)\right]. \tag{1}$$

Among them, Conf represents confidentiality assignment; *Int* represents integrity assignment; *Avail* represents availability assignment; INT represents rounding processing and rounding. The three security attributes are divided into 5 levels. The higher the level, the greater the impact on assets. There are 5 levels of corresponding security attributes, and the level of asset value is also divided into 5 levels. The greater the level is, the more important the asset is.

It can be seen from Table 1 that the disclosure of the account password during the chat process will lead to the loss of the entire account information, so its three

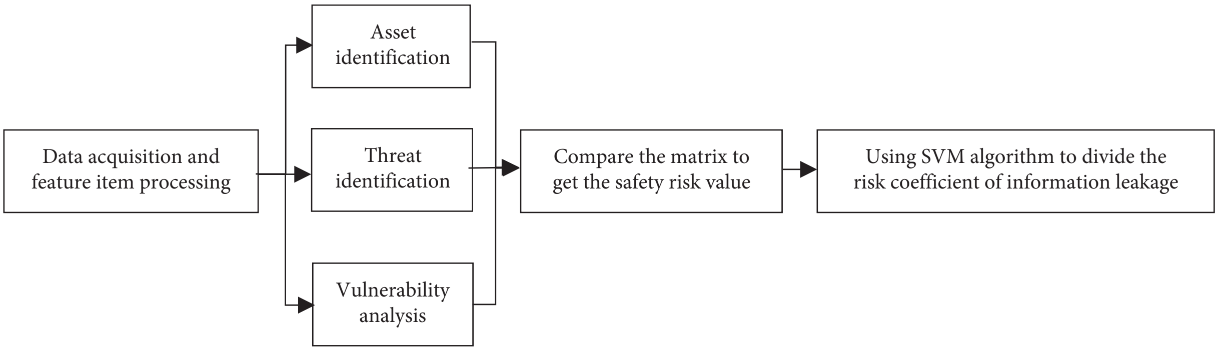

Data acquisition and feature item processing
Asset identification
Threat identification
Vulnerability analysis
Compare the matrix to get the safety risk value
Using SVM algorithm to divide the risk coefficient of information leakage

Figure 1: Schematic diagram of detection system structure.

Table 1: Quantification of asset identification.

Characteristic values	Security attributes			
	Conf	Int	Avail	Asset value
f1: Account password revealed during chat	5	5	5	5
f2: WeChat wallet consumption records (non-friends)	3	3	1	4
f3: WeChat wallet transfer records (friends)	3	1	1	3
f4: Moments permissions settings for strangers	3	3	4	4
f5: Setting nearby people	3	4	4	4
f6: Adding friends	3	4	3	4
f7: Moments location targeting	1	1	3	3
f8: Using mini-program	3	4	3	4

assignments and the calculated asset value are 5, which is the highest. Compared with WeChat wallet transfers between friends, the required protection information and processing methods are more accurate and complete than the WeChat wallet transfers between friends; that is, the integrity assignment is relatively high. For feature items that are likely to come in contact with strangers (Moments permissions settings for strangers, setting nearby people, adding friends, and using mini-program), we have assigned more average values. The location of the Moments is mostly limited to friends, so the value is lower.

3.4. Threat Identification. According to the frequency of threats, the possibility of threats is defined and divided into 5 levels. The higher the level, the higher the probability of threats.

It can be seen from Table 2 that the number of account password disclosures and WeChat wallet consumption records between non-friends during the chat process has a greater impact on each leak, so the interval assignment frequency of threats of different levels is smaller. The remaining eigenvalue assignment intervals are larger or assigned according to the settings in the specific WeChat.

3.5. Vulnerability Analysis. This paper uses the Common Weak Evaluation System (CVSS). The CVSS evaluation system consists of three measurement groups: the basic measurement group, the time measurement group, and the environment measurement group [15].

Basic metric = round_to_1_decimal (10 ∗ access vector ∗ access complexity ∗ authentication ∗ ((confidentiality impact ∗ confidentiality impact weight value) + (consistent impact ∗ consistency impact weight value) + (availability impact ∗ availability impact weight value)))

The values in Table 3 were selected according to Table 4 [16]. Since personal privacy leaks are based on local information, all access vectors are selected locally. WeChat has official protection measures, so the complexity of access is all high. Authentication refers to verifying whether the user has the right to access the system. Authentication is only required for special operations, so all selections are not required. If the account password disclosed in the chat is leaked, it will cause the user to lose all his accounts, so only the confidentiality impact, consistency impact, and availability impact of this feature item are selected all, and the rest are selected all or according to the impact. The confidentiality impact weight value, consistency impact weight value, and availability impact weight value are assigned according to the proportion of each characteristic item affected by the three attributes. Finally, the basic measurement value is calculated.

Time metric = round_to_1_decimal (basic metric ∗ available for use ∗ grade that can be repaired ∗ confidentiality of the report)

The values in Table 5 were selected according to Table 6 [16]. The leakage of the account password during the chat is most likely to be used, so this feature item can be selected for high utilization. The transfer records have low availability, so the selection is not confirmed. The availability of location selection in Moments is theoretically proven to be practical and feasible for the remaining feature items. The level that can be repaired is assigned according to the featured item according to whether it is

Table 2: Frequency of threats.

	Level				
Characteristic values	5	4	3	2	1
f1: Account password revealed during chat	10 times or above	7~9	5~6	3~4	0~2
f2: WeChat wallet consumption records (non-friends)	10 times or above		5~6	3~4	0~2
f3: WeChat wallet transfer records (friends)	21 times or above	16~20	11~15	6~10	0~5
f4: Moments permissions settings for strangers	Allow strangers to see ten Moments				Not allowing strangers to see ten Moments
f5: Setting nearby people	Open				Close
f6: Adding friends	No verification required; can be searched through WeChat, QQ number, mobile phone number; can be added through group chat, QR code, business card	No verification required; can be searched through WeChat, QQ number, mobile phone number; not added through group chat, QR code, business card	Requires verification; can be searched by WeChat, QQ number, mobile phone number; can be added through group chat, QR code, business card	Requires verification; can be searched by WeChat, QQ number, mobile phone number; not added by group chat, QR code, business card	Requires verification; cannot be searched by WeChat, QQ, or mobile phone number; can be added through group chat, QR code, business card
f7: Moments location targeting	10 times or above	7~9	5~6	3~4	0~2
f8: Using mini-program	10 and above	7~9	5~6	3~4	0~2

Table 3: Calculation table of basic metrics.

Characteristic values	Related parameters									
	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
f1	0.7	0.8	1.0	1.0	0.5	1.0	0.25	1.0	0.25	5.6
f2	0.7	0.8	1.0	0.7	0.333	0.7	0.333	0.7	0.333	3.92
f3	0.7	0.8	1.0	0.7	0.333	0.7	0.333	0.7	0.333	3.92
f4	0.7	0.8	1.0	0.7	0.333	0.7	0.333	1.0	0.333	2.3
f5	0.7	0.8	1.0	0.7	0.333	0.7	0.333	1.0	0.333	2.3
f6	0.7	0.8	1.0	0.7	0.333	0.7	0.333	1.0	0.333	2.3
f7	0.7	0.8	1.0	0.7	0.25	0.7	0.25	0.7	0.5	3.92
f8	0.7	0.8	1.0	0.7	0.333	0.7	0.333	1.0	0.333	2.3

a1: access vector, a2: access complexity, a3: authentication, a4: confidentiality impact, a5: confidentiality impact weight value, a6: consistency impact, a7: consistency influence weight value, a8: usability impact, a9: usability impact weight value, and a10: basic measure.

easy to recover after the leak. The transfer records and Moments positioning are better than other feature items. Therefore, high and theoretical are selected, and the rest are selected unconfirmed. The confidentiality of the report has been uniformly selected.

Environmental metric value = round_to_1_decimal ((time metric score + ((10-time metric score) ∗ incidental loss impact)) ∗ target distribution).

The values in Table 7 were selected according to Table 8 [16]. The impact of the loss of transfer records and location of the Moments is small, so the impact of incidental losses is selected as medium and low, and the rest are selected as high. The target distribution is assigned according to the distribution of the feature items. The account password and password in the chat are selected to be high, and the rest are selected to be low or medium. Calculate environmental metrics.

Vulnerability value calculation formula [16]: V = INT {(environmental measurement value ∗ 5)/10 + 0.5}, so vulnerability value is shown in Table 9.

Table 4: Basic metric value assignment table.

Parameters	Values corresponding to different degrees			
Access vector	Local: 0.7		Remote: 1.0	
Access complexity	Height: 0.8		Low: 1.0	
Authentication	Requires: 0.6		Not required: 1.0	
Confidentiality impact	None: 0	Section: 0.7	All: 1.0	
Confidentiality impact weight value	Normal: 0.333	Confidentiality: 0.5	Consistency: 0.25	Availability: 0.25
Consistency impact	None: 0	Section: 0.7	All: 1.0	
Consistency influence weight value	Normal: 0.333	Confidentiality: 0.25	Consistency: 0.5	Availability: 0.25
Usability impact	None: 0	Section: 0.7	All: 1.0	
Usability impact weight value	Normal: 0.333	Confidentiality: 0.25	Consistency: 0.25	Availability: 0.5

Table 5: Time measurement value calculation table.

Characteristic values	Related parameters			
	Exploitability	Repairable level	Confidentiality of the report	Time metric
f1: Account password revealed during chat	1.00	0.87	0.90	4.4
f2: WeChat wallet consumption records (non-friends)	0.85	1.00	0.90	3.0
f3: WeChat wallet transfer records (friends)	0.85	1.00	0.90	3.0
f4: Moments permissions settings for strangers	0.95	0.87	0.90	1.7
f5: Setting nearby people	0.95	0.87	0.90	1.7
f6: Adding friends	0.95	0.87	0.90	1.7
f7: Moments location targeting	0.90	0.90	0.90	2.9
f8: Using mini-program	0.95	0.87	0.90	1.7

Table 6: Time metric value assignment table.

Parameters	Values corresponding to different degrees			
Exploitability	Unconfirmed: 0.85	Proved by theory: 0.90	Practical: 0.95	High: 1.00
Repairable level	Unconfirmed: 0.87	Proved by theory: 0.90	Practical: 0.95	High: 1.00
Confidentiality of the report	Unconfirmed: 0.90	Unverified: 0.95	Confirmed: 1.00	

Table 7: Calculation table of environmental measures.

Characteristic values	Related parameters		
	Collateral loss effects	Target distribution	Environmental metric value
f1: Account password revealed during chat	0.5	1.00	7.2
f2: WeChat wallet consumption records (non-friends)	0.3	0.25	1.3
f3: WeChat wallet transfer records (friends)	0.3	0.25	1.3
f4: Moments permissions settings for strangers	0.5	0.75	4.4
f5: Setting nearby people	0.5	0.75	4.4
f6: Adding friends	0.5	0.25	1.5
f7: Moments location targeting	0.1	0.25	0.9
f8: Using mini-program	0.5	0.75	4.4

Note: round_to_1_decimal refers to rounding to one decimal place.

Table 8: Assignment table of environmental metrics.

	Values corresponding to different degrees			
Parameters	No	Low	Middle	High
Collateral loss effects	0	0.1	0.3	0.5
Target distribution	0	0.25	0.75	1.0

3.6. Risk Calculation: The Calculation of Risk Is as Follows. After completing asset identification, threat identification, and vulnerability identification, an appropriate model can be used to calculate the risk value of a security event caused by the vulnerability using threats. This article adopts the risk calculation model in Chinese National Standard GB/ *T* 20984 "Information Security Technology, Information Security Risk Assessment Specification".

The formula is expressed as risk value = R (A, T, V) = R (L (T, V), F (A, V)). Among them, R is the calculation function

Table 9: Vulnerability value calculation table.

Characteristic values	Vulnerability value
f1: Account password revealed during chat	4
f2: WeChat wallet consumption records (non-friends)	1
f3: WeChat wallet transfer records (friends)	1
f4: Moments permissions settings for strangers	3
f5: Setting nearby people	3
f6: Adding friends	1
f7: Moments location targeting	1
f8: Using mini-program	3

of security risk, A is the value of the asset, T is the threat, V is the vulnerability, L is the possibility of threatening the use of the vulnerability of the asset to cause a security event, and F is the loss caused by the security event.

In the specific calculation of risk, there are three key calculation links.

3.6.1. Calculate the Probability of a Security Incident. According to the frequency and vulnerability of threats, calculate the probability that a threat will cause a security event using vulnerability, that is, the probability of a security event = L (frequency of threats, the severity of vulnerability) = L (T, V).

This system uses a two-dimensional matrix algorithm to calculate the probability of a security event, as shown in Table 10 [15].

3.6.2. Calculate Losses Caused by Security Incidents. According to the value of the asset and the severity of the vulnerability, calculate the loss caused by the security event once it occurs, that is, the loss caused by the security event = F (asset value, severity of vulnerability) = F (A, V).

This system uses a two-dimensional matrix method to calculate the loss of security events, as shown in Table 11 [15].

3.6.3. Calculating the Value at Risk. According to the calculated probability of the security event and the loss caused by the security event, calculate the risk value, that is, risk value = R (the probability of the security event, the loss caused by the security event) = R (L (T, V), F (A, V)).

The system uses the two-dimensional code matrix method to calculate the risk value of security events, as shown in Table 12 [15].

4. Sum Based on Weights

The risk value of each data security event is obtained from Table 12, and each risk value is multiplied by the weight value of Table 13 to obtain the final risk value.

$$T = INT\left\{\left[\frac{\sum_{i=1}^{N}\left(t_i\right)}{2N + 0.5}\right]\right\}. \tag{2}$$

Table 10: Two-dimensional matrix of security event probability calculations.

Severity of vulnerability	Frequency of threats				
	1	2	3	4	5
1	2	4	7	9	12
2	4	6	9	13	16
3	6	9	13	17	21
4	8	11	14	21	23
5	9	13	18	23	25

Table 11: Two-dimensional matrix table of security event loss calculation.

Severity of vulnerability	Asset value				
	1	2	3	4	5
1	2	3	6	9	11
2	3	6	9	12	16
3	5	8	12	16	20
4	7	10	13	19	22
5	9	13	18	23	25

Table 12: Two-dimensional matrix table for calculating the risk value of security events.

Loss caused by the security event	Probability of the security event				
	1~5	6~10	11~15	16~20	21~25
1~5	3	6	9	14	13
6~10	6	11	17	21	21
11~15	11	18	22	30	30
16~20	15	21	31	40	55
21~25	22	35	55	85	100

Table 13: Comprehensive threat calculation table.

Characteristic values	Related parameters			
	T_s	T_i	t	Weights
f1: Account password revealed during chat	4	5	9	0.225
f2: WeChat wallet consumption records (non-friends)	3	3	6	0.15
f3: WeChat wallet transfer records (friends)	2	2	4	0.1
f4: Moments permissions settings for strangers	2	2	4	0.1
f5: Setting nearby people	2	3	5	0.125
f6: Adding friends	2	2	4	0.1
f7: Moments location targeting	2	1	3	0.075
f8: Using mini-program	2	3	5	0.125

All comprehensive threat value calculation formulas [15–17].

The calculation formula for the comprehensive calculation value of a single threat to an information asset: $t = T_s + T_i$.

Among them, t is a single threat comprehensive value, T_s is a threat source value, defined as a value between 1 and 5, and T_i is an impact degree value and is also defined as a value between 1 and 5.

4.1. SVM Algorithm Application. The intelligent detection system uses the SVM algorithm to divide the comprehensive threat value (as shown in Figure 2) and further divides the risk level of the user account more accurately.

The specific process is as follows.

Based on the comprehensive threats mentioned above, it is worth calculating the risk value. The obtained risk values are divided into two categories. The scores of 1 to 40 are low-risk areas, and the scores of 40 to 100 are high-risk areas. Among them, 1 to 20 in the low-risk areas are defined as safe, 21 to 40 are defined as basic safety, 41 to 59 in the high-risk areas are defined as higher risks, and 60 to 100 parts are expressed as high risks.

The feature quantities of two types of risk values defined as safety and basic safety are recorded in the initial feature vector set 1. The feature quantities defined as two types of risk values of higher-risk and high-risk are recorded in the initial feature vector set 2.

Normalize the feature data items to remove the extreme data. Convert the processed two types of data formats into an input format acceptable to the SVM classifier (class vector Y, feature vector Xi)

The corresponding classifier 1 is trained using data defined as safe and basic safety as training samples, and the corresponding classifier 2 is trained using data defined as safe and basic safety as training samples.

Set the SVM parameters and use the K-fold cross-validation algorithm to find the optimal parameters. Perform asset identification, vulnerability analysis, and threat identification from the characteristic values read by the user. After risk calculation, determine the low-risk area or high-risk area based on the score and enter the corresponding risk area as a test sample. The SVM classification model performs classification judgment. Substitute the results obtained by the SVM into the Naive Bayes formula to obtain the security risk probability, and send feedback of the final results to the user.

4.2. SVM Algorithm Training. The SVM calculation process is shown in Figure 3. The format of the training data and test data is:

<label> <index (1)>: <value1> <index (2)>: <value2> ...

For example: 0 1:1 2:1 3:1 4:1 5:2 6:2 7:2 8:2.

Among them:

<label> is the category identifier of the training data set, set to 0 and 1,0 for security, and 1 for basic security.

<index> refers to 8 feature quantities of the input algorithm, which are integers.

<value> is the value of the feature code for each item and is an integer.

SVM_train implements training on training samples to obtain SVM models.

SVM classification is a prediction of the classification result of the data set according to the trained model.

Use SVM_train to train the input training data set to obtain the SVM model file. The SVM algorithm maps each input training sample, that is, an n-dimensional vector into a high-dimensional space, forming multiple scattered points, and passing the aggregation of points. The region simulates the classification hyperplane and continuously uses the newly input training sample data to make corrections, generates template files, and records the classification features.

In this paper, the K-fold cross-validation method is used to obtain the optimal parameters by verifying the accuracy of the results. The main purpose of the verification algorithm is to divide the data set A into a training set B and a test set C. When the sample size is small, the data set A can be randomly divided into k packets, and one of the packets is used as the test set at a time. The remaining k-1 packets are trained as a training set. The cross-validation method is used to prevent overfitting caused by the model being too complicated [18]. By constantly transforming two important parameters of the SVM: the penalty factor C and the kernel function parameter g, the optimal parameters $C = 2048$ and $g = 0.0078$ are determined.

4.3. SVM Algorithm Processing

SVM classifier 1:

Input $x = \{a_1, a_2, \cdots, a_m\} y\{y_0, y_1\}$, x represents the feature value set of each sample in the test sample, y represents the categories are 0 and 1, which represent safety and basic safety respectively;

Output The user's security risk probability is less than or equal to 50% as safe, and greater than 50% as basic safety.

Step 1: Normalize the feature data

Step 2: Convert the processed feature data into an input format acceptable by the classifier (feature vector x, category vector y) to obtain training samples

Step 3: Set the SVM type to 0-SVM and the kernel function type to radial basis function (RBF)

Step 4: Set the penalty factor C and kernel function parameter G

Step 5: Set the K value of the K-fold cross-validation algorithm

Step 6: Use the SMO algorithm to find the support vector

Step 7: Build a hyperplane model from training samples

Step 8: Enter the test samples for classification, and get the classification result y

Step 9: Calculate the $P(a_i|y)$ to obtain the conditional probability ratio of each feature attribute in the result classification y

Step 10: Calculate $p(y)$ to get the probability of category y appearing

Step 11: Calculate $p(a_i)$ to get the probability of each characteristic attribute

Step 12: Substitute the formula

$$P(y|x) = \frac{P(x|y)P(y)}{P(a_i)P(a_2)\cdots P(a_m)} \cdot a. \tag{3}$$

Step 13: return $P(y|x)$

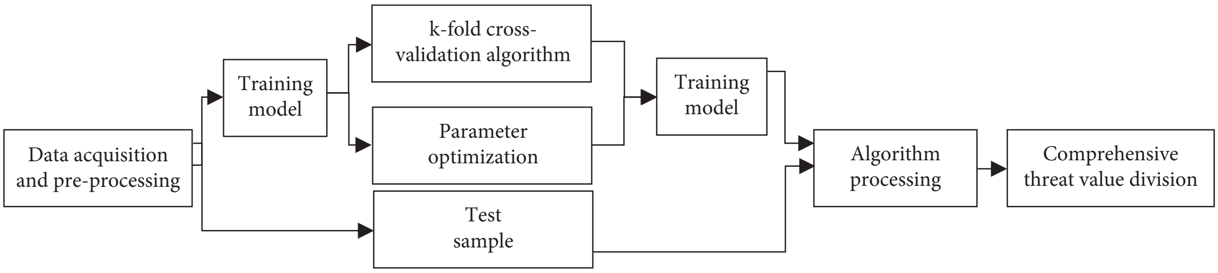

Data acquisition and pre-processing
Training model
k-fold cross-validation algorithm
Parameter optimization
Test sample
Training model
Algorithm processing
Comprehensive threat value division

Figure 2: SVM algorithm application diagram.

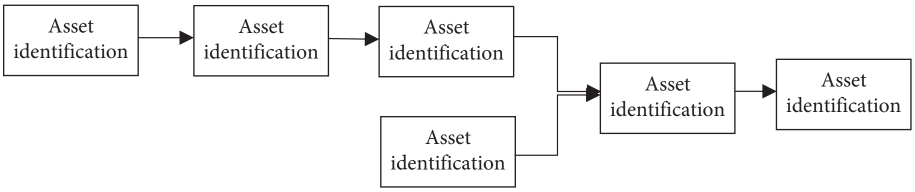


Figure 3: SVM calculation process.

SVM classifier 2:

Input $x = \{a_1, a_2, \cdots, a_m\} y\{y_0, y_1\}$, x is the feature value set of each sample in the test sample, y is the category is 0 and 1, which means high-risk and higher-risk respectively

Output The user's security risk probability is less than or equal to 50% as a high-risk and greater than 50% as higher-risk.

SVM classifier 2 process is the same as SVM classifier 1.

5. Experiment and Analysis of Intelligent Detection System

5.1. Environmental Configuration and Data Acquisition. This test system is designed to run on the Android platform. During the test phase, Android Studio is used to simulate the Android platform for various tests.

Due to the inconvenience of directly obtaining the personal privacy data of the user's WeChat, a questionnaire was used at this stage to collect the WeChat usage of 149 users as a training sample for the SVM classifier. The specific content of the questionnaire is shown as Appendix in Supplementary Materials (available here).

5.2. Functional Test. User test assignment table is shown in Table 14. For the functional test of this system, we first obtained the WeChat related records of a user for testing, as sample 1. The user has been tested and calculated a comprehensive threat value of 26. After obtaining the comprehensive threat value, the data format of this sample is converted into an input format acceptable to the classifier. Based on the user's comprehensive threat value of 26, the sample should be determined. Enter SVM classifier 1. The input format is 0 1 : 3 2 : 1 3 : 2 4 : 1 5 : 1 6 : 1 7 : 4 8 : 2, and processing of sample 1 is complete.

After removing the extreme data from the remaining samples, the above steps are processed and sent to the corresponding SVM classifier. The training samples are used to build a hyperplane model. When the system intelligently detects the risk leakage probability, it will automatically obtain the feature quantity, calculate the comprehensive threat value after calculation, and send it to the corresponding SVM classifier to obtain the final security risk probability.

According to this method, we processed the results of 149 user questionnaires and calculated the number of scores for each segment. The results are shown in the following Table 15.

From Table 15, we can see that a total of 89 users are in the low-risk area and 34 users are in the risk area, of which 26 users are in the security zone. This shows that the security awareness education has been effective, and people have realized that personal privacy is important, but there are also many users in high-risk areas, indicating that there is still a need to continue with efforts to expand coverage and increase everyone's security awareness.

5.3. Performance Testing. Obtain user WeChat related information through a questionnaire. As a sample, test the personal privacy leak detection value of a user's social network, and give a warning or suggestion to get the percentage of people at each risk level, and then get the current data of whether people know and implement the degree of privacy protection in place, which aspects are of importance to people, and which aspects are ignored by people, and provide directions for the promotion of privacy protection awareness in the future. The findings are shown in Figure 4.

At the same time, we counted the number of occurrences of high threats for each feature item (that is, the number of times assigned 4 or 5).

In Figure 5, we can see that most people have a certain awareness of self-privacy protection, but many people ignore the function of "people nearby" and allow strangers to view the private information that may be leaked in Moments. A system that can protect the privacy of the user's privacy is

Table 14: User test assignment table.

Characteristic values	User test related assignments
f1: Account password revealed during chat (6 times)	3
f2: WeChat wallet consumption records (non-friends) (3 times)	1
f3: WeChat wallet transfer records (friends) (10 times)	2
f4: Moments permissions settings for strangers (close)	1
f5: Setting nearby people (close)	1
f6: Requires verification; cannot be searched by WeChat, QQ, or mobile phone number; can be added through group chat, QR code, business card	1
f7: Moments location targeting (8 times)	4
f8: Using mini-program (2)	2

Table 15: Number distribution of each risk area.

Level	Number
Safety	60
Basic safety	29
High-risk	26
Higher-risk	34

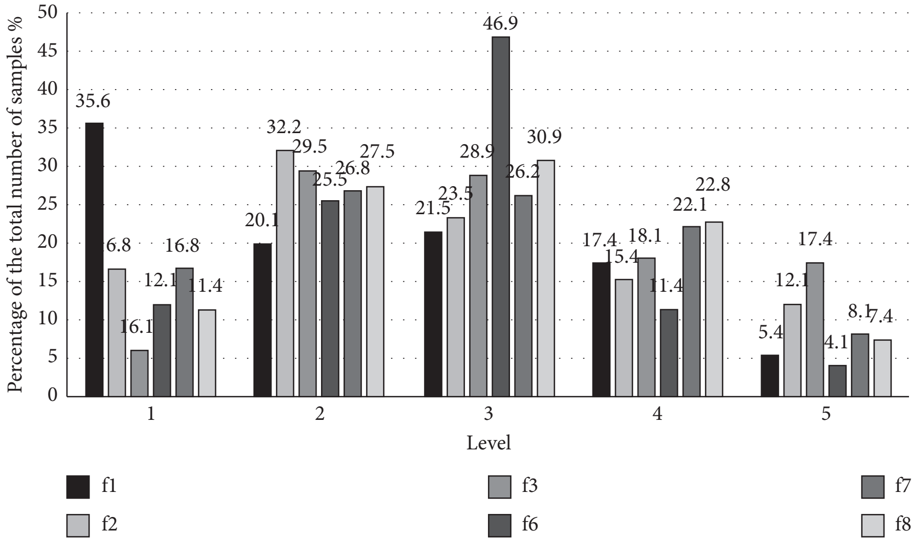


Figure 4: Statistics of the percentage of occurrences of each feature.

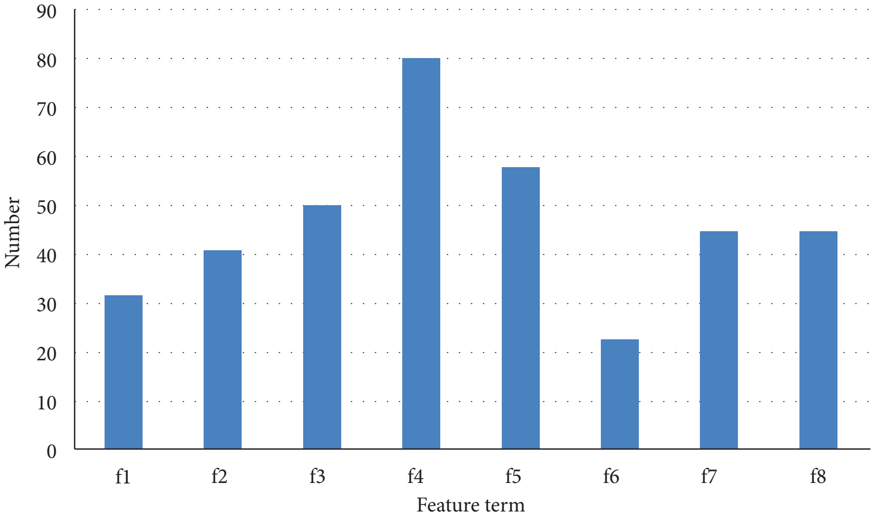


Figure 5: Statistics of occurrence times of high threats for each feature.

essential. Through this intelligent detection system for personal privacy leaks for social networks, users can clearly understand their negligence in the process of using WeChat and correct them to prevent problems before they occur.

6. Concluding Remarks

The system proposed in this article is based on reading multiple characteristic values of personal WeChat and establishing a model based on three aspects of asset identification, threat identification, and vulnerability analysis. According to the risk calculation models and methods in the national standards of information security risk assessment standards, the dimension matrix table calculates the possibility of security events, the loss of security events, and the risk value of security events, determines the risk level according to the magnitude of the risk, evaluates the personal privacy leakage of the user's online social software, gives a score, and informs the user about source of risk.

This article only mentions the scoring function in the system and the function of displaying the risk source of personal privacy leakage. In the future, more functions will be added to improve the entire system, which will also make the judgment more accurate and create a more accurate situation for the individual users, creating safe environment to use social networks.

Data Availability

Due to the inconvenience of directly obtaining the personal privacy data of the user's WeChat, a questionnaire was used at this stage to collect the WeChat usage of 149 users as a training sample for the SVM classifier. The specific content of the questionnaire is given in Supplementary Materials.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by humanities and social sciences research project of the Ministry of Education (No. 20YJAZH046) and Higher Education Research Projects (No. 2020GJZD02), "Practical training plan" (Entrepreneurship) for cross training of high-level talents.

Supplementary Materials

The detailed information of the questionnaire is given in Appendix I. According to the number, it is divided into five levels: 5 is the highest, it is gradually decreased in order, and 1 is the lowest. (*Supplementary Materials*)

References

[1] G. Dini, F. Martinelli, I. Matteucci, M. Petrocchi, A. Saracino, and D. Sgandurra, "Risk analysis of Android applications: a user-centric solution," *Future Generation Computer Systems*, vol. 80, pp. 505–518, C, 2018.

[2] P. Van Schaik, J. Jansen, J. Onibokun, J. Camp, and P. Kusev, "Security and privacy in online social networking: Risk perceptions and precautionary behaviour," *Computers In Human Behavior*, vol. 78, pp. 283–297, 2018.

[3] Y.-Q. Q. Zhang, S.-Q. Lv, and D. Fan, "Anomaly detection in online social networks," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 38, no. 10, pp. 2011–2027, 2015.

[4] Z. F. Huo, X.-F. Meng, and Y. Huang, "PrivateCheckIn: trajectory privacy-preserving for check-in services in MSNS," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 36, no. 4, pp. 716–726, 2013.

[5] Y. Y. Cheng, L.-Y. B. Ying, S.-B. R. Jiao, P.-R. G. Su, and D.-G. Feng, "Research on user privacy leakage in mobile social messaging applications," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 37, no. 1, pp. 87–100, 2014.

[6] K. Wang, *Research and Application of Android Platform Application Risk Detection*, Beijing University of Posts and telecommunications, Beijing, China, 2012.

[7] T. X. Li, Y.-X. Q. Xing, A.-Q. J. Hu, and Y.-J. Wang, "Research on multi-dimensional privacy leakage evaluation model for mobile terminals," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 41, no. 9, pp. 2134–2147, 2018.

[8] Q. Kuang, *Risk Assessment Based on Personal Privacy Disclosure [D]*, Guizhou University, Guizhou, China, 2016.

[9] H. Chen, A. B. Ünal, M. Akgün et al., "Privacy-preserving SVM on outsourced genomic data via secure multi-party computation," in *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, pp. 61–69, New Orleans, LA, USA, October 2020.

[10] S. Park, J. Byun, J. Lee, J. H. Cheon, and J. Lee, "HE-friendly algorithm for privacy-preserving SVM training," *IEEE Access*, vol. 8, pp. 57414–57425, 2020.

[11] J. Liang, Z. Qin, J. Ni et al., "Efficient and privacy-preserving outsourced SVM classification in public cloud," in *Proceedings of the ICC 2019-2019 IEEE international conference on communications (ICC)*, pp. 1–6, IEEE, Shanghai, China, May 2019.

[12] Y. Xu, C. Wu, K. Zheng, Xu Wang, X. Niu, and T. Lu, "Computing Adaptive Feature Weights with PSO to Improve Android Malware Detection," *Security and Communication Networks*, vol. 2017, Article ID 3284080, 14 pages, 2017.

[13] H. Xiang, O. Fu, and B. Zhan, *Information Security Measurement and Risk Assessment*, Electronic Industry Press, Beijing, China, 2014.

[14] CSDN: (Support Vector Machine) https://blog.csdn.net/android_ruben/article/details/78308868?utm_source=debugrun&utm_medium=referral.

[15] M. Shang and M. Chen, "A calculation method of risk value of information assets," *Network security technology and application*, vol. 5, pp. 163-164, 2014.

[16] Y. Wu, X. Li, and K. Lu, *Information Security Risk Assessment*, pp. 70–83, Tsinghua University Press, Beijing, China, 2007.

[17] 2007 Information Security Technology— Risk Assessment Specification for Information Security: General Administration of quality Supervision, Inspection, and Quarantine of the People's Republic of China and China National Standardization Administration.

[18] N. Jia, S. Fu, and M. Xu, *Privacy-Preserving Blockchain-Based Nonlinear SVM Classifier Training for Social Networks*, Security and communication networks, vol. 2020, Article ID 8872853, 10 pages, 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5530023, 11 pages
https://doi.org/10.1155/2021/5530023

Research Article

Adaptive Routing Strategy Based on Improved Double Q-Learning for Satellite Internet of Things

Jian Zhou, [1,2,3] Xiaotian Gong, [1,2] Lijuan Sun, [1,2] Yong Xie, [1,2] and Xiaoyong Yan [1,2]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[3] *Ministry of Education Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, Nanjing 211189, China*

Correspondence should be addressed to Lijuan Sun; lucifinil919@126.com

Received 17 February 2021; Revised 31 March 2021; Accepted 8 April 2021; Published 21 April 2021

Academic Editor: Hao Peng

Satellite Internet of Things (S-IoT), which integrates satellite networks with IoT, is a new mobile Internet to provide services for social networks. However, affected by the dynamic changes of topology structure and node status, the efficient and secure forwarding of data packets in S-IoT is challenging. In view of the abovementioned problem, this paper proposes an adaptive routing strategy based on improved double Q-learning for S-IoT. First, the whole S-IoT is regarded as a reinforcement learning environment, and satellite nodes and ground nodes in S-IoT are both regarded as intelligent agents. Each node in the S-IoT maintains two Q tables, which are used for selecting the forwarding node and for evaluating the forwarding value, respectively. In addition, the next hop node of data packets is determined depending on the mixed Q value. Second, in order to optimize the Q value, this paper makes improvements on the mixed Q value, the reward value, and the discount factor, respectively, based on the congestion degree, the hop count, and the node status. Finally, we perform extensive simulations to evaluate the performance of this adaptive routing strategy in terms of delivery rate, average delay, and overhead ratio. Evaluation results demonstrate that the proposed strategy can achieve more efficient and secure routing in the highly dynamic environment compared with the state-of-the-art strategies.

1. Introduction

Satellite Internet of Things (S-IoT) is an integration of satellite networks [1] and IoT [2]. S-IoT not only strengthens communication by using relay satellites, but also forms a new mobile Internet [3] oriented toward the integrated satellite-terrestrial information network architecture [4]. S-IoT has the advantages of wide coverage and high robustness and can provide ubiquitous services for social networks, so it has attracted considerable attention [5, 6].

As the fundamental of communication protocol for S-IoT, the routing strategy is responsible for data packet forwarding and is of great significance to the communication security [7–9]. Compared with terrestrial networks, S-IoT has the following characteristics. (1) The high-speed movement of satellite nodes and the frequent failure of sensor nodes result in the dynamic topology structure, causing unstable end-to-end path in S-IoT. (2) The complex space environment and the uneven amount of terrestrial access data lead to the dynamic node status. (3) Due to the limited energy of satellite nodes and sensor nodes, energy efficiency must be taken into account in the routing strategy to reduce the overhead ratio. (4) The large number of nodes and the heterogeneity among nodes impose specific requirements upon the efficiency and security during data packet forwarding. With all these characteristics in mind, we conclude that the routing strategy for the terrestrial network is not applicable to S-IoT.

We study S-IoT as a delay tolerant network (DTN) without intersatellite links. Since S-IoT involves heavy data service workloads, which are generally not in requirement of very low delay, the store-carry-forward mechanism of DTN, which can cope with the dynamic topology structure in S-IoT, is used by the satellite nodes to forward data packets. In recent years, DTN has attracted extensive attention of researchers, and many routing strategies for DTN have been proposed. Existing routing strategies usually can be classified into three categories including the flood-based, the utility-based, and the mobility model-based routing strategies. To be specific, we select several representative routing strategies falling within individual categories and discuss them in brief. The Epidemic routing strategy proposed by Vahdat et al. [10] is one of the flood-based routing strategies, in which one node forwards data packets to every node it encounters. This virus-like propagation mode results in excessive overhead ratio. In order to improve Epidemic, Spyropoulos et al. [11] proposed the Spray-and-Wait routing strategy. The process of data packet forwarding consists of two phases, i.e., spraying and waiting. The data packets are diffused into some copies in the spraying phase. These copies are directly forwarded to the destination node in the waiting phase. This strategy reduces the overhead ratio and achieves the similar performance in transmission with Epidemic. As one of the utility-based routing strategies, the Prophet routing strategy was proposed by Lindgren and Doria [12]. In this strategy, each data packet makes a copy to the node only in case of a high encountering probability, for the purpose of reducing the amount of replication and the overhead ratio. Sharma et al. [13] proposed the machine learning routing strategy based on Prophet (MLRSP). This strategy takes the speed and location of nodes into account and uses the decision tree as well as the neural network to calculate the encountering probability, achieving better performance than Prophet dose. Among mobile model-based routing strategies, the contact graph routing strategy, which was proposed by Araniti et al. [14], is capable of reducing the average delay by selecting the next hop node based on the minimum hop count and the shortest path.

However, the aforementioned routing strategies for DTN cannot quickly adapt to the frequent changes of node status, and the copies in these strategies bring in the challenge of communication security in S-IoTs. To tackle these challenges, we propose employing reinforcement learning on the basis of our previous work [15] to develop a novel adaptive routing strategy for S-IoT. Since the reinforcement learning can obtain optimal results even if the system environment changes frequently, it has been successfully applied in a variety of fields such as industrial manufacturing, analogue simulation, game competition, and scheduling management. As a reinforcement learning algorithm, double Q-learning [16] chooses the next better hop node by self-learning to cope with the dynamic changes of topology structure and node status in S-IoT while satisfying the communication security requirement.

In view of the dynamic topology structure and the dynamic node status, this paper presents an adaptive routing strategy based on improved double Q-learning for S-IoT. The main contributions of this paper are as follows:

(1) We apply the reinforcement learning to the S-IoT routing strategy to make it adapt to the dynamic changes of topology structure and node status in S-IoT.

(2) We improve the forwarding performance by means of optimizing the mixed Q value, the reward value, and the discount factor, respectively, based on the congestion degree, the hop count, and the node status.

(3) We establish the S-IoT model, which consists of a ground layer, a LEO layer, and a MEO layer, to perform simulation experiments. Simulation results demonstrate that the proposed strategy improves the performance of data packet forwarding, in terms of delivery rate, average delay, and overhead ratio, compared with the state-of-the-art strategies.

The rest of this paper is organized as follows. Section 2 introduces the related work. The description of the proposed adaptive routing strategy is detailed in Section 3. Section 4 discusses how to improve the Q value in double Q-learning. Simulation results and the associated analysis are given in Section 5. Section 6 concludes this paper.

2. Related Work

2.1. Routing Strategy for Satellite Networks. Satellite networks not only provide remote transmission capability for IoT, but also provide cloud computing capability [17–19], so satellite networks have direct impact on the overall performance of S-IoTs. The routing strategy for satellite networks is responsible for data transmission and distribution between satellites under various security requirements. In recent years, routing strategies for satellite networks are extensively studied in the literature.

Some researchers paid attention to the dynamic changes of the topology structure caused by the high-speed movement of satellites. Gounder et al. [20] proposed a routing strategy based on snapshot sequence. Mauger and Rosenburg [21] proposed a routing strategy based on virtual nodes. Hashimoto and Sarikaya [22] proposed a routing strategy based on division of the coverage area. Wang et al. [23] proposed a routing strategy based on position and velocity of the nodes. Though simple and easy-to-implement for routing computation, they often need high storage capacities. Some researchers focused on the limited energy caused by the lack of continuous energy supply. Ekici et al. [24] proposed a routing strategy for saving the energy cost. Yang et al. [25] proposed an energy-efficient routing strategy. Marchese and Patrone [26] proposed an energy-aware routing strategy. These strategies can reduce energy consumption, but they induce high computational burden. Some other researchers are concerned about the poor QoS caused by long distance between nodes and unstable links. Mao et al. [27] proposed a routing strategy separating the collection and calculation of QoS. Huang et al. [28] proposed a routing strategy under guaranteed delay constraints. Xu et al. [29] proposed a routing strategy based on asynchronous transfer mode. However, these strategies focused on improving the QoS of voice and multimedia services and failed to consider data services.

It is worth emphasizing that all the routing strategies mentioned above use the intersatellite links. In existing low earth orbit (LEO) and medium earth orbit (MEO) constellation systems, only Iridinm is equipped with intersatellite links due to the high cost and complex system. Other constellation systems, such as Ocbcomm, Globalstar, and O3b, have no intersatellite links [30]. For this reason, it is more reasonable to construct the S-IoT based on the constellation systems without intersatellite links, which is the main purpose of this work.

2.2. Routing Strategy Based on Reinforcement Learning. In recent years, reinforcement learning has attracted widespread attention. As a classic reinforcement learning algorithm, Q-learning [31] obtains the sample data sequence (state, action, and reward value) through interacting with the environment and uses the state-action function value (Q value) to find the best action for the current state. In addition, Q-learning ensures communication security by the self-learning mechanism. Q-learning has been applied in many fields. Deng et al. [32] applied Q-learning to the task allocation of edge computing. Zhao et al. [33] applied Q-learning to the DoS attack of many core systems.

In the routing field, Elwhishi et al. [34] proposed a Q-learning routing strategy for DTN. In this strategy, nodes collaborate with each other and make forwarding decisions based on connections. However, node status is not considered in this work. Plate and Wakayama [35] proposed a Q-learning routing strategy based on kinematics and sweep features. This strategy can adapt to the constantly dynamic changes of the topology structure caused by the node mobility and energy consumption. Rolla and Curado [36] proposed an enhanced Q-learning routing strategy for DTN. This strategy calculates the reward value based on the distance between nodes such that more data packets in densely populated areas can be delivered. Wu et al. [37] proposed an adaptive Q-learning routing strategy based on anycast (ARSA). This strategy focuses on anycast communication from a node to multiple destination nodes, while considering the encountering probability and the relative speed of nodes.

However, the abovementioned Q-learning routing strategies suffer from the overestimation issue in certain cases. The reason is that Q-learning algorithm uses the same Q value for the action selection with the action evaluation and uses the maximum Q value as an approximation to the maximum expected Q value. Q-learning tends to produce a positive estimate deviation, since the overestimated Q value has the higher chance to be selected.

The double Q-learning algorithm, which was proposed by Hasselt [16], uses two Q values to separate action selection and action evaluation. Double Q-learning has been applied in many fields. Zhang et al. [38] applied double Q-learning to the speed control of autonomous vehicle. Vimal et al. [39] applied double Q-learning to improve energy efficiency of cognitive radio networks. Zhang et al. [40] applied double Q-learning to the energy-saving scheduling of edge computing. So far, double Q-learning has been rarely used in the routing field.

The kernel idea of the double Q-learning algorithm is that the action is selected based on the greedy algorithm in each step and the two Q values are adaptively updated with the changes of environment. One Q value selects the action, and the other one evaluates the selected action. The selection is decoupled from the evaluation for reducing the positive deviation. Furthermore, double Q-learning algorithm has a similar computational efficiency compared with Q-learning algorithm. Therefore, we use double Q-learning to avoid selecting neighbor nodes with overestimation.

3. Proposed Strategy

The whole S-IoT is regarded as a reinforcement learning environment in this paper. Satellites in S-IoT are regarded as satellite nodes, whereas sensors and data centers are regarded as ground nodes. For each individual node, all other nodes it can encounter constitute its neighbor node set. In particular, ground nodes generate and receive data packets, and satellite nodes use the store-carry-forward mechanism to forward data packets.

Both satellite nodes and ground nodes are considered as intelligent agents. Each node learns the network environment of the whole S-IoT through interacting with other nodes it encounters. Furthermore, all nodes are included to form the state set of reinforcement learning. A ground node or satellite node selects one node from its neighbor node set to forward data packets. This procedure is considered as an action selection of reinforcement learning. In this manner, the neighbor node set for this node can be regarded as the possible action set. The state transitions are equivalent to forwarding data packets from one node to a neighbor node.

In the proposed strategy, each node is assigned with two Q tables (Q^A and Q^B) to store the Q value of the action which is referred to as selecting a neighbor node to forward data packets to the destination node. Each node only updates its own two Q tables and shares its local information only with its neighbor nodes. The two Q values stored in the corresponding Q tables are used to determine and evaluate the greedy strategy, respectively. More importantly, the two Q values are decoupled to address the issue of overestimation which may cause the local optima of routing. The two Q values change with the topology structure and node status such that the proposed strategy can be adaptive to the highly dynamic environment.

Initially, a new node has no knowledge of the whole S-IoT environment with two empty Q tables. When this node encounters other nodes, it records the identities of other nodes and initializes the corresponding Q values to 0 in two Q tables.

The selection of neighbor node for each data packet would update the two Q values. Each data packet has a destination node. When the data packet reaches its destination node, the Q values of all nodes on this forwarding path will be updated by a rewarding procedure. In the proposed strategy, the two Q values are intensively learned from two different experience sets of the S-IoT. The mixed Q value depending on the two Q values decides which node should be selected to forward data packets.

Figure 1 illustrates the general routing process of a specific node. If the destination node is in its neighbor node set, this node forwards data packets to the destination node to complete data transmission. Otherwise, depending on the largest mixed Q value, this node selects a neighbor node to forward data packets. It stores and carries these data packets until it encounters the selected node. Such operations are repeated until the simulation is terminated. The greedy algorithm ensures the largest cumulative future rewards. Take node c, for example, the node selected from its neighbor node set, can be expressed as

$$x^* = \arg\max_{x\in N_c} \hat{Q}_c(d,x), \tag{1}$$

where N_c is the neighbor node set of node c and node x is one of the neighbor nodes in N_c. $\hat{Q}_c(d,x)$ is the mixed Q value of the node selection action, and node d is the destination node of the data packets. The improved method for calculating $\hat{Q}_c(d,x)$ will be given in the next section. If two nodes have identical mixed Q value, we select one of them at random.

As the learning task is assigned to each node, the learning process is accordingly the updating process of Q tables. If the topology of node c changes, the Q values in Q_c^A will be updated. If the status of node c changes, the Q values in Q_c^B will be updated. In this sense, Q_c^A and Q_c^Brepresent an experience set of topology change and an experience set of status change, respectively. Q_c^A and Q_c^B learn from each other. The updates of Q_c^A and Q_c^B are given by

$$Q_c^A(d,x) = (1-\alpha)Q_c^A(d,x) + \alpha\left(R_c(d,x) + \gamma_c(d,x)Q_x^B(d,y^*)\right), \tag{2}$$

$$Q_c^B(d,x) = (1-\alpha)Q_c^B(d,x) + \alpha\left(R_c(d,x) + \gamma_c(d,x)Q_x^A(d,z^*)\right), \tag{3}$$

where N_x is the neighbor node set of node x and α is the learning rate manipulating the updating speed of Q values. $R_c(d,x)$ is the instant reward value (R value) and $\gamma_c(d,x)$ is the discount factor of the node selection action. y^* and z^* are the nodes with the largest Q value in Q_x^B and Q_x^A, respectively. The improved method for calculating $R_c(d,x)$ and $\gamma_c(d,x)$ will be given in the next section.

4. Improvement of Q Value

4.1. Mixed Q Value Based on the Congestion Degree. The next hop node of data packets is determined according to the mixed Q value. Because network congestion has an important impact on routing, we consider the congestion degree to give the corresponding weights of two Q values to calculate the mixed Q value.

Take node c for example; if node c selects neighbor node x to forward data packets, the mixed Q value is calculated by

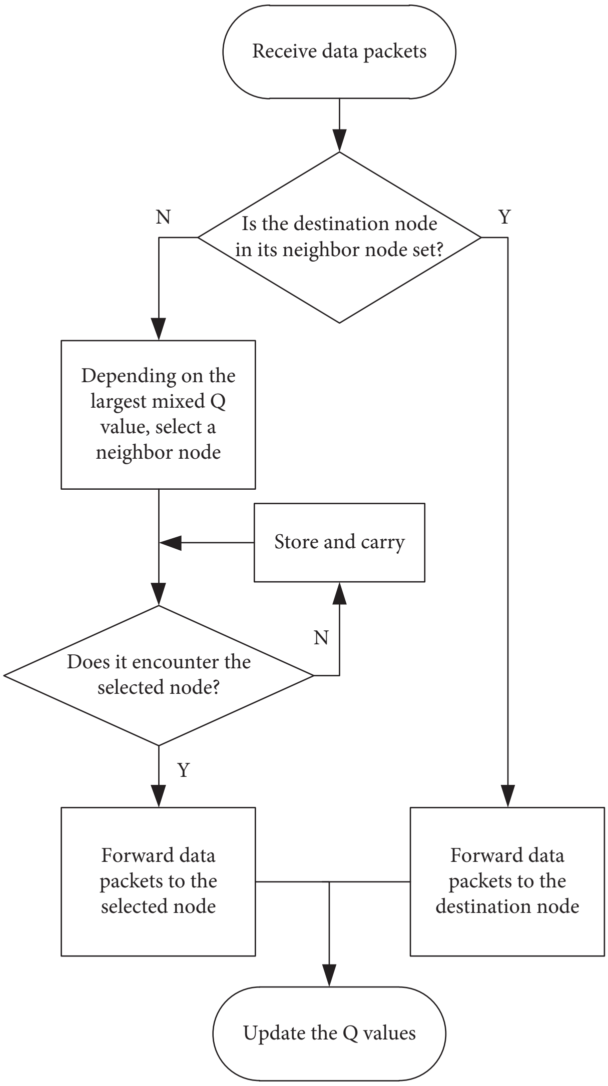

Receive data packets
N
Is the destination node in its neighbor node set?
Y
Depending on the largest mixed Q value, select a neighbor node
Store and carry
N
Does it encounter the selected node?
Y
Forward data packets to the selected node
Forward data packets to the destination node
Update the Q values

Figure 1: Routing process of each node.

$$\hat{Q}_c(d,x) = \beta(x)Q_c^A(d,x) + (1-\beta(x))Q_c^B(d,x), \tag{4}$$

where node d is the destination node of the data packets and $Q_c^A(d,x)$ and $Q_c^B(d,x)$ are the Q values provided by Q^A and Q^B, respectively, indicting the Q values of the action in which node c selects node x to forward data packets. $\beta(x)$ is the congestion factor of node x, and it is calculated by

$$\beta(x) = \begin{cases} 0.6, & \text{if } 0 < \; = con\_d(x) < 0.5, \\ 0.3, & \text{if } 0.5 < \; = con\_d(x) < 0.75, \\ 0.1, & \text{if } 0.75 < \; = con\_d(x)) < 1. \end{cases} \tag{5}$$

In particular, the smaller $con\_d(x)$ value is, the larger $\beta(x)$ value is, so that the influence of topology change is greater. Under the reverse situation, the influence of status change is greater. $con\_d(x)$ can be calculated by

$$con\_d(x) = \frac{\sum_{y\in N_x} S(y)/B_y}{C(x)}, \tag{6}$$

where $S(y)$ is the size of all data packets currently in the buffer of neighbor node y and B_y is the buffer size of neighbor node y. In addition, N_x is the neighbor node set of node x, and $C(x)$ is the number of neighbor nodes of node x.

4.2. Reward Value Based on the Hop Count. An important component in the Q value updating rule (refer to equations (2) and (3)) is the calculation of R value defining the instant reward value after forwarding data packets. R value reflects the advantages and disadvantages of one-time forwarding. Limited by the energy capacity of the S-IoT, the hop count is taken into account in the calculation of reward value to control energy consumption and to reduce the overhead ratio.

Take node c, for example; if node c has forwarded the data packets to neighbor node x, the reward value for the node selection action can be calculated by

$$R_c(d,x) = \begin{cases} 0, & \text{otherwise}, \\ e^{-\left(w_1h_1+w_2h_2+\ldots+w_ih_i+\ldots+w_kh_k\right)}, & \text{if } c == d, \end{cases} \tag{7}$$

where node d is the destination node of the data packets, $h_1, h_2, \ldots, h_i, \ldots, h_k$ are the hop counts on different satellite orbits, and $w_1, w_2, \ldots, w_i, \ldots, w_k$ are the weights of different satellite orbits satisfying $\sum_{i=1}^{k} w_i = 1$. A higher satellite orbit height stands for a greater amount of energy consumption for data transmission between the ground node and the satellite node. Hence, we set a relatively higher w_i value for a satellite node with a higher height orbit. As a result, the reward value for forwarding data packets to a satellite with a higher height orbit is lower.

4.3. Discount Factor Based on the Node Status. The discount factor is a multiplicative coefficient for the sum of subsequent reward values, which affects the possibility of reselecting a previously selected neighbor node to forward data packets. In order to adapt to the node status, the distance, direction, and buffer occupancy are considered in the calculation of discount factor.

Take node c, for example; if node c has forwarded the data packets to neighbor node x, the discount factor for the node selection action is calculated by

$$\gamma_c(d,x) = \gamma \times Dir\_F(d,x) \times Dis\_F(d,x) \times Buf\_F(d,x), \tag{8}$$

where node d is the destination node of the data packets and γ is the setting value subject to $0<\gamma<1$. $Dir\_F(d,x)$, $Dis\_F(d,x)$, and $Buf\_F(d,x)$ denote the direction factor, the distance factor, and the buffer factor, respectively. The larger these factors are, the larger the discount factor is and accordingly the larger the updated Q value is. As such, the possibility of reusing this node to forward data packets in the next time will be larger.

The direction factor is calculated by

$$Dir\_F(d,x) = 1 - \frac{\theta(d,x)}{180}, \tag{9}$$

where $\theta(x,d)$ stands for the angle between neighbor node x and destination node d. The smaller $\theta(x,d)$ value is, the larger $Dir\_F(d,x)$ value is.

The distance factor is calculated by

$$Dis\_F(d,x) = 1 - \frac{D(d,x)}{D_{\max}}, \tag{10}$$

where $D(x,d)$ is the distance from node x to destination node d and $D_{\max}$ is the maximum distance between the nodes in the network. The smaller $D(x,d)$ value is, the larger $Dis\_F(d,x)$ value is.

The buffer factor is calculated by

$$Buf\_F(d,x) = 1 - \frac{S(x)}{B_x}, \tag{11}$$

where $S(x)$ is the size of all data packets currently in the buffer of neighbor node x and B_x is the buffer size of neighbor node x. The smaller $S(x)$ value is, the larger $Buf\_F(d,x)$ value is.

5. Simulation Analysis

5.1. Simulation Environment. We use the ONE simulator to analyze and evaluate the proposed routing strategy. The S-IoT model in simulation experiments is shown in Figure 2. The ground layer is composed of 110 ground nodes, which are uniformly distributed over the Earth's surface. The LEO layer consists of 48 satellite nodes as the Globalstar constellation system. The MEO layer consists of 24 satellite nodes as the GPS constellation system. Table 1 lists the node parameters in each layer. Ground nodes generate and receive data packets, and both the source node and the destination node are randomly generated among ground nodes. Since we assume no intersatellite links in this S-IoT model, data packets cannot be forwarded between any two satellite nodes moving through their orbits periodically.

The network environment parameters in simulation experiments are shown in Table 2. Regarding the double Q-learning procedure, the learning rate is set to 0.8, and γ in the discount factor is set to 0.9. The weights of hop count on LEO and MEO satellite orbits are set to 0.3 and 0.7, respectively. The delivery rate, average delay, and overhead ratio are used to evaluate the routing strategies at different data packet generation intervals with different failure probabilities.

5.2. Simulation Results. We compare the proposed adaptive routing strategy based on improved double Q-learning for S-IoT (ARSIDQL) with the adaptive routing strategy based on original double Q-learning (ARSDQL), the adaptive routing strategy based on original Q-learning (ARSQL), the Spray-and-Wait routing strategy [11], MLRSP [13], and ARSA [37] in terms of delivery rate, average delay, and overhead ratio with different failure probabilities.

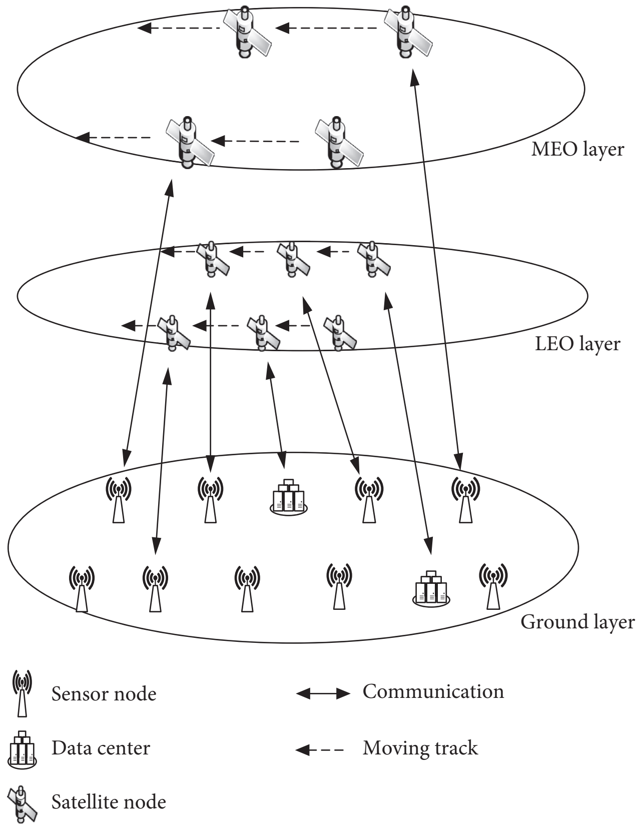


Figure 2: S-IoT model.

Table 1: Node parameters.

Layer	LEO layer	MEO layer	Ground layer
Constellation	Globalstar	GPS	Distributed evenly
Orbit numbers	8	6	/
Node numbers	48	24	110
Height	189 Km	20200 Km	0 Km

Table 2: Network environment parameters.

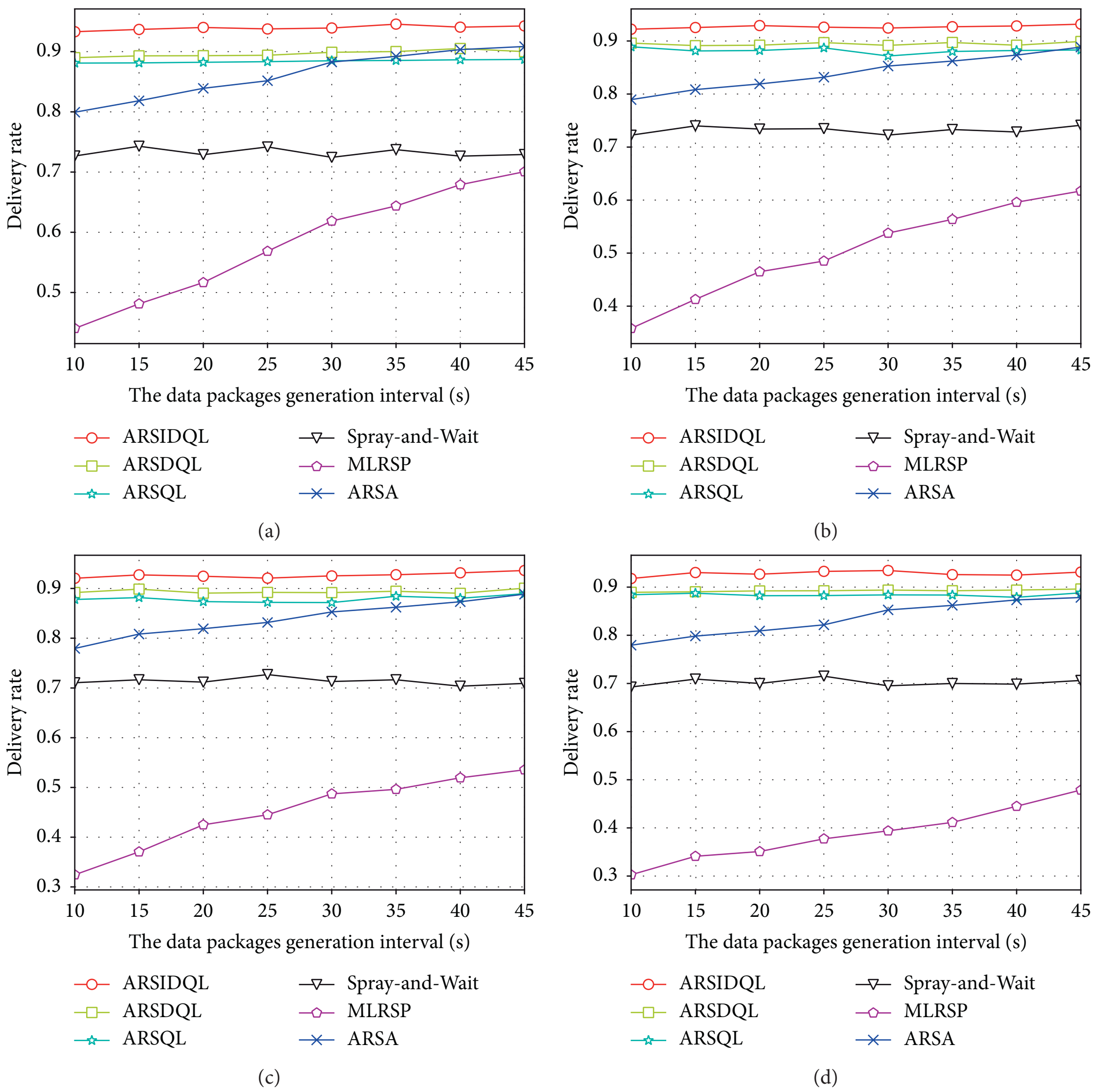
Parameters	Values
Buffer size	35 Mb
Transmission speed	250 Kb
Data packet generation intervals	10–50 s
Data packet size	500 Kb–1 Mb
Data packet TTL	3600 s

5.2.1. Delivery Rate. Figure 3 shows the comparison of delivery rates achieved by all routing strategies at different data packet generation intervals with different failure probabilities. On the whole, MLRSP achieves the lowest delivery rate, since MLRSP calculates the encountering probability of each node and copies data packets only to the node with the largest encountering probability. However, MLRSP fails to take into account the data packet loss caused by the high buffer occupancy of nodes. The delivery rate of Spray-and-Wait is higher than that of MLRSP by taking the advantage of flood. To be specific, the data packets are diffused into several copies to increase the probability of data packets arriving at the destination node. The delivery rates of ARSA and ARSQL are higher than that of Spray-and-Wait; since the Q-learning algorithm is self-learning and self-adaptive, ARSA and ARSQL can explore a suitable path in a highly dynamic environment. However, the encountering probabilities of nodes in S-IoT are fixed. ARSA considers the encountering probability, resulting in lower delivery rate than ARSQL. The delivery rate of ARSDQL is higher than that of ARSQL. The reason is that ARSDQL decouples data packet forwarding from the Q value evaluation of this forwarding, and the node used for forwarding is determined depending on the mixed Q value without positive deviation. Built upon ARSDQL, ARSIDQL incorporates the congestion degree and node status. Hence, data packets are more likely to arrive at the destination node before arriving at the end of their TTLs, so ARSIDQL achieves the highest delivery rate.

With the increase of the data packet generation interval, the delivery rate of MLRSP improves significantly. Since there are a large number of data packets in the network at low generation interval, the buffer size of each node is limited, causing many data packet losses. The delivery rate of Spray-and-Wait remains unchanged, since Spray-and-Wait limits the number of data packet replicas to reduce the buffer occupancy rate and further the number of data packet losses. The delivery rates of ARSA, ARSQL, and ARSDQL are relatively stable, because they can find the best action in the current state depending on the Q value through interacting with the environment. The delivery rates of ARSIDQL are relatively stable and high at low generation interval. This strategy can adapt to the buffer occupancy and forward data packets to nodes with low buffer occupancy rates to reduce the number of data packet losses and to achieve good performance.

With the increase of the failure probability, the delivery rate of MLRSP decreases. Since MLRSP forwards data packets depending on the encountering probability even if node failures have taken place, MLRSP cannot adapt to the changes of topology structure. The delivery rate of Spray-and-Wait decreases slightly. Because the data packets are diffused into some copies, the delivery rate can be guaranteed with insignificant degradation. The delivery rates of ARSA, ARSQL, ARSDQL, and ARSIDQL are relatively stable and high even with high failure probabilities owing to their abilities of self-learning. Since the Q value of forwarding data packets to the failed node would be smaller, these strategies can avoid forwarding data packets to the failed node and thus can adapt to the dynamic topology structure.

5.2.2. Average Delay. Figure 4 shows the comparison of average delays of routing strategies at different data packet generation intervals with different failure probabilities. On the whole, the average delay of Spray-and-Wait is the highest, due to the fact that in this strategy each node can only move and cannot forward data packets until it encounters the destination node in the waiting phase. The



(a)

(b)

(c)

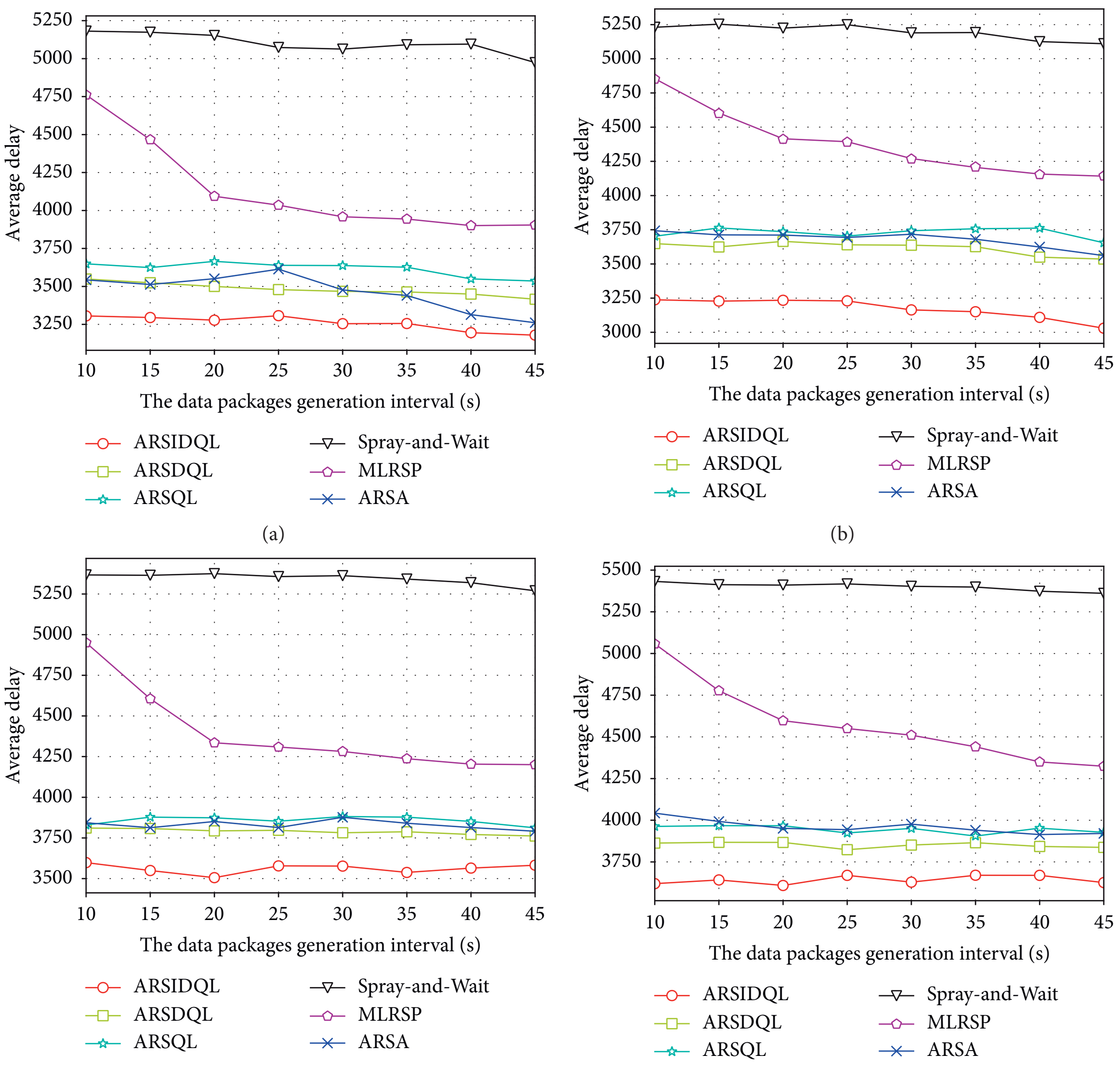
(d)

Figure 3: Delivery rates with different failure probabilities. (a) 0% failure probability. (b) 10% failure probability. (c) 20% failure probability. (d) 30% failure probability.

average delay of MLRSP is also high, since MLRSP only takes into account the encountering probability when each node forwards data packets. However, MLRSP cannot find an appropriate path as the encountering probability cannot reflect the node status. ARSQL can learn by itself to find the next hop node with a relatively low average delay. The average delay of ARSA is lower than that of ARSQL, since ARSA considers the relative speed of nodes. The average delay of ARSDQL is low, since ARSDQL solves the overestimation problem through two Q values and can find the global optimal path to reduce the average delay. Built upon ARSDQL, ARSIDQL can adapt to the congestion degree and hop count to achieve the lowest average delay.

With the increase of the data packet generation interval, the total number of data packets in S-IoT decreases such that the waiting time in the buffer and the average delay of Spray-and-Wait can be reduced. The average delay of MLRSP is reduced to a greater extent. However, the large number of data packets and copies made by MLRSP in S-IoT at low packet generation interval would lead to node congestion and long waiting times in the buffer. The average delays by suing ARSA, ARSQL, ARSDQL, and ARSIDQL decrease slightly with low failure probabilities, because the total number of data packets in S-IoT decreases with the increase of the data packet generation interval. In the cases of high failure probabilities, the average delays remain stable, since these strategies have found a suitable path at low packet generation interval. In addition, the high failure probability leads to fewer nodes in the network. The change of generation interval no longer affects the average delay.

With the increase of the failure probability, the average delays of Spray-and-Wait and MLRSP get worsen accordingly, due to the fact that these strategies cannot make adjustments to failed nodes in a timely fashion. The average delays of ARSA, ARSQL, ARSDQL, and ARSIDQL also degrade slightly. The good thing is that, because the update



(a)

(b)

(c)

(d)

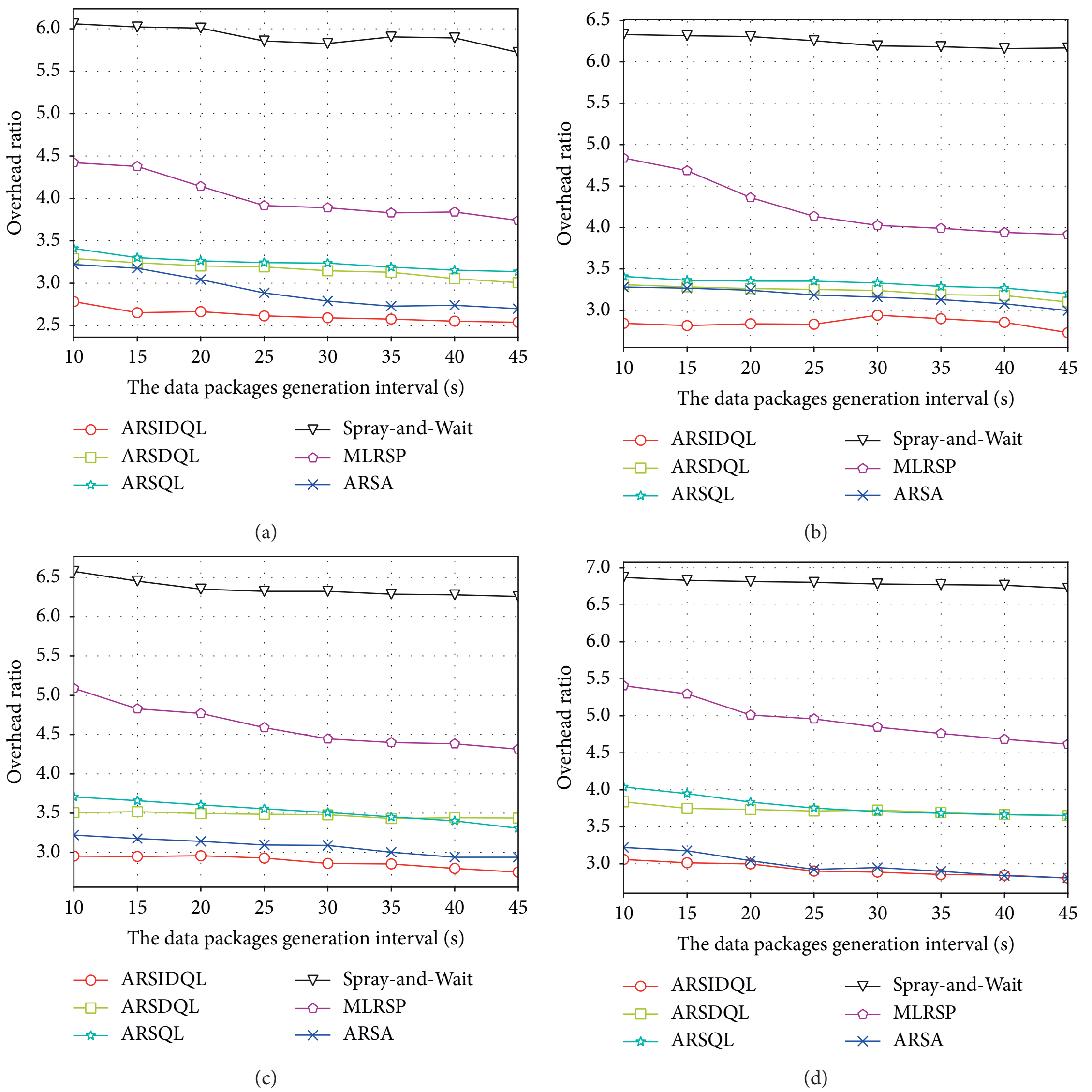
Figure 4: Average delays with different failure probabilities. (a) 0% failure probability. (b) 10% failure probability. (c) 20% failure probability. (d) 30% failure probability.

of the Q value reflects the changes of topology structure, the routes by using these strategies can bypass failed nodes and the degradation of average delay is not significant.

5.2.3. Overhead Ratio. Figure 5 shows the comparison of overhead ratios of various routing strategies at different data packet generation intervals with different failure probabilities. The overhead ratio depending on the forwarding time reflects the energy efficiency. On the whole, the overhead ratio of Spray-and-Wait is the highest. As a flood-based routing strategy, Spray-and-Wait increases the forwarding time in case of a large number of copies of data packets in the network. Compared with Spray-and-Wait, MLRSP achieves a lower overhead ratio, since MLRSP copies data packets only to the node with the largest encountering probability to restrict the forwarding time. ARSQL and ARSDQL, which are not flood-based routing strategies, result in less forwarding time due to fewer data packets in the network. The overhead ratio of ARSA is lower than that of ARSDQL. The reason is that ARSA reduces the forwarding time since it considers multiple destination nodes as the same virtual destination. Built upon ARSDQL, ARSIDQL takes the hop count and node status into consideration, thus achieving the lowest overhead ratio.

With the increase of the data packet generation interval, the overhead ratios of all strategies decrease slightly. As the total number of data packets in S-IoT decreases as data packet generation interval increases, the forwarding time is reduced. As a consequence, lower energy consumption and overhead ratio are achieved.

With the increase of the failure probability, the overhead ratios of Spray-and-Wait and MLRSP increase. The reason is that, under the circumstance of node failures, Spray-and-Wait retransmits data packets in order to maintain a fixed number of copies, whereas MLRSP still forwards data



(a)

(b)

(c)

(d)

Figure 5: Overhead ratios with different failure probabilities. (a) 0% failure probability. (b) 10% failure probability. (c) 20% failure probability. (d) 30% failure probability.

packets depending on the encountering probability. The overhead ratios of ARSA, ARSQL, ARSDQL, and ARSIDQL increase slightly. These strategies are capable of bypassing failed nodes. The bypassing procedure would inevitably lead to the increase of forwarding time, energy consumption, and overhead ratio.

In summary, compared with ARSDQL, ARSIDQL can improve the delivery rate, average delay, and overhead ratio by taking into account the congestion degree, hop count, and node status in the S-IoT model. Also, compared with ARSQL and ARSA, ARSIDQL can find the best next hop node of data packets due to the decoupling of selection and evaluation. Compared with traditional routing strategies, such as the flood-based routing strategy and the utility-based routing strategy, ARSIDQL can significantly improve the delivery rate, average delay, and overhead ratio with the integration of reinforcement learning.

6. Conclusions

S-IoT is a new mobile Internet to provide services for social networks. The routing strategy determines the communication performance of S-IoT. Traditional routing strategies cannot cope with frequent changes of topology structure and node status and cannot meet the requirement of communication security in S-IoTs. This paper proposes an adaptive routing strategy based on improved double Q-learning for S-IoT. The proposed strategy selects the next hop node of data packets relying on the mixed Q value. Moreover, in order to optimize the Q value, this paper makes improvements on the mixed Q value, the reward value, and the discount factor, respectively, based on the congestion degree, the hop count, and the node status. Simulation experiments show that the proposed strategy not only can operate efficiently and securely in complex environments but also can increase the delivery ratio and reduce the average delay and overhead ratio. Considering the large sizes of the two Q tables due to the increasing number of nodes in S-IoT, future work can be directed toward replacing the two Q tables with two neural networks.

Data Availability

The simulated evaluation data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61972210, 61873131, 61872191, and 61803212) and the 1311 Talent Program of Nanjing University of Posts and Telecommunications.

References

[1] M. Liu, N. Qu, and J. Tang, "Signal estimation in cognitive satellite networks for satellite-based industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2062–2071, 2020.

[2] T. Wang, Y. Mei, X. Liu, J. Wang, H.-N. Dai, and Z. Wang, "Edge-based auditing method for data security in resource-constrained internet of things," *Journal of Systems Architecture*, vol. 114, p. 101971, 2021.

[3] X. Liu, M. S. Obaidat, C. Lin, T. Wang, and A. Liu, "Movement-based solutions to energy limitation in wireless sensor networks: state of the art and future trends," *IEEE Network*, vol. 35, no. 2, pp. 188–193, 2021.

[4] G. X. Zhang, X. Jie, and C. Z. Qu, "Development status and challenges of IoT for LEO satellites," *Journal of IoT*, vol. 1, no. 3, pp. 6–9, 2017.

[5] Z. Zhang, W. Zhang, and F. H. Tseng, "Satellite mobile edge computing: improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Network*, vol. 33, no. 1, pp. 70–76, 2018.

[6] F. Wang, D. Jiang, S. Qi, C. Qiao, and L. Shi, "A dynamic resource scheduling scheme in edge computing satellite networks," *Mobile Networks and Applications*, 2020.

[7] A. B. Gabis and M. Koudil, ""NoC routing protocols–objective-based classification," *Journal of System Architecture*, vol. 66, pp. 14–32, 2016.

[8] W. Zhang, G. Han, Y. Feng, and J. Lloret, "IRPL: an energy efficient routing protocol for wireless sensor networks," *Journal of Systems Architecture*, vol. 75, pp. 35–49, 2017.

[9] Q. Li, A. Liu, T. Wang, M. Xie, and N. N. Xiong, "Pipeline slot based fast rerouting scheme for delay optimization in duty cycle based M2M communications," *Peer-to-Peer Networking and Applications*, vol. 12, no. 6, pp. 1673–1704, 2019.

[10] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," *Technical report*, Duke University, Durham, NC, USA, 2000.

[11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking*, Philadelphia, PA, USA, August 2005.

[12] A. Lindgren and A. Doria, "Probabilistic routing in intermittently connected networks," in *Proceedings of International Workshop on Service Assurance with Partial and Intermittent Resources*, Fortaleza, Brazil, August 2004.

[13] D. K. Sharma, S. K. Dhurandher, I. Woungang, R. K. Srivastava, A. Mohananey, and J. J. P. C. Rodrigues, "A machine learning-based protocol for efficient routing in opportunistic networks," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2207–2213, 2018.

[14] G. Araniti, N. Bezirgiannidis, E. Birrane et al., "Contact graph routing in DTN space networks: overview, enhancements and performance," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 38–46, 2015.

[15] X. T. Gong, L. J. Sun, J. Zhou et al., "Adaptive routing strategy based on improved Q-learning for satellite internet of things," in *Proceedings of International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, Nanjing, China, December 2020.

[16] H. V. Hasselt, "Double Q-learning," in *Proceedings of Advances in Neural Information Processing Systems*, Vancouver, BC, USA, December 2010.

[17] J. Sun, Y. Zhang, Z. Wu et al., "An efficient and scalable framework for processing remotely sensed big data in cloud computing environments," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4294–4308, 2019.

[18] Z. Wu, J. Sun, Y. Zhang et al., "Scheduling-guided automatic processing of massive hyperspectral image classification on cloud computing architectures," *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[19] J. Zhou, J. Sun, M. Zhang, and Y. Ma, "Dependable scheduling for real-time workflows on cyber-physical cloud systems," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[20] V. V. Gounder, R. Prakash, and H. Abu-Amara, "Routing in LEO-based satellite networks," in *Proceedings of IEEE Emerging Technologies Symposium, Wireless Communications and Systems*, Richardson, TX, USA, April 1999.

[21] R. Mauger and C. Rosenberg, "QoS guarantees for multimedia services on a TDMA-based satellite network," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 56–65, 1997.

[22] Y. Hashimoto and B. Sarikaya, "Design of ip-based routing in a LEO satellite network," in *Proceedings of International Workshop on Satellite-Based Information Servies*, Dallas, TX, USA, 1998.

[23] S. Wang, C. Fan, C. Deng, W. Gu, Q. Sun, and F. Yang, "A-GR: a novel geographical routing protocol for AANETs," *Journal of Systems Architecture*, vol. 59, no. 10, pp. 931–937, 2013.

[24] E. Ekici, I. F. Akyildiz, and M. D. Bender, "A distributed routing algorithm for datagram traffic in LEO satellite networks," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 137–147, 2001.

[25] Y. Yang, M. Xu, D. Wang, and Y. Wang, "Towards energy-efficient routing in satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3869–3886, 2016.

[26] M. Marchese and F. Patrone, "Energy-aware routing algorithm for DTN-Nanosatellite networks," in *Proceedings of IEEE Global Communications Conference*, Abu Dhabi, UAE, December 2018.

[27] T. Mao, B. Zhou, and Z. Xu, "A multi-QoS optimization routing for LEO/MEO satellite IP networks," *Journal of Multiple Sclerosis*, vol. 9, no. 4, Article ID 576, 2014.

[28] Q. Huang, B. S. Yeo, and P. Y. Kong, "A routing algorithm to provide end-to-end delay guarantee in low earth orbit satellite networks," in *Proceedings of IEEE Vehicular Technology Conference*, Los Angeles, CA, USA, September 2004.

[29] H. Xu, F. Huang, and S. Wu, "A distributed QoS routing based on ant algorithm for LEO satellite network," *Journal of Electronics (China)*, vol. 24, no. 6, pp. 765–771, 2007.

[30] Y. Z. Qin, S. G. Shu, and W. Ye, "Distributed data storage and transmission technology of the space internet of things," *Journal of IoT*, vol. 2, no. 4, pp. 26–34, 2018.

[31] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[32] X. H. Deng, J. Li, and E. L. Liu, "Task allocation algorithm and optimization model on edge collaboration," *Journal of System Architecture*, vol. 110, pp. 1–14, 2020.

[33] Y. M. Zhao, X. H. Wang, and Y. T. Jiang, "On hardware-trojan-assisted power budgeting system attack targeting many core systems," *Journal of System Architecture*, vol. 109, pp. 1–11, 2020.

[34] A. Elwhishi, P. H. Ho, and K. Naik, "ARBR: Adaptive reinforcement-based routing for DTN," in *Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Ontario, Canada, 2010.

[35] R. Plate and C. Wakayama, "Utilizing kinematics and selective sweeping in reinforcement learning-based routing algorithms for underwater networks," *Ad Hoc Networks*, vol. 34, pp. 105–120, 2015.

[36] V. G. Rolla and M. Curado, "A reinforcement learning-based routing for delay tolerant networks," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2243–2250, 2013.

[37] C. Wu, T. Yoshinaga, D. Bayar, and Y. Ji, "Learning for adaptive anycast in vehicular delay tolerant networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 1379–1388, 2019.

[38] Y. Zhang, P. Sun, and Y. Yin, "Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning," in *Proceedings of IEEE Intelligent Vehicles Symposium*, Changshu, China, June 2018.

[39] S. Vimal, M. Khari, R. G. Crespo, L. Kalaivani, N. Dey, and M. Kaliappan, "Energy enhancement using Multiobjective Ant colony optimization with Double *Q* learning algorithm for IoT based cognitive radio networks," *Computer Communications*, vol. 154, no. 1, pp. 481–490, 2020.

[40] Q. Zhang, M. Lin, and L. T. Yang, "A double deep Q-learning model for energy-efficient edge scheduling," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 739–749, 2018.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5577514, 18 pages
https://doi.org/10.1155/2021/5577514

Research Article

Design and Simulation of Lightweight Identity Authentication Mechanism in Body Area Network

Jianglong Yang, [1] **Wanlin Yang**, [2] **Huwei Liu**, [1] **and Li Zhou** [3]

[1] *School of Management Engineering, Capital University of Economics and Business, Beijing, China*
[2] *School of Computer Science and Engineering, Northeastern University, Shenyang, China*
[3] *School of Information, Beijing Wuzi University, Beijing, China*

Correspondence should be addressed to Huwei Liu; darion8@163.com

Received 5 January 2021; Revised 14 March 2021; Accepted 27 March 2021; Published 19 April 2021

Academic Editor: Zhe-Li Liu

Wearable medical devices rely on the human body to form a small LAN around the human body, called body area network (BAN). Users can use these devices to monitor the changes of various body indicators in real time. The physiological data involved in this process belongs to personal privacy. Therefore, the security requirements of BAN are relatively high, and its current research focus is on authentication mechanisms. To meet the requirements of security and resource consumption of BAN, this paper proposes a lightweight identity authentication mechanism that meets the characteristics of BAN resource constraints. Based on the characteristics of BAN, a simple and mature star topology structure is applied to establish the network model of BAN. For the human body in normal situations and emergencies, the corresponding authentication mechanism and encryption and decryption method of physiological data are designed by using the physical unclonable function (PUF) and cloud database, physiological data, and cross-correlation algorithm. Furthermore, the formal and informal security analysis of the designed authentication mechanism proves that the authentication mechanism designed in this paper has certain security, and the lightweight authentication mechanism is simulated and evaluated. The experimental results show that compared with the benchmarking mechanism, the authentication mechanism designed in this paper solves more security problems and has certain advantages in terms of calculation cost, communication cost, and energy cost.

1. Introduction

In recent years, wearable devices are developing at an amazing speed, followed by intelligent and interconnected medical sensor devices and the popularization of medical sensor networks [1, 2]. With more and more medical sensor devices for monitoring and treatment on the human body, researchers have put forward the concept of personal local area network based on the human body [3], which is the predecessor of BAN. BAN refers to a kind of network attached to the human body, which is composed of medical sensor equipment implanted in the human body or worn on the human body surface. Data collected by nodes is transmitted to the remote server by external equipment for medical diagnosis [4, 5]. The existence of the BAN and wearable devices enables individuals to collect their own physiological data in real time and monitor their physical activity and health status [6]. It is very convenient to know their own physical condition without going to the hospital regularly for examination, and its existence will not have any impact on people's life. With the aggravation of the aging of the social population, the number of chronic disease cases that need to consume a lot of human, material, and financial resources is also increasing. BAN has become the best choice to solve this problem [7]. The medical sensor equipment deployed in the BAN collects physiological data of different indicators of the human body. These data belong to personal privacy and cannot be accessed without permission. If we do not take any measures, then these data are easy to obtain, which may leak personal privacy and affect people's life, work, and other aspects, so the research on the security of BAN is urgent.

Therefore, based on the highly secure and limited resource requirements of the BAN, a lightweight authentication mechanism is constructed. (1) Under normal conditions, design the authentication mechanism and encryption and decryption method of physiological data using PUF and cloud database. (2) Design the authentication mechanism in emergencies using physiological data and cross-correlation algorithm. (3) Carry out formal and informal security analysis for the designed authentication mechanism. (4) Simulate and evaluate the lightweight authentication mechanism.

2. Related Works

The emergence of BAN brings new opportunities and challenges to human health care. The network connection of wearable devices and implantable medical devices has been quite advanced, but its system security problem has not been effectively solved, and its security mechanism is relatively weak, which has been attacked within the scope of the network, resulting in device security problems [8]. If we cannot solve all kinds of security problems existing in the BAN, then it is very difficult to be applied in the field of health care because the existence of various security problems will cause great harm to the human body. At present, research studies on the security of BAN are springing up. Among them, there are many research studies on authentication mechanisms in BAN.

Singla and Sachdeva [9] proposed a two-stage authentication mechanism. The first stage is the authentication between the medical sensor device and the receiver. Since the authentication resources are limited in this stage, it is necessary to design a lightweight authentication method; the second stage is the authentication between the receiver and the server, which is not limited by resources, so the traditional encryption method can be used. Based on the three-layer communication of the BAN, the mechanism is comprehensive, but it does not solve the problem of resource consumption reasonably. Under the premise of lightweight protocol and the requirement of physiological data sensitivity, Gritti et al. [10] limited the public information that could be obtained by the device. If more physiological data need to be obtained in some cases, such as an emergency, the access rights of personal information need to be realized based on device authentication. In reference [11], a device-to-device authentication scheme is proposed, that is, the authentication mechanism between two sensor nodes deployed on the human body. The key is always fresh and secure from the user's gait mode through the change of instantaneous acceleration. This method is only resistant to noise attacks (generated by sensor nodes when transmitting data) and active attacks. Anada [12] proposed a distributed multiauthority anonymous authentication scheme for the Internet of Things and blockchain, in which the verifiers were noninteractive. This scheme can dynamically increase/decrease the independent attribute permissions. When an entity wants the authority to issue attribute credentials, the authority only needs to generate a digital signature on its global identity to solve the problems of resource consumption and authentication reliability. Based on the comprehensive consideration of data connectivity and user privacy and the joint scenario of eduGAIN, STORK, and eIDAS. Torroglosa-Garcia and Skarmeta-Gomez [13] proposed an interoperability mechanism for data connectivity to reduce the recurrence of identity authentication.

Some authentication mechanisms shield sensor nodes and ensure their security by adding proxy devices between sensor nodes and control nodes. Denning et al. [14] proposed a new method for the safety of the implantable medical device system from another angle, which added an additional external device (agent) to ensure the safety of the implantable medical device. The agent has no resource limitation. Some operations between the control node and the medical sensor device are completed by the agent. Only the agent exists outside, and the implantable medical device does not exist. The agent can solve some problems, but it also brings new and additional security risks to the system. In reference [15], a distance boundary method is proposed, in which a piezoelectric implantable device is implanted 1 cm or deeper into the human skin, and the random key is generated and sent out by means of sound emission, which can only be received by the control node at a certain distance. This method can resist remote attacks, but the addition of additional devices will bring new security problems, and the method of transmitting data by means of sound transmission itself will have security problems.

Other authentication mechanisms are designed using PUF. Lee et al. [16] proposed a mutual authentication mechanism between the control node and sensor node by using PUF. The challenge-response pairs generated by PUF were used, and each node was needed to complete the hash function operation and MAC operation. The mechanism designed in this paper can solve some security problems. However, resource consumption is not analyzed in this paper, and a good balance between resource consumption and security is not achieved. In reference [17], a mutual authentication mechanism between sensor nodes is designed by using PUF. In this method, the control node acts as the third party, and the key stored in the third-party control node is the shared key between two sensor nodes, which guarantees the reliability of data transmission when the sensor nodes authenticate with each other. The implementation of this method is based on the assumption that the control node is trusted, but the control node is not necessarily trusted.

Some use the characteristics of physiological signals to design authentication mechanisms. Steffen et al. [18] designed a secure authentication mechanism in BAN using ECG signals to identify whether sensor nodes are attached to the same body. The process of the mechanism is as follows: firstly, the sensor nodes including the analog filter and data preprocessor are deployed; then, the features of the data after preprocessing are extracted by technical means, and finally, the parameters of authentication protocol are determined by the extracted features. Hu et al. [19] designed a key protocol based on ordered physiological signals. The protocol authenticates control nodes and implantable medical devices by extracting and quantifying ECG signals. Rostami et al.

[20] designed a contact access mechanism, in which the control node contacts the human body, extracts the heart rate signal of the human body, analyses the pulse interval, and designs a new encryption pairing protocol to achieve good authentication between the control node and the implantable medical device. This method can be implemented in the existing devices without the need for additional devices. But when people are in a critical situation and need to receive treatment in time, the demand for speed is far higher than safety.

The medical sensor equipment in the BAN collects the physiological data of the human body in real time, which leads to the existence of a large amount of data on the local side. The storage of these data needs to consume a lot of local resources. Therefore, some scholars propose that the BAN and cloud should be integrated. Wan et al. [21] pointed out that, with the support of mobile cloud computing, the implementation of deploying the wireless human local area network (WLAN) on a large scale in pervasive medicine can be enhanced, but there are some technical problems and challenges in the process of integrating wireless BAN and mobile cloud computing. This paper introduces the framework of the cloud-assisted human LAN, as well as the challenges of routing mechanism, cloud resource allocation mechanism, semantic attack, and data security. According to the characteristics of telemedicine, an efficient cloud-assisted message authentication scheme was proposed in reference [22]. In this scheme, the cloud server is responsible for storing and transmitting the encrypted data of patients to doctors for diagnosis and treatment, and then, the processing results are stored in the cloud server. Through the security analysis and performance evaluation of this scheme, it is concluded that the scheme not only ensures the privacy of stored human physiological data but also meets the purpose of saving local resources. Yu et al. [23] proposed a new solution to the security and privacy threats in the cloud-assisted wireless BAN, focusing on the confidentiality and integrity of data. The data confidentiality is guaranteed by the improved order-preserving symmetric encryption method, and the data integrity is guaranteed by using the virtual linear segmentation method. Although this method saves the resources at one end of the sensor node, it increases the cost of the remote control unit. The future research direction is to reduce the cost of both ends at the same time, instead of sacrificing one end to complete the other.

To sum up, at present, there are a lot of research studies on the authentication mechanism of BAN, but many designs do not meet the requirements of resource limitation and high security of BAN at the same time.

3. Lightweight Authentication Mechanism in BAN

3.1. Network Model. At present, two kinds of topology structure are widely used in BAN. One is two-layer star topology, that is, some sensor nodes need two hops to send data to the control node, and the other is star topology, that is, sensor nodes only need one hop to send data to control nodes. BAN is a network with very limited resources. If one node interacts with the master node, it will consume a lot of resources. If it needs to interact with other nodes in the same network, it will consume resources faster. Moreover, even if the data of one node is transmitted to another node, the data is meaningless to this node at present because this node cannot process the data. This paper uses a mature, simple, and representative star topology. Figure 1 shows the star topology. As shown in the figure, there are N sensor nodes (medical sensor equipment) $\{S_1, S_2, S_3, \ldots, S_n\}$ deployed on the guardians. Each node has an ID (unique and greater than 1), and there is no connection between these nodes. There is a control node whose ID value is equal to 0. It is used to collect the data collected by all sensor nodes deployed on the human body and is responsible for communicating with the outside world and sending the data to the medical staff so that the medical staff can make a correct and effective judgment on the health status of the guardians in time. The control node can communicate with all sensor nodes in one hop. Because different sensor nodes represent different types of medical sensor equipment and have different functions, there are many sensor nodes deployed in the human body in the BAN. It is mentioned in reference [2] that medical sensor equipment is divided into three security levels (I, II, and III). The greater the category number, the higher the risk level. Different security levels correspond to different medical sensor devices (sensor nodes).

3.2. Authentication Mechanism under Normal Circumstances

3.2.1. Authentication Protocol. In the design of normal authentication protocol, this paper uses the physical unclonable function and cloud database to achieve. Once the authentication mechanism is used on the device, it will last for a lifetime until the device is not available. It can be imagined that if the authentication mechanism is designed by using PUF, a large number of challenge-response pairs are needed, and the relationship between these challenge-response pairs is one-to-one. In order to avoid unnecessary waste of local resources due to the storage of a large number of challenge-response pairs, these challenge-response pairs are stored in the cloud database.

In authentication protocol, some data need to be transmitted between two entities, so the freshness, integrity, and nonrepudiation of data should be guaranteed in the process of transmission. Among them, freshness refers to the guarantee that the data used is up-to-date rather than has been used; integrity refers to ensuring that the messages in transmission are not partially missing due to malicious attacks by attackers; nonrepudiation means that if an entity has sent the message, it must ensure that the entity has no reason to deny this fact.

In order to authenticate both parties as trustworthy entities to each other, this paper designs a two-way authentication protocol, that is, the sensor node should prove to the control node *phone* that it is an honest entity, and the control node should also prove itself to the sensor node that it is an honest entity. Therefore, it is necessary to verify that the control node is trusted during authentication. It is not feasible for malicious sensor nodes to use the previous

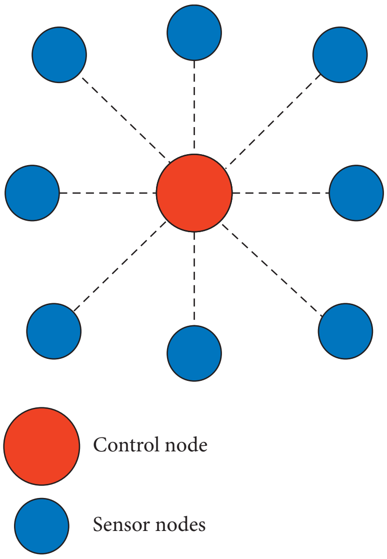


Figure 1: Star topology.

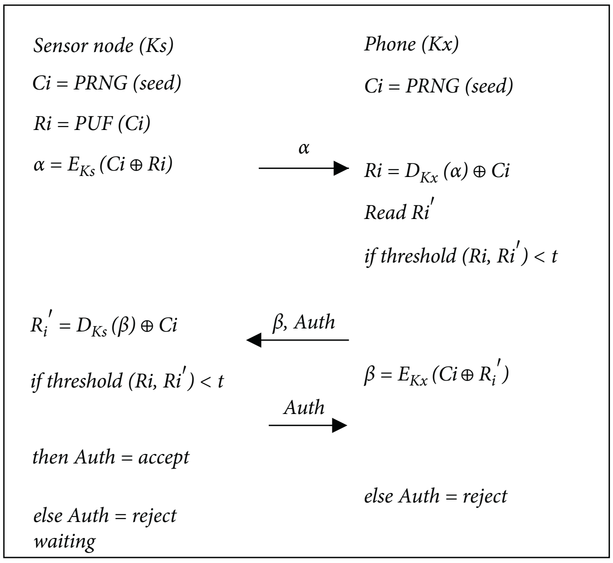

Sensor node (Ks)
Phone (Kx)
Ci = PRNG (seed)
Ci = PRNG (seed)
Ri = PUF (Ci)
$\alpha = E_{Ks}(Ci \oplus Ri)$
α
$Ri = D_{Kx}(\alpha) \oplus Ci$
Read Ri'
if threshold $(Ri, Ri') < t$
$R_i' = D_{Ks}(\beta) \oplus Ci$
β, Auth
if threshold $(Ri, Ri') < t$
$\beta = E_{Kx}(Ci \oplus R_i')$
Auth
then Auth = accept
else Auth = reject
else Auth = reject
waiting

Figure 2: Authentication protocol under normal circumstances.

challenge-response pairs to achieve the purpose of authentication because the principle of challenge-response pairs is to discard one by one. When the control node obtains the previously used incentive response pair, it will not find a matching response in the database, then the authentication will not succeed. Figure 2 shows the normal authentication protocol, mainly including the initialization phase and authentication phase. The symbols used in the protocol are illustrated in Table 1.

(1) Initialization Phase. In order to ensure the normal use of the protocol, we need to complete the following initialization work. Firstly, the challenge-response pairs required by the authentication process are stored in the cloud database. Secondly, the initial seed of the butterfly seed generation algorithm [24] is set. The key can then be generated using this method. The butterfly seed generation algorithm can get the required seed, and only a few bit changes can get random results. The method to get random seeds by using the variable function is to invert these bits bit by bit from the least significant bit to the j bit. j can be a value-added number or other to improve the unpredictability where sin *it* is the initial seed, Sj is the current seed, $Sj = \varphi(s)$ is the seed variation function, and $r = g(Sj)$ is the random number generation function.

(2) Certification Phase. In the authentication phase, both entities authenticate each other's trustworthiness. In the authentication process, sensor nodes and control nodes use the pseudorandom generation sequence to generate excitation Ci. The sensor node uses its own implanted PUF and Ci to execute PUF to generate response Ri, then uses the key Ks to encrypt the results of excitation Ci and response Ri XOR, and sends the encryption results α to the control node. The control node decrypts the received α to get Ri, then extracts its own stored Ri from the database, and records it as Ri'. If the error between the two is within an acceptable range, then the authentication is successful, that is, the other party is an honest and trustworthy entity. Similarly, the control node sends a message β to the sensor node to prove its identity. After receiving the message, the sensor node decrypts the message β by using the key Kx to obtain Ri' and compare it with its own Ri to verify whether the party sending the message is a trusted entity. If so, mutual authentication is successful. The algorithm pseudocode of authentication protocol is as follows. (Algorithm 1)

Table 1: Symbol description.

Symbol	Description
Ci	Challenge
Ri	Response generated by sensor nodes
Ri'	Response when reading from cloud database
Ks	Key of sensor node
Kx	Key of phone
seed	Seed
t	Critical value

3.2.2. Information Encryption and Decryption Process. After the sensor node and the control node prove that they are honest and trustworthy entities through the authentication protocol, the sensor node sends the collected human physiological data to the control node. In this process, the transmitted information needs to be encrypted to ensure its security. Figure 3 is the process flow chart of encryption and decryption. The sensor node performs a pseudorandom generator to generate challenge, then uses the generated excitation and the human physiological signal collected by the sensor node for XOR, that is, encryption, and then sends the message to the control node; after receiving the message, the control node gets the excitation generated by its own end and decrypts the specific and correct human physiological data according to the excitation. The control node can also access the cloud database to obtain the physiological data

```
//INPUT: The excitation response pairs generated by PUF
//OUTPUT: Authentication success/failure message
BEGIN
Execute butterfly seed generation algorithm to generate *seed* and key $Ks$, $Kx$;
phone and sensor nodes simultaneously execute pseudo-random sequence generator to generate $Ci$;
The sensor node calculates the value of $\alpha$ by $\alpha = E_{Ks}(Ci \oplus Ri)$, and then sends it to phone;
phone reads the database to get $Ri'$, and calculates $Ri$ by $Ri = D_{Kx}(\alpha) \oplus Ci$;
IF the error between $Ri$ and $Ri'$ is within the acceptable range, threshold$(Ri, Ri') < t$;
THEN phone authenticates the sensor node successfully, considers the sensor node to be a trusted entity, and sends authentication success message, Auth = accept;
ELSE send authentication failure message, Auth = reject;
    phone calculates $\beta$ by $\beta = E_{Kx}(Ci \oplus Ri')$ and sends it to sensor node;
END IF
IF the error between $Ri$ and $Ri\prime$ is within the acceptable range, threshold$(Ri, Ri') < t$;
    THEN the sensor node authenticates phone successfully and considers phone as a trusted entity, and sends an authentication success message, Auth = accept;
ELSE send authentication failure message, Auth = reject;
END IF
END
```

Algorithm 1: Authentication protocol under normal circumstances.

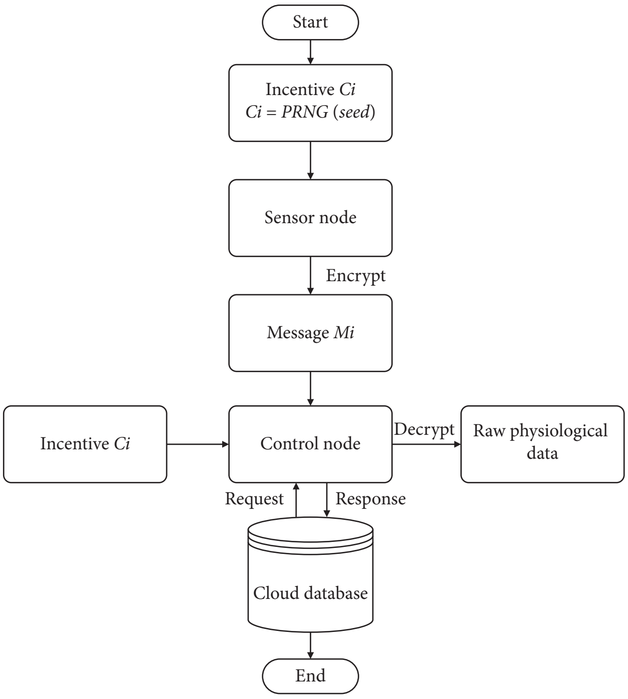


Figure 3: Encryption and decryption process.

needed. The secure data transmission between the cloud database and the control node is realized by using the traditional encryption method.

Because the cloud database also has unsafe factors, two steps need to be done here. First, the challenge-response pairs generated by PUF are stored in the cloud database by the XOR encryption method. In order to prevent security problems in data transmission, the data in the transmission process is encrypted by the XOR method, and the cloud database also stores the data after XOR. Second, doctors obtain cloud database data through traditional encryption methods (such as AES). Therefore, after obtaining XOR encrypted data, cloud database uses traditional encryption methods for stronger security. The details are shown in Figure 4.

3.2.3. Design of Safety Assurance. (1) Security guarantee of authentication protocol: freshness. A large number of challenge-response pairs are needed in the designed authentication mechanism. The use principle of these challenge-response pairs is to use up one and then discard it, and there will be no use of two duplicate challenge-response pairs. This method ensures that the challenge-response pairs used in each certification are fresh.

(2) Security guarantee of authentication protocol: nonrepudiation. When one entity sends data to another entity, the sending entity cannot deny that it has sent the data. In this paper, the unique ID number of each medical sensor device on the human body is used to ensure nonrepudiation. The ID number of the medical sensor equipment and the seed change function are combined into a function. The combination function is used as the parameter of the random number generation function as $r = g(f(\varphi(Sj), \mathrm{ID}))$, where f is a combination function.

(3) Security guarantee of authentication protocol: integrity. Because of the limited resources of the BAN, a critical value is used to judge whether the authentication is successful or not. In addition, the frequency and times of the implementation of the authentication mechanism should be considered. Therefore, in the authentication mechanism, it is necessary to design a method to check the integrity rather than to ensure the integrity [25]. The specific method is as follows: divide the message into n parts, each part has $L -$ bit, and take the bit in the corresponding position out from each part; there are $n$bit*s* in total, then XOR these bit*s* to get a new bit block, so as to get L blocks, and then XOR L blocks to get a new message. Both entities need to perform this process. If

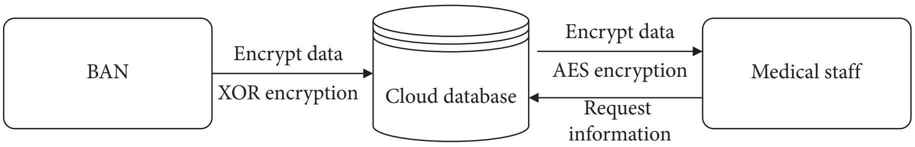


Figure 4: Data encryption and decryption in the cloud database.

the final result is the same, then it can be considered that the message has not been tampered within the transmission process, as shown in Figure 5.

(4) Security assurance of encryption and decryption: freshness. The challenges used in the encryption process are not like the challenge-response pairs used in the above authentication mechanism, which are discarded once, but need to be saved to the cloud database so that the collected physiological data can be decrypted. The challenge response here can be reused. Therefore, the freshness can be guaranteed by adding time variables into the formula as $r = g(f(\varphi(Sj), ti))$.

(5) Security assurance of encryption and decryption: nonrepudiation. The guarantee of nonrepudiation is realized by ID of the sensor node as $r = g(f(\varphi(Sj), ID, ti))$.

(6) Security assurance of encryption and decryption: confidentiality. The guarantee of confidentiality is to use the response generated by PUF as the key to encrypt the information. Because the response of PUF will be affected by temperature and environment, the response generated each time will be different. However, as a key, the response is required to be the same every time so that the encrypted message can be correctly decrypted. Because the instability of PUF is difficult to solve, it is not appropriate to use Ri encryption. At present, Ci encryption is considered. The guarantee of confidentiality is $E(Mi) = Ci \oplus Mi$, which utilizes the XOR operation on the challenge Ci and message Mi. The decryption process of the received message is to XOR the message, and Ci is obtained from the cloud database as $Mi = \mathrm{E}(Mi) \oplus Ci$.

(7) Security assurance of encryption and decryption: integrity. Because the data encryption and decryption process require higher integrity, we cannot use the method of integrity checking to determine whether the message has been tampered with, but use the method to ensure the integrity of the message [26]. This paper uses the method of inserting parity bits into messages. The specific operation is as follows: first, add the check bit to the message, and then, encrypt the message with the check bit, and finally, get the Frame Check Sequence (FCS) and the encrypted message body. The check bit is selected by the control node and broadcast to all sensor nodes in the deployment phase. The message structure after adding check bits is shown in Figure 6.

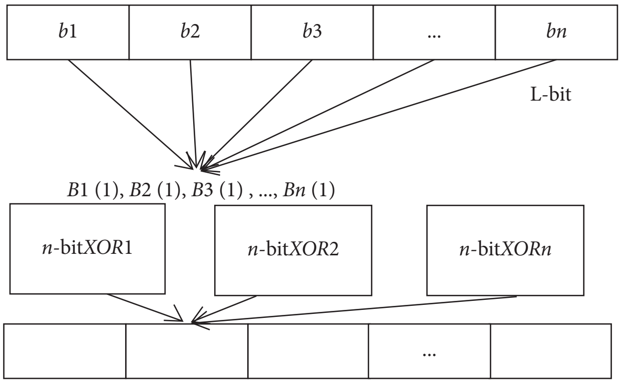


Figure 5: Integrity test.

3.2.4. *Safety Analysis.*

Formal and informal security analysis methods are mainly used to analyze the security of the designed authentication mechanism under normal conditions. The formal security analysis method is BAN logic. Through the analysis, it can be proved that the authentication mechanism has certain security.

(1) Formal Security Analysis. In this paper, BAN logic is used to analyze the formal security of authentication protocol under normal conditions. Assuming that the communication between external devices and the cloud database is secure, then the cloud database and control node can be regarded as a whole DP. The following is the formal security analysis of the authentication protocol designed in this paper.

The purpose of mutual authentication between the control node and sensor node is to ensure that both sides receive the data from the trusted entity. If the authentication purpose is expressed by expressions, the expressions are as follows:

$$\begin{aligned} &\text{phone}| \equiv \{\text{if DP has } Ri \approx Ri' \text{ then phone}| \equiv Ri\},\\ &\text{sensor}| \equiv \{\text{if sensor has } Ri' \approx Ri \text{ then sensor}| \equiv Ri'\}. \end{aligned} \tag{1}$$

The initialization assumptions for the authentication protocol are as follows:

$$\begin{aligned} &S1\colon \text{sensor}| \equiv \text{sensor} \overset{(Ks,Kx)}{\leftrightarrow} (Ks, Kx)\text{DP}, S2\colon \text{DP}| \equiv \text{sensor} \overset{(Ks,Kx)}{\leftrightarrow} (Ks, Kx)\text{DP},\\ &S3\colon DP| \equiv \#(Ci),\ S4\colon \text{sensor}| \equiv \#(Ci \oplus Ri),\\ &S5\colon \text{DP}| \equiv \text{sensor} \Rightarrow Ri,\ S6\colon \text{sensor}| \equiv \text{DP} \Rightarrow Ri'. \end{aligned} \tag{2}$$

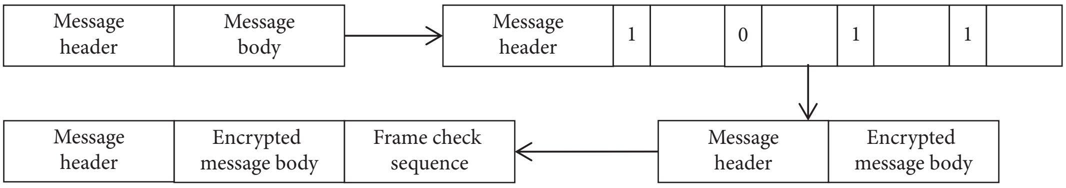


Figure 6: Message structure.

The ideal model of the authentication protocol is shown as follows:

$$\begin{aligned} \mathrm{DP} &\longrightarrow \text{sensor: Query}, \\ \text{sensor} &\longrightarrow \mathrm{DP}\text{: } \left\{Ci, (Ci\oplus Ri)_{Ks}\right\}(\alpha), \\ \mathrm{DP} &\longrightarrow \text{sensor : } \left\{\left(Ci\oplus Ri_{Kx}{'}\right)\right\}(\beta). \end{aligned} \tag{3}$$

The analysis process of the authentication protocol is as follows.

According to the above initialization assumption and idealized model of the authentication protocol designed in this paper, the formal security analysis of the protocol is given below. First, DP will receive a message α from the sensor node, thus obtaining $\mathrm{DP}\nabla\left\{(Ci\oplus Ri)_{Ks}\right\}$.

Combined with the hypothesis $S2$, according to the formal reasoning criterion of the BAN logic criterion, $\mathrm{DP}| \equiv \text{sensor} \sim (Ci\oplus Ri)$ is obtained, which shows that DP believes that α is sent by a sensor node with the same key as it. According to the message freshness criterion in $S3$ and BAN logic criteria, $\mathrm{DP}| \equiv \#(Ci\oplus Ri)$ is obtained. From $\mathrm{DP}| \equiv \text{sensor} \sim (Ci\oplus Ri)$ and $\mathrm{DP}| \equiv \#(Ci\oplus Ri)$, according to the random number verification criterion in the BAN logic criterion, $\mathrm{DP}| \equiv \text{sensor}| \equiv (Ci\oplus Ri)$ can be obtained. From $\mathrm{DP}| \equiv \text{sensor}| \equiv (Ci\oplus Ri)$ and belief union criterion in the BAN logic criterion, we can get $\mathrm{DP}| \equiv \text{sensor}| \equiv Ri$. If the response is extracted from the cloud database according to the challenge, then according to $\mathrm{DP}| \equiv \text{sensor}| \equiv Ri$ and protocol initialization hypothesis $S5$, it is concluded that $\mathrm{DP}| \equiv Ri$. If not, the message α is sent by the attacker, and the execution of the protocol is terminated.

If there is Ri', the sensor node receives the message β from DP and gets $\text{sensor}\,\nabla\left\{(Ci\oplus Ri')_{Kx}\right\}$. Combined with the hypothesis $S2$, according to the message meaning criterion in the BAN logic criterion, $\text{sensor}| \equiv \mathrm{DP} \sim (Ci\oplus Ri')$ is obtained. In other words, the sensor node believes that β is sent by a DP, which has a shared key with it. From the hypothesis $S4$, it is concluded that $\text{sensor}| \equiv \#(Ci\oplus Ri')$. According to $\text{sensor}| \equiv \mathrm{DP} \sim (Ci\oplus Ri')$ and $\text{sensor}| \equiv \#(Ci\oplus Ri')$, combined with the random number verification criterion in the BAN logic criterion, $\text{sensor}| \equiv \mathrm{DP}| \equiv (Ci\oplus Ri')$ is obtained. According to $\text{sensor}| \equiv \mathrm{DP}| \equiv (Ci\oplus Ri')$ and belief union criterion in the BAN logic criterion, it is concluded that $\text{sensor}| \equiv \mathrm{DP}| \equiv Ri'$. If there is $Ri' \approx Ri$ in the sensor node, it can be concluded from $\text{sensor}| \equiv \mathrm{DP}| \equiv Ri'$ and $S6$ that $\text{sensor}| \equiv Ri'$. It can be seen from the $\mathrm{DP}| \equiv Ri$ and $\text{sensor}| \equiv Ri'$ that the authentication protocol under normal conditions can withstand the logical reasoning authentication of BAN. The message received by the sensor node or control node is indeed sent by the trusted control node or sensor node, and the two are mutually trusted entities.

(2) Informal Security Analysis. Informal security analysis is the security analysis of the normal authentication protocol's resistance to attacks mainly including eavesdropping attack, replay attack, forward/backward security, and middleman attack.

(3) Informal Security Analysis (Eavesdropping Attack). Attackers use the data overheard in the transmission process of sensor nodes and control nodes to conduct improper activities. Ks is used to encrypt the information transmitted between the sensor node and the control node, but the eavesdropping attacker does not know about Ks, so the messages between the sensor node and the control node cannot be eavesdropped and recorded.

(4) Informal Security Analysis (Replay Attack). The data packet received by the sensor node or control node is sent again by attackers to get away with the other party's authentication, so as to cheat the sensor node or control node. Suppose the attacker sends the message α to the control node, which has been sent α before, but because the seed has changed at this time, Ks, Ci, and Ri' obtained after receiving α for the first time have changed, and finally, the authentication fails. Suppose the attacker sends the message β to the sensor node, which has been sent before. Because the sensor node performed a series of operations after receiving β last time, resulting in changes in Ks and Ci, so the final authentication will not succeed.

(5) Informal Security Analysis (Forward/Backward Safety). Forward security means that the attacker cannot get the previous data from the known data. Backward security means that the attacker cannot use the current data to carry on the malicious attack to the later operation. There is no relationship between the challenge-response pairs generated by PUF, so attackers cannot infer the used or future challenge-response pairs according to the existing. There is no relationship between the former seed and the latter seed. It is difficult for attackers to analyze and infer useful information based on the existing seeds.

(6) Informal Security Analysis (Middleman Attack). The attacker steals the transmitted data and masquerades as an aggressive sensor node or a control node to maliciously attack the authentication mechanism. Authentication protocol guarantees the integrity and freshness of the transmission message, and due to the nonclonality of PUF and the honesty of the control node, if the message changes in the transmission process, it can be detected.

3.3. Authentication Mechanism in Emergency. In the BAN, an urgent problem to be solved is that, in case of emergency, medical staff can access the medical equipment worn by patients, without authentication or simplified authentication, so as to know the patient's physical condition in time, reconfigure the equipment parameters, and timely treat the patients. At the same time, due to the sensitivity and complexity of patients' electronic health records and physiological data, the access rights of medical staff to patients' data should be limited in a specific range [27]. In this paper, the cross-correlation algorithm is used to calculate the correlation between the two signals, so as to know whether the patient is in a normal situation or an emergency and then take different measures. In case of emergency, the message is broadcast to the receiving device within a safe distance. In this way, in case of emergency, even if the doctor is not the commonly used treatment doctor of the patient, he can also obtain access right to the medical equipment of the patient in time so that the patient can be treated in time.

3.3.1. Selection of Physiological Signals. Now many heart patients have implanted the cardiac pacemaker, if the human body has a pacemaker, then the physiological signal will choose the ECG signal. If the human body does not wear a pacemaker, then the heart rate signal can be selected as the physiological signal. Now the bracelet, wristwatch, and so on can measure the human body's heart rate signal, this signal is related to the heart beat, easy to find the human body's health problems. The physiological signal used in this paper is the ECG signal. In an emergency, the reasons for choosing signals like this are as follows: first, most of the sensor nodes deployed on the human body will contact the blood vessels of the human body, and most sensor nodes can monitor the heart rate signal; second, in an emergency, the patient's pulse changes obviously, and using this signal will be easier and faster to detect human health problems than other signals.

The ECG signal of normal people is shown in Figure 7. Among them, the wave with the small waveform and similar shape to the sin function from 0 to π is the P wave. The wave with a flat shape and low amplitude (not less than 1/10 of the R wave) is the T wave. The most dramatic change of the waveform is the QRS wave group, which is composed of the Q wave with the downward waveform, R wave with the upward waveform, and S wave with the downward waveform. The interval between the starting point of the QRS wave group and the ending point of the T wave group is called the QT period. In a complete ECG signal diagram, the duration of the P wave is 0.08~0.11s, the duration of the PR interval is 0.120~0.200s, the duration of the QRS wave group is 0.06~0.10s, and the duration of the QT period is 0.340~0.430s. When the human atrium is excited, the P wave will be generated. The waveform generated by the right atrium is similar to the sin function from 0 to $(\pi/2)$, while that of the left atrium is from $(\pi/2)$ to π. If someone is older or has a slower heart rate, his PR interval will be longer than normal people. The P wave, QRS complex wave, and T wave all represent the potential change. The former two represent the depolarization process, and the latter represents the ventricular repolarization process. The P wave and QRS complex wave are used to describe two atria and two ventricles, respectively.

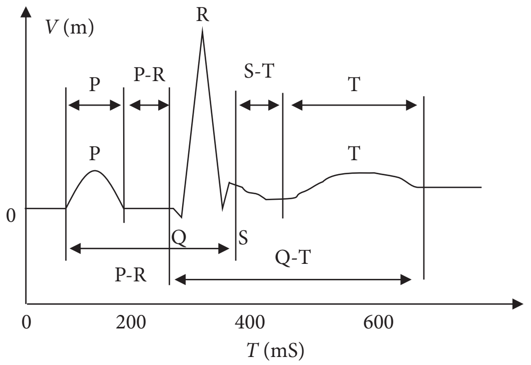


Figure 7: Electrocardiogram signal.

3.3.2. Design of the Cross-Correlation Algorithm. According to the use environment and purpose, this paper improves the cross-correlation algorithm to determine whether the current human body is in an emergency. As shown in Figure 7, the waveform is divided into some segments or intervals by five special points. The waveform near each point represents different heart conditions, and the abnormal waveform represents different heart problems. The heart problems of each person are different, and the waveform changes around each point are also different when conditions occur. So, we divide an ECG waveform into three parts: PR interval, QRS wave group, ST segment and T wave. Then, set the weight values of the three parts according to the different disease conditions of each person. If the weight is 0.8, 0.2, and 0.2, it means that if the patient has an emergency, the PR section is easy to appear abnormal. This approach makes the method more targeted and more accurate.

In the process of execution, if the selected ECG signal cycle is as shown in Figure 7, the result may have a large error because it is impossible to accurately obtain the start time and end time of the cycle, so it is difficult to obtain the time cycle. Therefore, the cycle shown in Figure 8 is adopted in this paper.

According to the choice of segment and period of the ECG signal, we can assume that the duration of a cycle is 800 ms, then the PR interval accounts for 300 ms, ST segment and T wave part account for 300 ms, the first half QRS interval accounts for 100 ms, and the second half QRS interval accounts for 100 ms.

The control node is responsible for executing the cross-correlation algorithm to get the correlation number. The control node receives the ECG signal sent at Ti time and then performs the cross-correlation algorithm with the ECG signal stored at $Ti - 1$ time to judge the correlation degree. If x represents the signal sent at Ti time and y represents the signal sent at $Ti - 1$ time, the calculation formula of the cross-correlation function is as follows:

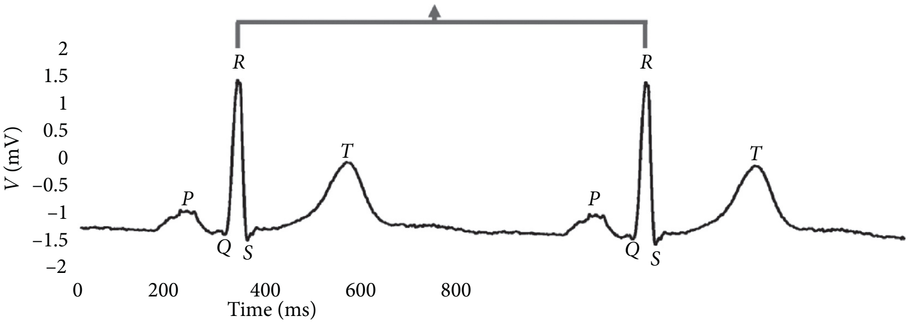


Figure 8: Time period.

$$R_{xy} = \frac{1}{n}\sum_{i=1}^{n}(w_1 x(\mathrm{PR})y(\mathrm{PR}) + w_2 x(\mathrm{QRS})y(\mathrm{QRS}) \qquad (4)$$
$$+ w_3 x(T)y(T)).$$

Among them, $x(\mathrm{PR})$ represents the data of the PR part at Ti time, $y(\mathrm{PR})$ represents the data of the PR part at $Ti-1$ time, and the other two represent data of the QRS and T part, respectively. $W1$, $W2$, and $W3$ represent three weight values, $W1 + W2 + W3 = 1$. n means averaging the data.

After getting the results of the cross-correlation algorithm, it is necessary to normalize the results in order to judge whether it is an emergency.

The average value of the signal $x(t)$ and $y(t)$ is calculated as follows:

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n}(w_1 x(\mathrm{PR}) + w_2 x(\mathrm{QRS}) + w_3 x(T)), \qquad (5)$$

$$\mu_y = \frac{1}{n}\sum_{i=1}^{n}(w_1 y(\mathrm{PR}) + w_2 y(\mathrm{QRS}) + w_3 y(T)). \qquad (6)$$

The calculation of the variance value of the signal $x(t)$ and $y(t)$ is shown in the following equations:

$$\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}\left\{(w_1 x(\mathrm{PR}i) + w_2 x(\mathrm{QRS}i) + w_3 x(Ti)) - \mu_x\right\}^2, \qquad (7)$$

$$\sigma_y^2 = \frac{1}{n}\sum_{i=1}^{n}\left\{[w_1 y(\mathrm{PR}i) + w_2 y(\mathrm{QRS}i) + w_3 y(Ti)] - \mu_y\right\}^2. \qquad (8)$$

The calculation of the correlation number is as follows:

$$\rho_{xy} = \frac{R_{xy} - \mu_x\mu_y}{\sqrt{\sigma_x^2}\sqrt{\sigma_y^2}}, \quad -1 \le \rho_{xy} \le 1. \qquad (9)$$

The pseudocode of the cross-correlation algorithm is described as follows. (Algorithm 2)

3.3.3. Determination of the Correlation Coefficient. The correlation coefficient is determined according to each person's physical condition. Because each person's physical condition is different, the possible disease situation is not the same, and the correlation degree between normal ECG and abnormal ECG is also different. For example, for patient A, when the similarity value is 0.8, it belongs to an abnormal condition, while for patient B, it may be normal. According to the different situations of each person, we plan to set the critical value of each person according to the characteristics of each person so that it can be closer to the real situation of patients and get more accurate final results. According to the above description, it is necessary to set the unique critical value of the patient in each control node, judge whether the patient is in an emergency according to the critical value, and then perform the corresponding operation.

3.3.4. Safety Analysis. The formal security analysis of the mechanism's resistance to attack in an emergency mainly includes long-range attack, close attack, false signal attack, and misjudgment during movement.

(1) Long-Range Attack. For BAN, the protocol is sent by the sensor node to the control node within a safe distance by broadcasting. If the attacker is 2 meters away, there is no way to obtain the data, and the attack is invalid.

(2) Close Attack. Within 2 meters, the attacker can obtain the data and perform operations on the sensor node. When the patient is in an emergency, if the attacker is right beside him, there is a device that can collect data, and no doctor or witness, the attack is hard to be prevented. Otherwise, the attack is invalid.

(3) False Signal Attack. The attacker sends the wrong or tampered signal to the control node, and the control node requires the sensor node to broadcast the message within a safe distance. After the control node receives the message sent by the attacker, the alert function of the control node works after the operation, and the attack will not cause harm to the human body.

(4) Misjudgment during Movement. The cross-correlation algorithm compares the cross-correlation degree of the two ECG signal waveforms. No matter the frequency acceleration or the amplitude enhancement will not affect the waveform, so the judgment result is that the human body is still in the normal condition. When the human body is in a real emergency, the waveform of the ECG signal will change, and the waveform characteristics of five obvious points will disappear.

```
//INPUT: ECG signals at $Ti$ and $Ti-1$
//OUTPUT: in normal/abnormal condition
BEGIN
IF the result of $Ti$ executing the algorithm is emergency;
    Select the ECG signal at $Ti-2$;
    Use equation (4) to calculate the results of the cross-correlation algorithm of ECG signals at $Ti-2$ and $Ti$;
    Use equation (9) to calculate the cross-correlation coefficient;
ELSE the result of $Ti$ executing the algorithm is normal;
    Select the ECG signal at $Ti-1$;
    Use equation (4) to calculate the results of the cross-correlation algorithm of ECG signals at $Ti-2$ and $Ti$ or $Ti-1$ and $Ti$;
    Use equation (9) to calculate the cross-correlation coefficient;
    IF the error between $\rho_{xy}$ and $\rho_{xy}'$ is acceptable, threshold$(\rho_{xy}, \rho_{xy}') > t'$;
        The human body is in normal condition, the certification under normal condition shall be carried out;
        ELSE the error between $\rho_{xy}$ and $\rho_{xy}'$ is not acceptable, threshold$(\rho_{xy}, \rho_{xy}') > t'$ does not hold;
        The human body is in an emergency, and the certification under emergency shall be carried out;
    END IF
END IF
END
```

Algorithm 2: Judge whether it is an emergency.

3.4. Information Access Based on Node Security. The application of medical sensor equipment in the human body is more and more common. There may be several or even more than ten medical sensor equipment in a person's body, including heart beat measurement, blood pressure measurement, and blood glucose measurement; each medical sensor equipment has different requirements for safety. The risk level of medical sensor equipment is divided into three categories, and the greater the category number, the higher the risk [2, 28]. Because each doctor has his own department, it is impossible to obtain all the physiological data of patients, so it is necessary to set access rights for doctors. In the local area network of the human body, the data monitored by the medical sensor equipment is transmitted to the control node. Doctors can access the control node to obtain the required data. As long as it can authenticate with the control node successfully, the authentication of the two can be realized traditionally because they have no resource restriction, or the control node authorizes doctors to access the contents in the database. After the request message sent by the doctor reaches the control node, the control node queries the table stored by itself. If it is found that the doctor requests the data in the medical sensor device with a low-risk level, it will authenticate with the doctor. If the authentication is successful, the data stored by the control node will be sent to the doctor. If it is found that the doctor requests physiological information collected by the medical sensor device with a high-risk level, the control node finds that it has no right to make decisions and needs the medical sensor device to make its own decisions. The control node sends the data to the doctor with the consent of the medical sensor device. Because of the corresponding settings in the control node, the control node not only sends the data collected by the medical sensor device to the doctor but also transmits the data collected by other medical sensor devices with lower security level than the device, which the doctor has the right to access and helps make accurate medical decisions. The information access process is shown in Figure 9.

The pseudocode of the information access process based on the security level of the sensor node is given below. (Algorithm 3)

4. Simulation Implementation and Performance Evaluation

In order to evaluate and verify the authentication mechanism designed in this paper, the simulation is carried out based on OMNET++ in Windows [29]. Its underlying programming language is C++. The simulation results of OMNET++ have been gradually recognized, which provide an important basis for us to use OMNET++ for simulation. Among them, the ECG data set used in an emergency is processed by MATLAB.

4.1. Simulation Environment

4.1.1. Simulation of Functional Modules. In this paper, the network includes three roles: sensor node, phone, and doc. The sensor node is the medical sensor device worn by the human body. Different sensor nodes are used to collect different physiological data of the human body. In the process of authentication protocol implementation, it acts as an authenticated entity and authenticates whether the external receiving data device is an honest entity. phone, namely, control node, is used to send the physiological data of the human body sent by sensor nodes through the network. In the process of authentication protocol implementation, it is used as an authenticated entity and to authenticate whether the external receiving device is an honest entity. doc, that is, medical staff or other people who need physiological data. The information transmission mode between various sensor nodes and phone is wireless, while the information transmission mode between phone and doc is wired. The network model based on three roles is shown in Figure 10 (five sensor nodes).

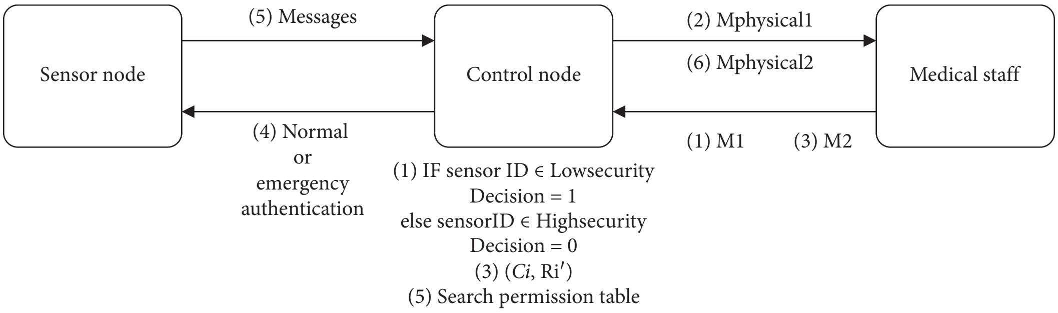


Figure 9: Process of information access.

```
//INPUT: request access to information.
//OUTPUT: physiological information collected by nodes.
BEGIN
Doctors send request message $M1$ to phone;
phone certifies doctors according to DoctorID in the message;
  IF the certification is successful, judge the safety level of medical sensor equipment according to SensorID;
    IF SensorID ∈ LowSecurity;
      phone sends physiological messages collected by phone to doctors;
    ELSE SensorID ∈ HighSecurity;
      phone sends unable to process message Not Message to doctors;
      After receiving the Not Message, the doctor sends the request message $M2$ to phone again;
    END IF
    phone judges the state of the human body according to Algorithm 2;
    IF the human body is in the normal state, execute Algorithm 1 for authentication;
      IF the authentication is successful, phone looks up the authority table and sends the information;
      ELSE Authentication failed;
      END IF
    ELSE the human body is in an emergency;
      Send physiological information to phone within safe distance;
    END IF
  ELSE Authentication failed. No message will be sent.
  END IF
END
```

Algorithm 3: Information access process based on the security level of sensor node.

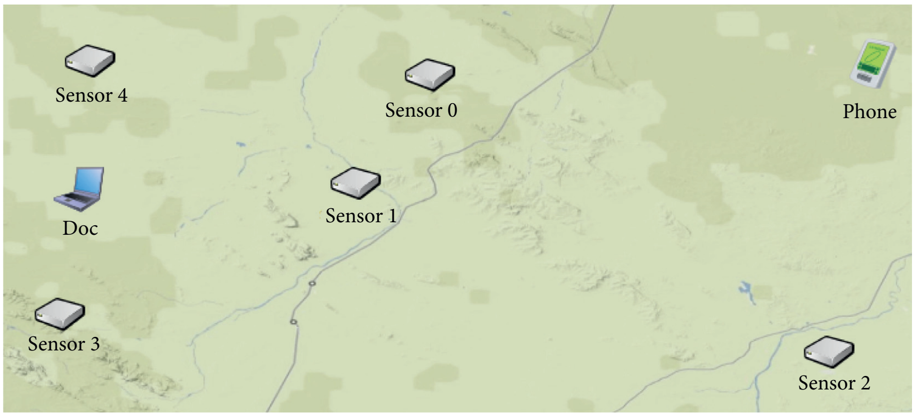


Figure 10: Network model.

4.1.2. Message Design. In OMNET++, messages are represented by the cMessage class and cPacket class, where cPacket is a subclass of cMessge. In this experiment, messages are mainly used for data transmission. Table 2 shows the message files used in the simulation process.

4.1.3. Statistical Analysis of Results. Table 3 shows the various statistical signals involved in the simulation process. After the simulation, we can get the statistical results of these signals, and we can evaluate the network performance according to these results.

4.2. Simulation Results and Analysis

4.2.1. Topology Use Case. Table 4 shows the settings of simulation parameters. In OMNET++, simulation parameters are set by the configuration file omnetpp.ini, including CPU running time, simulation time, and network topology usage type. When only the parameters set in the configuration file are changed without modifying other files in the project, new simulation results can be obtained without deploying the project. Among them, there are three kinds of network scenarios, including 5 sensor nodes, 10 sensor nodes, and 15 sensor nodes in the network.

4.2.2. Performance Evaluation

(1) Normal Performance Evaluation. Because the design of this paper is an authentication mechanism and the design goal is lightweight, then how to prove security and lightweight is the focus of our performance evaluation, which conforms to the characteristics of the integrated domain network resource constraints and high security requirements of BAN, and the performance evaluation of the authentication mechanism in the BAN should focus on resource overhead and security. For the evaluation of network performance, the end-to-end delay and packet loss rate are selected to verify the correctness of the parameter setting.

(2) Normal Performance Evaluation (End-to-End Delay). End-to-end delay refers to the average time of packets from the source node to destination node in BAN, which is calculated as $\mathrm{EED} = (\sum_{i=1}^{n}(T_{\text{receivei}} - T_{\text{sendi}})/n)$. Among them, T_{sendi} and T_{receivei} represent the sending time and receiving time of the packet i, respectively, and n represents the number of i packets. The delay time increases with the number of nodes. The delay time is 0.0168 ms when 5 sensor nodes, 0.0183 ms when 10 sensor nodes, and 0.0196 ms when 15 sensor nodes are set in BAN. The reason for this situation is that when the number of nodes increases, the number of information exchange in the network increases. Too much information exchange leads to network congestion, and the blocked network will naturally lead to the increase of information exchange time, that is, the end-to-end delay time.

(3) Normal Performance Evaluation (Packet Loss Rate). Packet loss rate refers to the ratio between the number of packets lost and the number of packets sent during the operation of the BAN, which is calculated as $\mathrm{PLR} = 1 - (N_{\text{receivepkt}}/N_{\text{totalpkt}}) \times 100\%$. $N_{\text{receivepkt}}$ and $N_{\text{totalpk}t}$ represent the number of received and sent packets, respectively. The packet loss rate increases with the number of nodes. When the number of sensor nodes in the BAN is 5, the packet loss rate is 2%. When the number of sensor nodes in the BAN is 10, the packet loss rate is 9%. When the number of sensor nodes in the BAN is 15, the packet loss rate is 14%. The reason for the above situation is that, with the increase of the number of nodes, the number of information exchange and transmission between nodes increases. A large quantity of information is transmitted in the network, and the delay time increases, resulting in the packet loss due to the long time of transmission to the receiving end or not to the receiving end. At the receiving end, when a large number of data packets are transmitted, the receiving end may not be able to process these data packets in time due to some restrictions or other reasons and may also have the phenomenon of packet loss.

(4) Normal Performance Evaluation (Storage Overhead). In the authentication protocol, in order to get different random numbers by using a pseudorandom generation sequence, we set up the initial seed and butterfly seed generation algorithm. In the butterfly seed generation algorithm, in order to get different seeds, a parameter is set, and the value of j needs to be saved in real time. Its type can be an integer, and the size of the integer is generally 2 byte, that is, 16 bits. In addition, it is necessary to save an initial seed and then transform the seed randomly, which is unpredictable. The size of the initial seed and control nodes is 640 bits. Both the receiver and the sender need to store this initial seed. The receiver has only one device, while the sender is multiple sensor nodes in the network. This value can be set to n. In this paper, the values of n are 5, 10, and 15, respectively. So, the storage overhead can be expressed as $640\,(\mathrm{n} + 1) + 16$.

(5) Normal Performance Evaluation (Communication Overhead). Communication overhead refers to the size of messages transmitted between two entities. Both sender and receiver need to send encryption and acknowledgment messages to the receiver. The size of the encrypted message is 640 bits, and the confirmation message is a string of authentication success or failure, with the size of 96 bits. In conclusion, the communication overhead is 1472 bits, in which both the communication overheads of control nodes and sensor nodes are 640 + 96 bits.

(6) Normal Performance Evaluation (Computational Overhead). In the authentication protocol, the computation of sensor nodes includes the execution of hardware PUF, XOR operation, encryption, Hamming distance, seed generation, and pseudorandom sequence generator. The total calculation time of these operations is about 0.28 ms. The calculation of the receiver includes XOR operation, decryption, seed generation algorithm, Hamming distance, and reading data from the cloud and pseudorandom sequence generator. The total calculation time of these operations is about 2.91 ms.

Table 2: Message files in the simulation.

Message files	Description
α.msg	Authentication message
β.msg	Authentication message
endRxEvent	Self-information
MessagePhysical.msg	Physiological information
MessageEmergency.msg	Emergency information
MessageAuth.msg	Confirmation message
MessageMerger.msg	Merged message

Table 3: Statistical signal.

Statistical signal	Function
endToEndDelaySignal	End-to-end delay
packetLossRate	Packet loss rate
pkCount	Packets received
pkNumber	Packets sent
pkTime	Packet transfer time
compTime	Computing time

Table 4: The setting simulation parameters.

Parameters	Values
Simulation time	300 s
Scene size	3 m∗3 m
Network scenarios	I:5 sensor nodes II:10 sensor nodes III:15 sensor nodes
Transmission speed	40 kbps
Interval time	Exponential 2 s
Packet size	640 bits
Delay time	10 ms

(7) Normal Performance Evaluation (Energy Cost). Ideally, the longer a medical sensor device is worn on or implanted into the human body, the better, so the lower the energy consumption, the better. When a 32-bit Cortex-M3, 72 MHZ microcontroller is active at 27°C, it requires 36 mA current and 36 V voltage, and the electric power is about 118.8 mW [30]. According to the above situation, the corresponding energy consumption can be calculated by reusing the calculation cost. Assuming that the computation cost is t(ms), the energy consumption is 118.8 t/1000. The calculation cost of this paper is about 3.19 ms, so the energy cost is 0.379 mJ. The energy cost of control nodes is 0.346 mJ, and the energy cost of sensor nodes is 0.033 mJ.

(8) Performance Evaluation in Emergency (ECG Signal Data Set). The ECG signal data set used in this experiment is from the PhysioBank database, which is a large scientific research resource database mainly based on ECG signals and supplemented by other data such as magnetic resonance imaging (MRI) [31]. The ECG database in PhysioBank records the physiological signals of healthy people and patients. Each person's record consists of three files, which are data file (also known as binary file, suffix is .dat), annotation file (suffix is .atr), and header file (suffix is .hea). PhysioNet provides a toolkit WFDB for developers to use in the development process. With WFDB, we can connect the data on PhysioNet with MATLAB development software so that we can get the data we want in MATLAB. Figure 11 is the ECG signal diagram obtained by executing the demo file after WFDB is configured successfully in MATLAB. It is the ECG signal diagram with the number 105.

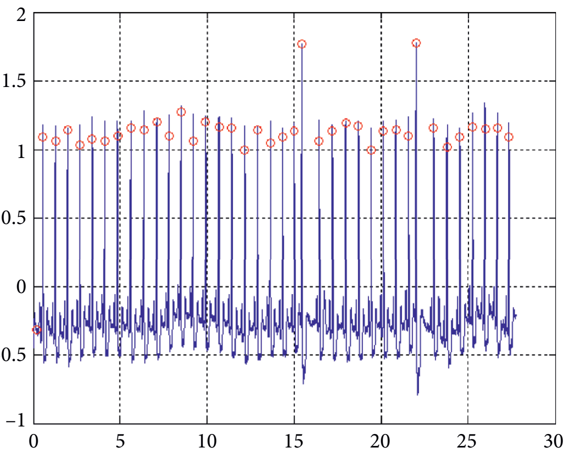
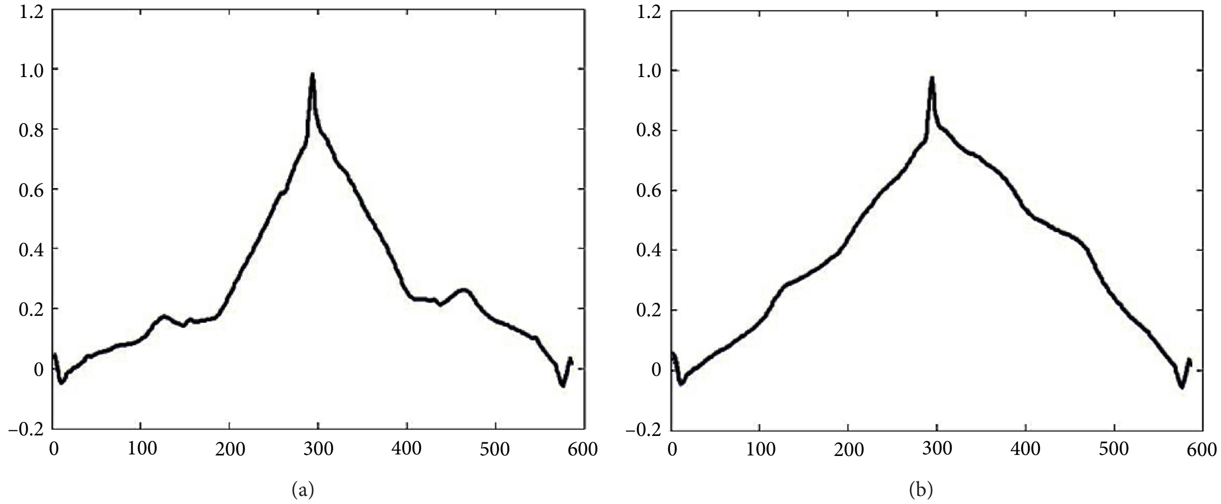


Figure 11: ECG signal of wfdbdemo.m.

(9) Performance Evaluation in Emergency (Performance Analysis). In this paper, the improved cross-correlation algorithm is implemented in MATLAB; there are two methods to use, one is to use the function expression of the data set, the other is to use the specific value of the data set, and then form the vector sequence according to the specific value. Since there is no fixed function expression for the ECG signal of the human body, this paper chooses the second method. Through the calculation of the cross-correlation algorithm and normalization operation, we hope to get the required cross-correlation coefficient, that is, the value of ordinate in the graph, and the maximum ordinate value in the graph is the result we want. The results obtained by the cross-correlation algorithm are shown in Figures 12–14.

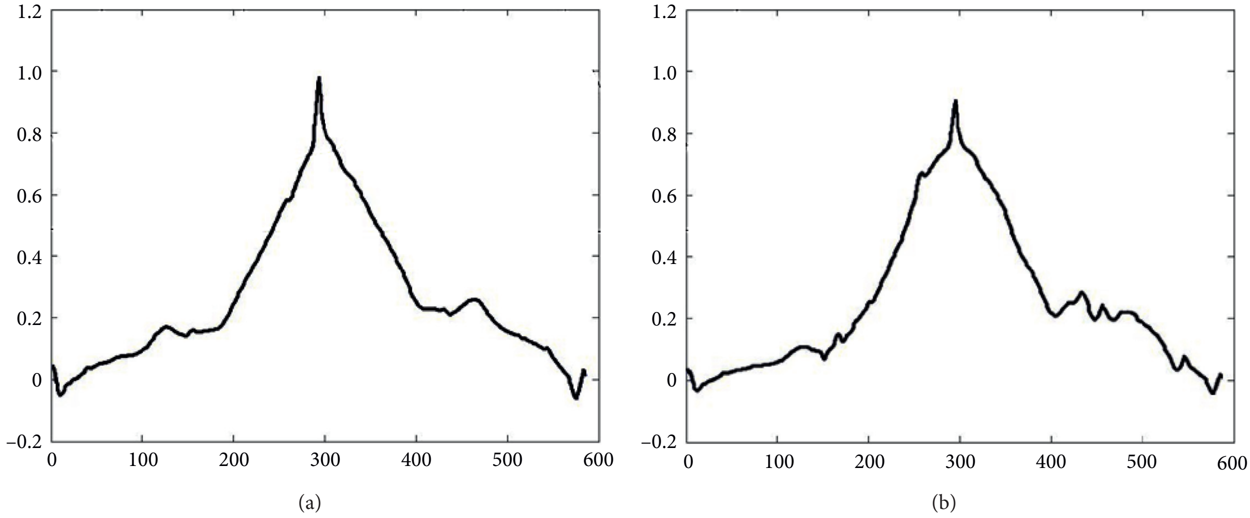
The ECG signals selected in Figure 12 are the data from the subject 100 on the website, which shows the correlation number of the subject 100 in two different periods. It can be seen from the figure that the cross-correlation coefficient is closer to 1 after the weighted processing of the ECG signal, and the result is the same as expected, which also achieves the purpose of improving the cross-correlation algorithm and improving the accuracy of identification.

The ECG signals selected in Figure 13 are normal and abnormal data from the subject 100 on the website. It can be seen from the figure that there is a certain distance between the maximum value of the normal ECG signal and the abnormal ECG signal obtained by the improved cross-correlation algorithm, so this method can easily identify the abnormal ECG signal. When people's body is suddenly abnormal, it can be relatively easy and accurate to judge the abnormal situation and make the correct response to the abnormal situation so that the human body can be treated in time.



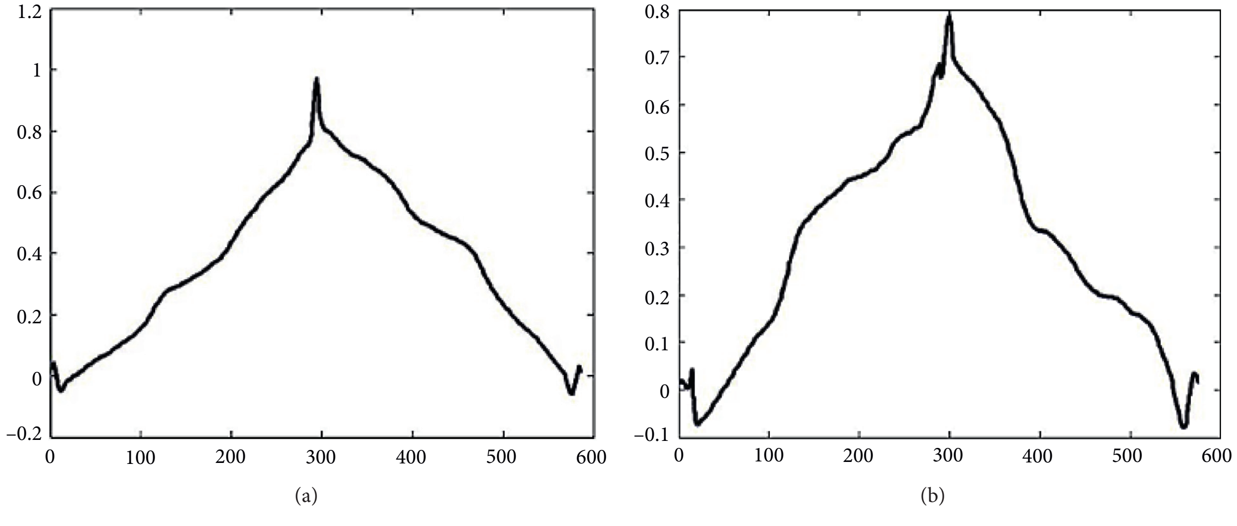
(a) (b)

Figure 12: Comparison figure of weighted 100 and 100 and unweighted 100 and 100.



(a) (b)

Figure 13: Comparison figure of weighted 100 and 100 and weighted 100 and exception 100.



(a) (b)

Figure 14: Comparison figure of weighted 100 and 100 and weighted 100 and 105.

The selected ECG signals in Figure 14 are ECG data from subjects 100 and 105 on the website. It can be seen from the figure that if an attacker uses another person's ECG signal to impersonate the party's ECG signal and attempts to muddle through and destroy the normal operation of the authentication mechanism, it is obviously not feasible because it is easy to be found.

4.2.3. Benchmark Mechanism. One of the benchmark mechanisms selected in this paper is the effective anonymous authentication mechanism based on the elliptic encryption algorithm (ECC-based) [32]. This mechanism ensures the security of BAN by improving the traditional security methods. It is a classic scheme to solve the security problems of BAN by using the traditional security methods and has certain representativeness in solving the security problems of BAN.

In this mechanism, there are three roles: control node (client), third-party entity (nm), and sensor node (AP). The sensor node collects and sends the physiological data of the human body; the control node can obtain the collected physiological data and send it to the doctor for treatment; the main task of the third-party entity is to generate the required private key.

The mechanism is divided into three phases: initialization phase, registration phase, and authentication phase. In the initialization phase, the third party is responsible for generating the required system parameters. In the registration stage, the control node and the third party establish a legal relationship through certain measures so that the control node becomes a legal node. When the control node operates again next time, it can be known that it is a legal node that has passed the authentication. In the authentication stage, the control node can obtain the service it needs from the common node after passing the authentication. The running process of the ECC mechanism is shown in Figure 15.

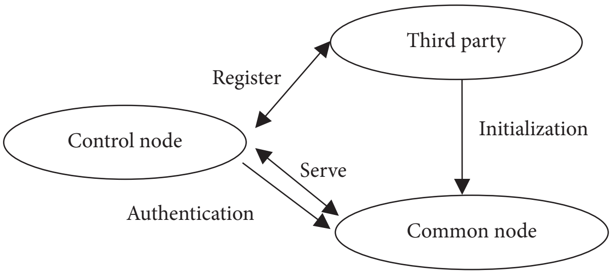


Figure 15: The operation process of the ECC mechanism.

Another benchmark mechanism selected in this paper is based on simple cryptographic primitives (HASH-BASED) [30]. The mechanism is divided into three stages: initialization stage, authentication and key sharing stage, and joining sensor node stage. In the initialization stage, before the deployment of sensor nodes and control nodes, the third party will perform some operations, which include generating a master key that can be used by the control node for a long time and assigning a unique identification number to the sensor node. In the stage of authentication and key sharing, the sensor node and the control node can judge whether the other party is a trusted entity by using the designed mechanism, and the public session key established in this process is saved by both parties for safe use in future communication. In the dynamic joining stage, the third-party entity assigns a unique identification number to the new sensor node, calculates its key, checks the vector value, stores tuples, and then deploys it to the corresponding location and notifies the control node.

4.2.4. Performance Comparison. Through the performance analysis of the authentication mechanism in an emergency, we can see that the authentication mechanism designed in this paper can make a good response when the emergency needs to be handled. The performance of the improved cross-correlation algorithm is better, the cross-correlation coefficient calculated by the improved algorithm is more accurate, and the probability of misjudgment is reduced. Most of the previous studies do not consider the emergency, but the design of this paper comprehensively considers each situation so that it can make different responses to different situations so that the BAN can play a better role.

By comparing the performance of PUF-BASED in terms of resource consumption and security with the benchmark mechanisms, Table 5 shows the communication overhead, computational overhead, and energy consumption of PUF-BASED, HASH-BASED, and ECC-BASED.

Table 6 shows the different security categories that can be guaranteed by the design scheme and comparison mechanism. It can be seen from the table that although the design schemes are different, they can resist certain security attacks. However, the security categories guaranteed by different schemes are different. It cannot simply indicate which scheme has better security performance. It can only be said that the scheme in this paper can resist a certain degree of attacks, and the security performance is guaranteed. It can be seen from the table that the scheme in this paper can resist eavesdropping tampering attack, replay attack, middleman attack, and simulation attack and has forward/backward security. The ECC-BASED authentication scheme can resist eavesdropping tampering attack, replay attack, and middleman attack and has elastic recovery ability. The HASH-BASED authentication scheme can resist eavesdropping tampering attack, replay attack, middleman attack, and simulation attack and has forward/backward security and elastic recovery ability.

From the above analysis, it can be seen that the mechanism of this paper has certain advantages over the improved traditional security methods in terms of both resource consumption and security performance. In this paper, the design does not use symmetric encryption or asymmetric encryption method; nor does it make use of the characteristics of large numerical value, large quantity, and difficult calculation to ensure security, such as the elliptic curve encryption algorithm takes advantage of the difficulty of numerical calculation; and, there is no particularly tedious calculation, so the effect of resource consumption is better. In addition, the authentication mechanism designed in this paper comprehensively considers the security problems in various situations, so it has good security performance.

Table 5: Resource consumption of different authentication mechanisms.

		Communication overhead (bits)	Computational overhead (ms)	Energy consumption (mJ)
PUF-BASED	Control nodes	1472	2.91	0.379
	Sensor nodes		0.28	
HASH-BASED	Control nodes	1120	0.48	0.093
	Sensor nodes		0.3	
ECC-BASED	Control nodes	1856	26.46	4.73
	Sensor nodes		13.35	

Table 6: Safety performance of different authentication mechanisms.

	Eavesdropping tampering attack	Replay attack	Forward/ backward security	Nonclonability	Middleman attack	Simulation attack	Elastic recovery ability
PUF-BASED	✓	✓	✓	✓	✓	✓	—
HASH-BASED	✓	✓	✓	—	✓	✓	✓
ECC-BASED	✓	✓	—	—	✓	—	✓

However, the mechanism designed in this paper is not as good as the mechanism designed with simple cryptographic primitives in terms of total computing cost and energy consumption. This is because this paper introduces the cloud database to store a large amount of data, so as to reduce the consumption of local resources. Therefore, there are operations of reading and writing data, which will increase the computing cost of the control node and then increase the energy consumption. However, the computational cost of the mechanism designed in this paper is slightly lower than that of the authentication mechanism designed with cryptographic primitives, so it can reduce the resource consumption of the sensor node and prolong its service life. In terms of security performance, the mechanism designed in this paper is similar to that designed with cryptographic primitives.

In the process of its implementation, there may be the following challenges and limitations. In terms of possible challenges, firstly, this paper mentions that, in an emergency, a certain characteristic signal of each individual is used as a way to judge whether it is safe. Because there is no personal data in the early stage or there may be a lack of data due to various reasons such as the network, the use in the early stage and when the network is poor may not reach the ideal state. Second, it is mentioned in the article that the cloud database is used to store a large amount of data, and the data should be kept confidential. At present, this kind of cloud database needs to pay, so how to effectively reduce the cost is a certain challenge. In terms of possible limitations, one is that the network topology of this scheme is limited to star topology, and other topologies have not been considered. Although star topology is widely used at present, other topologies have been studied in some papers. Second, the current emergency use of the ECG signal as a method of using instructions; in real cases, there will be other physiological characteristics of data; for different physiological characteristics of data, we can further select its data feature points as function parameters.

5. Conclusions

By analyzing the communication model, security and performance requirements, as well as various existing authentication methods, this paper explains the importance of a lightweight authentication mechanism for BAN. As for the design of the authentication mechanism under normal conditions, according to the characteristics of PUF, the mutual authentication mechanism between the sensor node and control node is designed by using the challenge response generated by the function. In case of emergency, patients need timely treatment, and the demand for timely treatment is much higher than safety. Given this situation, the improved cross-correlation algorithm is used to judge whether the human body is in an emergency; if so, broadcast the data to get timely and effective treatment. Different data access methods are designed according to the security level of medical sensor devices, and the designed authentication mechanism is simulated by using the OMNET++ simulation platform, and the results are compared and analyzed with the comparison mechanism. The experimental results show that the authentication mechanism designed in this paper has good effects in four aspects: computing cost, communication cost, energy consumption, and security.

The lightweight identity authentication mechanism proposed in this paper mainly includes two parts: one is to design the authentication mechanism under normal circumstances; the other is to design the authentication mechanism in an emergency, and according to the authentication, the information access method based on the node level is designed, but there are still some shortcomings. This paper uses the mature and representative star topology. This topology can be further studied in the future. In the aspect of simulation, the simulation of BAN should use real sensors with a simulation platform to achieve so that the simulation results will be more accurate. In the evaluation of security and resource consumption, this paper analyses these two aspects separately and then compares them with other methods. BAN requires high security performance and low resource consumption, so as to achieve a balance between them. In the future, we will further consider the comprehensive analysis of security and resource consumption.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the Beijing Social Science Foundation Key Project (18GLA009), Beijing The Great Wall Scholars' Program (no. CIT & TCD20170317), and Beijing Intelligent Logistics System Collaborative Innovation Center Open Topic (no. BILSCIC-2019KF-03) and funded by the Graduate Science and Technology Innovation Project of Capital University of Economics and Business.

References

[1] P. Kumar and H. J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: a survey," *Sensors*, vol. 12, no. 1, pp. 55–91, 2012.

[2] J. Sametinger, J. Rozenblit, R. Lysecky, and P. Ott, "Security challenges for medical devices," *Communications of the ACM*, vol. 58, no. 4, pp. 74–82, 2015.

[3] E. Jovanov, D. Raskovic, J. Price et al., "Patient monitoring using personal area networks of wireless intelligent sensor," *Biomedical Sciences Instrumentation*, vol. 37, pp. 373–378, 2001.

[4] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. M. Leung, "Body area networks: a survey," *Mobile Networks and Applications*, vol. 16, no. 2, pp. 171–193, 2011.

[5] M. Rushanan, A. D. Rubin, D. F. Kune et al., "SoK: security and privacy in implantable medical devices and body area networks," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, pp. 524–539, Berkeley, CA, USA, May 2014.

[6] S. Faye, N. Louveton, G. Gheorghe et al., "A two-level approach to characterizing human activities from wearable sensor data," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 7, no. 3, pp. 1–21, 2016.

[7] M. Kumar, "Security issues and privacy concerns in the implementation of wireless body area network," *Food Policy*, vol. 34, no. 3, pp. 245–251, 2015.

[8] S. Aram, R. A. Shirvani, E. G. Pasero et al., "Implantable medical devices; networking security survey," *Journal of Internet Services and Information Security*, vol. 6, no. 3, pp. 40–60, 2016.

[9] A. Singla and R. Sachdeva, "Review on security issues and attacks in wireless sensor networks," *International Journal of Future Generation Communication & Networking*, vol. 8, no. 4, pp. 81–88, 2015.

[10] C. Gritti, M. Onen, R. Molva et al., "Device identification and personal data attestation in networks," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 9, no. 4, pp. 1–25, 2018.

[11] N. Nguyen, C. A. Y. Kaya, A. Bru¨Sch et al., "Demo of BANDANA-body area network device-to-device authentication using natural gAit," in *Proceedings of the 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 421–423, San Jose, CA, USA, July 2018.

[12] H. Anada, "Decentralized multi-authority anonymous authentication for global identities with non-interactive proofs," *Journal of Internet Services and Information Security*, vol. 10, no. 4, pp. 23–27, 2020.

[13] E. M. Torroglosa-Garcia and A. F. Skarmeta-Gomez, "Towards interoperabilty in identity federation systems," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 8, no. 2, pp. 19–43, 2017.

[14] T. Denning, K. Fu, and T. Kohno, "Absence makes the heart grow fonder: new directions for implantable medical device security," in *Proceedings of the USENIX Workshop on Hot Topics in Security*, San Jose, CA, USA, July 2008.

[15] D. Halperin, T. S. Heydt-Benjamin, B. Ransford et al., "Pacemakers and implantable cardiac defibrillators: software radio attacks and zero-power defenses," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy (Sp 2008)*, pp. 129–142, Oakland, CA, USA, May 2008.

[16] Y. S. Lee, H. J. Lee, and E. Alasaarela, "Mutual authentication in wireless body sensor networks (WBSN) based on physical unclonable function (PUF)," in *Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1314–1318, 2013.

[17] B. Ovilla-Martinez, A. Díaz-Pérez, and J. J. Garza-Saldaña, "Lightweight mutual authentication among sensors in body area networks through physical unclonable functions," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, May2017.

[18] P. Steffen, P. R. Bhanu, M. Farshad et al., "Design of secure ECG-based biometric authentication in body area sensor networks," *Sensors*, vol. 16, no. 4, p. 570, 2016.

[19] C. Hu, X. Cheng, F. Zhang et al., "OPFKA: secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," *2013 Proceedings IEEE INFOCOM*, pp. 2274–2282, 2013.

[20] M. Rostami, A. Juels, and F. Koushanfar, "Heart-to-heart (H2H): authentication for implanted medical devices," in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 1099–1112, Berlin, Germany, November 2013.

[21] J. Wan, C. Zou, S. Ullah, C.-F. Lai, M. Zhou, and X. Wang, "Cloud-enabled wireless body area networks for pervasive healthcare," *IEEE Network*, vol. 27, no. 5, pp. 56–61, 2013.

[22] H. Liu, Y. Chen, H. Liu et al., "An efficient cloud-assisted message authentication scheme in wireless body area network," *International Journal of Security and Its Applications*, vol. 11, no. 3, pp. 71–80, 2017.

[23] C.-M. Yu, C.-Y. Chen, and H.-C. Chao, "Verifiable, privacy-assured, and accurate signal collection for cloud-assisted wireless sensor networks," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 48–53, 2015.

[24] R. Sampangi and S. Sampalli, "Butterfly encryption scheme for resource-constrained wireless networks," *Sensors*, vol. 15, no. 9, pp. 23145–23167, 2015.

[25] E. Nigussie, T. Xu, and M. Potkonjak, "Securing wireless body sensor networks using bijective function-based hardware

primitive," *IEEE Tenth International Conference on Intelligent Sensors*, pp. 1–6, 2015.

[26] C. Liu, P. Cronin, and C. Yang, "A mutual auditing framework to protect IoT against hardware Trojans," in *Proceedings of the Asia & South Pacific Design Automation Conference*, pp. 69–74, Macao, Macao, January 2016.

[27] C. Gritti, W. Susilo, and T. Plantard, "Certificate-based encryption with keyword search: enabling secure authorization in electronic health record," *Journal of Internet Services and Information Security*, vol. 6, no. 4, pp. 1–34, 2016.

[28] "Medical device-Classify your medical device," *Food and Drug Administration*, http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/Overview/ClassifyYourDevice/default.htm, 2020.

[29] J. Y. Wang, Y. H. Wei, H. B. Qiu et al., *Network Simulator OMNeT++*, Xidian University Press, Xi'an, China, 2014.

[30] M. H. Ibrahim, S. Kumari, A. K. Das et al., "Secure anonymous mutual authentication for star two-tier wireless body area networks," *Computer Methods & Programs in Biomedicine*, vol. 135, p. 37, C, 2016.

[31] A. L. Goldberger, L. A. N. Amaral, L. Glass et al., "PhysioBank, PhysioToolkit, and PhysioNet," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[32] Z. Zhao, "An efficient anonymous authentication scheme for wireless body area networks using elliptic curve cryptosystem," *Journal of Medical Systems*, vol. 38, no. 2, pp. 1–7, 2014.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5593178, 11 pages
https://doi.org/10.1155/2021/5593178

Research Article

A Mobile Malware Detection Method Based on Malicious Subgraphs Mining

Yao Du,[1] Mengtian Cui,[1] and Xiaochun Cheng[2]

[1]*Key Laboratory of Computer System, State Ethnic Affairs Commission, Southwest Minzu University, Chengdu 610041, China*
[2]*Department of Computer Science, Middlesex University, London NW44BE, UK*

Correspondence should be addressed to Mengtian Cui; hangkongzy@163.com

Received 6 February 2021; Revised 10 March 2021; Accepted 7 April 2021; Published 19 April 2021

Academic Editor: Zhe-Li Liu

As mobile phone is widely used in social network communication, it attracts numerous malicious attacks, which seriously threaten users' personal privacy and data security. To improve the resilience to attack technologies, structural information analysis has been widely applied in mobile malware detection. However, the rapid improvement of mobile applications has brought an impressive growth of their internal structure in scale and attack technologies. It makes the timely analysis of structural information and malicious feature generation a heavy burden. In this paper, we propose a new Android malware identification approach based on malicious subgraph mining to improve the detection performance of large-scale graph structure analysis. Firstly, function call graphs (FCGs), sensitive permissions, and application programming interfaces (APIs) are generated from the decompiled files of malware. Secondly, two kinds of malicious subgraphs are generated from malware's decompiled files and put into the feature set. At last, test applications' safety can be automatically identified and classified into malware families by matching their FCGs with malicious structural features. To evaluate our approach, a dataset of 11,520 malware and benign applications is established. Experimental results indicate that our approach has better performance than three previous works and Androguard.

1. Introduction

Mobile device has become an essential social network communication tool which stores a huge amount of user privacy data. Therefore, it attracts persistent malicious attacks. Due to the open source policy, Android has become the most popular operating system for mobile devices and has the largest market share. With the widespread use of Android applications, Google company is committed to maintain the safety of its official application market—Google Play Store [1]. Aiming at security problems, Google has used various strategies to fight against malicious attacks such as regularly scanning billions of installed mobile applications, providing remote security services for the mobile device, and isolating malicious websites to protect users. However, Google's large investment only blocks some of the malware that threatens Google Play Store. Many third-party application markets are still facing an increasing numbers of malware. A report [2] showed that 97% of the total number of mobile malware was related to Android platform in 2013, up from 79% during the previous year and 66% in 2011. In 2015, the total number of Android malware rose to 884,774 [3]. In 2019, Android malware variants grew 31% in a year and the total number closed to 20 million [4].

A commonly used detection strategy adopted by commercial antivirus tools (such as Norton and Lookout) is collecting as many as possible malware and extracting signature code as features [5, 6]. Then, these features are used to match with the signature code that is extracted from target applications to identify malware. Although this strategy can achieve high detection accuracy and low false positive rate (FPR), it still faces two challenges: (1) lagging behind malicious attacks and cannot detect unknown malware and (2) minor changes of applications may lead to failure of the detection method.

To solve the problems, researchers begin to use more effective expert features to recognize malicious code, such as permissions, component information, and APIs [7–9]. Many of them also apply machine learning technology to improve the detection performance further. Although these approaches have been proved effective, the grammatical feature-based detection methods are easily affected by code obfuscation and injection technologies.

Fortunately, many research works have [10–12] shown that high-level properties of code, especially structural features, can promote the resilience ability of repackaging and code obfuscation technologies. The most commonly used method is to compare the application's structural features with existing malicious ones. Thus, it is a well-known strategy to transform a graph matching problems into an isomorphism. However, the isomorphism problem has been proved to be a nondeterministic polynomial (NP) problem and may be very inefficient when the graphs are large. How to find malicious subgraphs efficiently is still a problem to be solved.

In this paper, we propose a new malware detection method based on malicious subgraph searching. The subgraphs are generated from FCGs of Android applications. During the training phase, the FCGs are generated from the malware of each malware family to get malicious structural features. It can improve the detection efficiency and help to analyze the homology and evolution of variant viruses. In the test phase, structural features are used to automatically detect malware and classify them into different families. In the evaluation phase, several experiments are designed to evaluate the detection performance and execution efficiency of our method.

According to the above descriptions, the main contributions of our method can be summarized as follows:

(1) A new efficient Android malware detection method is proposed. In this method, the malicious features extracted from applications are structural. It has a positive effect on the resilience to code obfuscation and repackaging technologies in static analysis. At the same time, the method can find a class of similar variant samples that could be useful for malware detection and new variant analysis.

(2) A fast common subgraph searching and matching algorithm based on nodes similarity calculation is designed. There are several well-known algorithms that can match subgraphs by using graph isomorphism calculation, such as VF2 [13] and graph edit distance algorithms [14]. However, it is a big challenge for them to successfully match a large number of subgraphs with complex structure and large scale. In this situation, our method has much better performance compared with the VF2 algorithm and so on.

(3) The evaluation processes of our method are executed based on several datasets of 7520 malware and 4000 benign Android applications. Evaluation results indicate that our method has better detection ability than three previous works and Androguard.

The following parts of this paper are organized as follows. The relevant research studies of our topic are collected in Section 2. Section 3 introduces the whole architecture of our method. The specific execution process of our method is discussed in Section 4. Section 5 applies some experiments to test the detection efficiency of our method. At last, the conclusion and future works are discussed in Section 6.

2. Related Works

Structural analysis has been widely used in Android malware detection methods. It can be divided into the following two main categories.

2.1. Static Structural Feature-Based Detection Methods. In the analysis of structural features, many research studies have been done to solve the NP problem caused by isomorphism algorithms. At the same time, new methods were designed to find malicious structural features effectively.

Crussell et al. [15] proposed a new method to identify the replication and clone behaviors of mobile applications called "DNADroid." This method constructed function call graphs of applications at first. Then, the similarity values of function call graphs were calculated based on the VF2 algorithm. At last, these similarity values were used to evaluate the similarity of applications. In the experiment part, DNADroid was used to test the applications of their experimental dataset. It found that at least 114 applications had been cloned.

Xu et al. [16] proposed a malicious code detection method based on function call graphs. This method firstly extracted function call graphs from mobile applications. Secondly, graph edit distances were calculated based on applications' methods and methods' operation code. Finally, the similarity measurement of applications can be got according to the graph edit distance score. Experimental results showed that this method can identify variants of malware.

Zhang et al. [11] proposed a malicious code detection method based on semantics information called "DroidSIFT." The method can extract function dependency graphs from mobile applications by implementing a graph generation tool on top of Soot [17]. Then, feature vector spaces were constructed according to the similarity of function dependency graphs. At last, these feature vector spaces were used to build two different classifiers to identify malicious code. In their experiment, DroidSIFT was used to test 13500 benign samples and 2200 malware. Experimental results showed that DroidSIFT could achieve 93% detection accuracy and 5.15% false positive rate (FPR).

Suarez-Tangil et al. [18] proposed a new malware detection method which can automatically extract code chunks (CCs) from the control flow graph of Android applications. Then, the CCs were used to be analyzed by the text mining model to classify malware into different families.

Hu et al. [19] proposed a static malware detection method called "MIGDroid." The method extracted invocation graphs from Android applications and then divided them into subgraphs. By calculating threat scores of subgraphs, malicious code of applications can be found.

Niu et al. [20] built the opcode-level FCG of Android applications and used the long short-term memory model to analyze malicious behaviors. The detection accuracy of the method is 97% based on their dataset. Gao et al. [21] used the invocator-invocator relationship of Android application's FCG to generate topological signatures. The malware detection method based on the topological signatures was designed and evaluated by 1249 malware and 49000 benign applications.

Sun et al. [22] proposed a new graph-based Android malware detection method called "DroidSim." DroidSim can construct component-based control flow graph (CB-CFG) by using APIs as nodes and control flow precedence order of Android components as edges. The evaluate similarity scores of CB-CFGs were used for malware detection. In their experiment, DroidSim achieved 96.6% detection accuracy on 121 benign Android applications and 706 malware.

Atici et al. [23] extracted static features from control flow graph-based and used machine leaning classification to identify malware.

2.2. Dynamic Structural Feature-Based Detection Methods. John et al. [24] extracted the system call graph of the Android application and used graph convolutional nets to detect malware. The detection accuracy of their method is 92.3% on the experimental datasets with 2130 samples.

Zhang et al. [25] extracted the information of object reference graphs from the execution processes of Android applications to build an object reference information model. Then, a two-step malware detection method was designed based on the improved graph isomorphism algorithm.

Abdurrahman and Acarman [26] constructed API call graphs of Android applications and then transformed them into low-dimensional feature vectors. Finally, a deep neural network-based method was designed to detect malware.

Yerima et al. [27] researched the generation process of stateful events and designed a new method to improve the code coverage in dynamic analysis for malware detection.

Lin et al. [28] extracted features from behavior dependency graphs for machine learning classification process. Then, a prototype system was implemented to identify malware.

Xu et al. [29] proposed a new efficient representation of system call graph. Then, feature vector labels of the representations were used and optimized to improve the classification ability of the SVM algorithm.

Hou et al. [30] extracted dynamic behavior features from weighted directed graphs. Then, a deep learning model was applied to identify malware based on these features.

Although both static and dynamic methods can analyze structure features, our method is proposed based on static detection for the following two reasons: (1) static detection can get a complete function call graph without missing malicious subgraphs and (2) static detection does not need a virtual running environment, which means that it can detect a large number of viruses faster than dynamic detection. Differently from the above research studies, our method discards analyzing and matching large-scale graph structures directly. A new method which can iteratively construct structural malicious features from a single node is designed. It can quickly locate the malicious code generated by code injection technology and identify variant viruses effectively.

3. Architecture Overview

The architecture of our malware detection method is depicted in Figure 1. The main steps are as follows:

(1) *Decompile Process*. The "AndroidManifest.xml" and ".dex" files of the Android applications are used in our method. Thus, Android applications whose executable file named Android package (APK) files are decompiled by Androguard [31] to obtain these files.

(2) *Static Information Collection*. Several Python script files are implemented on top of the Androguard tool. They are used to collect static information from the disassembled code files of step (1), including Android applications' methods, method call sequences, permissions, and APIs. Then, the FCG is constructed based on the methods' information. All these static information is output to text files in a uniform format, respectively.

(3) *Structural Features Generation*. By analyzing the permissions of malware and benign applications, we choose the permissions that are used more frequently in malware as sensitive permissions. Then, two kinds of malicious structural features are constructed. The first one is a sensitive permission-based subgraph. It extracts sensitive permissions' relative APIs as initial nodes. The method call sequences which contain these initial nodes are used to construct structural features. The second one is the common subgraph of malware families. A nodes similarity-based subgraph searching method is designed. The method firstly searches the most similar node of the graphs, and then its similar adjacent nodes are searched based on the improved Kuhn–Munkres (KM) algorithm [32]. This process will be iteratively executed until the termination condition (introduced in Subsection 4.3) is met. To improve the execution efficiency, the maximum path length that contains the initial node is 3. Once all structural features are generated, they are put together as a malicious feature set.

(4) *Malware Detection*. The application to be tested should be decompiled and generate its static information as step (1) and (2). The nodes similarity-based matching process is also implemented between the application's FCG and the feature set to identify malware.

4. Malicious Subgraphs Generation

This section introduces the details of the FCG and malicious subgraph feature generation processes as well as the nodes similarity-based matching method and improved KM algorithm.

4.1. FCG Generation. An Android application implements its operations based on methods and method call sequences. Thus, the function call graph contains the affluent behavior

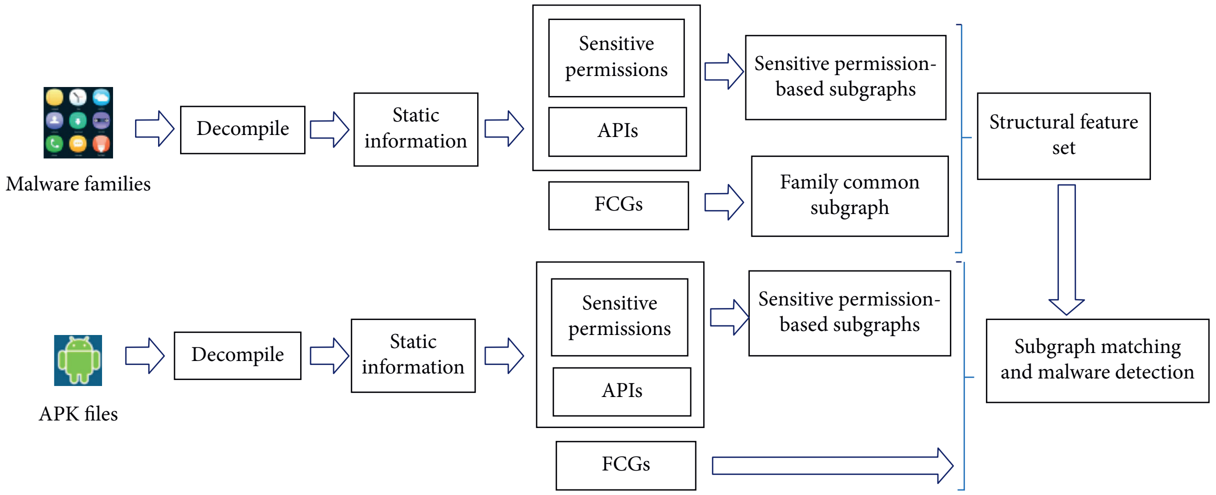


Figure 1: The architecture of our method.

information of an application. As mentioned in the previous section, our method constructs the function call graph by implementing a Python script file on top of the Androguard. The Python script file needs to exhaustively search all methods which are likely to be ignored by indirect calls.

The first step is to get an Android application's packages, classes, and methods' information from different objects of decompiled files. In this process, Androguard will automatically assign a numeric label to each method. These numeric tags are fixed, which means they are not changed with multiple executions of decompiling operations. The second step is to search all methods and store them as a graph's nodes in a node set. The third step is to apply a depth-first searching process to find nodes' all related nodes and call relations. To finish this work, two more node sets are established to store nodes' parent nodes and child nodes and then continuously search the parent nodes of every node in the parent node set, as well as the child nodes of every node of the child node set. When all parent nodes and child nodes are searched, all call sequences of a node are collected. At last, all call sequences are joined based on the numeric labels of nodes to construct the complete function call graph.

4.2. *Sensitive Permission-Based Subgraph Extraction.*

Android has set up a permission mechanism to control the access behaviors of applications. It can limit the excessive abuse of user privacy information and system resources by application developers. For example, if a program wants to obtain the information of Wireless Fidelity (WiFi) network status, its request should be written in the "AndroidManifest.xml" file, as follows:

<uses-permission android: name = "android.permission.ACCESS_WIFI_STATE" ></uses-permission>

Permissions can be used both by malware and benign applications. Therefore, a malicious score is assigned to each permission by calculating the frequency of occurrence of a permission between malware and benign applications. The top 20 permissions with the highest malicious score are treated as sensitive permissions in our method. At the same time, the APIs which are related to the sensitive permissions can be found, as shown in Figures 2 and 3.

In the framework provided by Android, a lot of drivers and functions are encapsulated in the bottom layer. Users can invoke these drivers by calling the APIs in their user-defined methods. Thus, the method call sequences which contain these user-defined methods can be found and used to construct sensitive permission-based subgraphs.

4.3. *Generation of Common Subgraph of Malware Families.*

In our method, the common subgraph is generated based on the nodes similarity calculation process. Considering the different definitions of similarity, there are different computational processes to get the value of similarity, for example, the use of graph isomorphism algorithms to evaluate the similarity based on the graph's structural features or calculate the similar distances of string features which are extracted from structural information and so on.

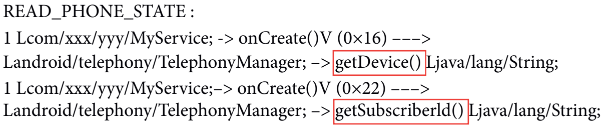
In Android malware detection, a large number of malicious variants are generated by repackaging technologies. Although many of them also have been processed by obfuscation technologies, they still have partial similarities in internal structures. In addition, code injection technologies can inj[[parms resize(1),pos(50,50),size(200,200),bgcol(156)]]ct malicious code into various kinds of benign applications. It makes completely unrelated applications implement similar attacks. This situation is particularly evident in the variants of the same malware family.

To find malicious code of applications, the nodes similarity calculation process is proposed.

Suppose that a graph is defined as $G = (V, E)$, where V represents the node set and E represents the edge set. If there are two graphs G_A and G_B, node $v_a \in G_A$ and node $v_b \in V_B$. v_{ai} is the ith adjacency node of v_a, and v_{bi} is the ith adjacency node of v_b. The similarity Sim_{ab} of two nodes can be defined as follows:

```
['android.permission.WAKE_LOCK','android.permission.WRITE_APN_SETTINGS',
'android.permission.RECEIVE_BOOT_COMPLETED',
'android.permission.ACCESS_NETWORK_STATE','android.permission.READ_PHONE_STATE',
'android.permission.WRITE_EXTERNAL_STORAGE','android.permission.INTERNET',
'android.permission.MODIFY_PHONE_STATE']
```

Figure 2: The permissions of "6a0bfabcc1cce2a5424313b34ca967fbc8f98bea.apk."



```
READ_PHONE_STATE :
1 Lcom/xxx/yyy/MyService; -> onCreate()V (0×16) --->
Landroid/telephony/TelephonyManager; -> getDevice() Ljava/lang/String;
1 Lcom/xxx/yyy/MyService;-> onCreate()V (0×22) --->
Landroid/telephony/TelephonyManager; -> getSubscriberId() Ljava/lang/String;
```

Figure 3: The APIs related to READ_PHONE_STATE permission of "6a0bfabcc1cce2a5424313b34ca967fbc8f98bea.apk."

$$\Delta w(v_i) = \alpha \cdot \frac{\left|D_a^{\text{in}} - D_b^{\text{in}}\right|}{D_a^{in} + D_b^{in}} + \beta \cdot \frac{\left|D_a^{\text{out}} - D_b^{\text{out}}\right|}{D_a^{\text{out}} + D_b^{\text{out}}},$$
$$E_{ab} = \max\left(S\left(V_A, V_B\right)\right), \tag{1}$$
$$Sim_{ab} = \left(\Delta w\left(v_i\right), E_{ab}\right),$$

where D_a^{in} and D_b^{in} are the in-degree of nodes v_a and v_b, D_a^{out} and D_b^{out} are the out-degree of nodes v_a and v_b, $\Delta w(v_i)$ is the node similarity value between v_a and v_b, $S(V_A, V_B)$ is the similarity value between the adjacency nodes of V_A and V_B, and E_{ab} is the optimal value of $S(V_A, V_B)$. The calculation process of E_{ab} will be introduced in Subsection 4.4 in detail. α and β whose values between (0, 1) are constant coefficients are used to optimize the matching efficiency which can be determined in the training phase.

According to the above definitions, our common subgraph searching method is designed as follows:

(1) Search different function call graphs of malware in the same family, and find the most similar nodes as initial nodes of the structural malicious feature.

(2) Continue to search initial nodes' the most similar adjacency nodes, and then add new-found nodes and edges to the existing subgraphs. This process will be iteratively executed once the values of $1/\Delta w(v_i)$ or E_{ab} are less than the thresholds whose range is (0.9, 1). When the iterative process has been stopped, the structural feature of a malware family is found.

4.4. Optimal Matching Strategy of Nodes. As mentioned in the previous subsection, a node probably has many adjacency nodes to be matched. It makes the calculation of E_{ab} a multimatching problem. To find the optimal matching result, a new matching method is designed.

The first step is to take a pretreatment. During the subgraph searching process, once $1/\Delta w(v_i)$ is less than the threshold, our method will continue to calculate the value of E_{ab}. It is necessary to note that there are many adjacency nodes which are leaf nodes. To reduce computational complexity, the similarity value ΔL of these leaf nodes is calculated as follows:

$$\Delta L = \frac{|x - y|}{x + y}, \tag{2}$$

where x is the number of leaf nodes of node v_a and y is the number of leaf nodes of node v_b. If ΔL is less than the threshold whose range is (0.9, 1), the leaf adjacency nodes will be deleted and will go to the next step. Otherwise, it means V_A and V_B cannot be matched, and the subgraph searching process will be terminated.

The second step is to construct a bipartite graph $G\prime = ((V_A', V_B'), E\prime)$, where V_A' is the adjacency nodes set of v_a, V_B' is adjacency nodes set of v_b ($V_A' \cap V_B' = \varnothing$), and E' is edges set between V_A' and V_B'.

As shown in Figure 4, v_{ai} is the ith adjacency node of v_a, v_{bj} is the jth adjacency node of v_b, and $\exists e_{ij} \in E'$ represents a possible connection between v_{ai} and v_{bj}. The weight of e_{ij} is $w(e_{ij})$. Equation (3) represents the total weight of a matching W_T.

$$W_T = \sum_{e_{ij} \in E'} w\left(e_{ij}\right). \tag{3}$$

Thus, if there are more than one matche, the max W_T ($W_{\max}$) can be treated as the optimal matching, as shown in the following:

$$W_{\max} = \max \sum_{e_{ij} \in E'} w\left(e_{ij}\right). \tag{4}$$

To get $W_{\max}$, the third step is to assign a weight $w(e_{ij})$ to the edges of E' as shown in Table 1.

Table 1 shows the edge-weighted matrix of E', where m is the row number and n is the column number. If $(m! = n)$, our method will always make $(m < n)$ and add $(m - n)$ virtual nodes whose value of $w(e_{ij})$ is 0. The edge-weighted matrix generation process can be described in Algorithm 1.

According to the edge-weighted matrix, the maximum weight $W_{\max}$ of G' can be calculated based on the improved KM algorithm. The KM algorithm assigns each node a label value and transforms the maximum weight finding process into the complete match searching. Suppose that the label value of v_{ai} is $l(v_{ai})$ and the label value of v_{bj} is $l(v_{bj})$, the condition $l(v_{ai}) + l(v_{bj}) \geq w(e_{ij})$ should be met during the whole execution of the KM algorithm.

Although the KM algorithm can find $W_{\max}$ successfully, the time complexity of KM is (O^4). It can be found that the efficiency of the algorithm is obviously reduced when the number of nodes exceeds 1000 as shown in Figure 5. To solve this problem, the KM algorithm is improved as follows:

(1) Calculate the label value of nodes based on edge's weight. Let $w(e_{ij}) = l(v_{ai}) + l(v_{bj})$. In this step, let $l(v_{ai}) = \max(w(e_{ij}))$ and $l(v_{bj}) = 0$.

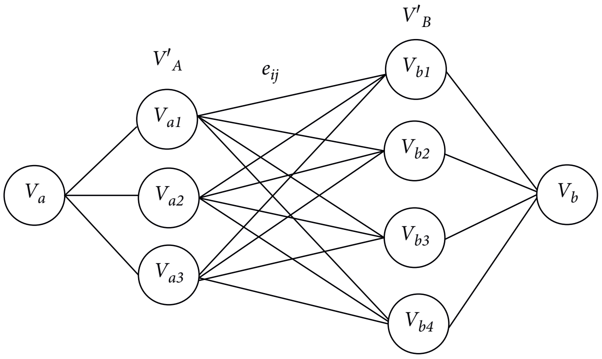


Figure 4: The bipartite graph of adjacency nodes.

Table 1: Edge-weighted matrix of E'.

	v_{b1}	v_{b2}	$\cdots$	v_{bn}
v_{a1}	$w(e_{ij})$	$\cdots$	$\cdots$	$w(e_{ij})$
v_{a2}	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
v_{am}	$w(e_{ij})$	$\cdots$	$\cdots$	$w(e_{ij})$

```
 **Input:** Bipartite graph of adjacency nodes $G' = ((V'_A, V'_B), E')$ and the weight of all adjacency nodes.
**Repeat:**
    Choose $\exists v_{ai} \in V'_A$ and $\exists v_{bj} \in V'_B$,
    **if** ($v_{ai}$ or $v_{bj}$ is not a virtual node)
        **if** $(w(v_{ai}) - w(v_{bj}) \neq 0)$
        Calculate the weight of edge:
        $w(e_{ij}) = (w(v_{ai}) + w(v_{bj}))/|w(v_{ai}) - w(v_{bj})|$
        Insert $w(e_{ij})$ into the edge-weighted matrix
    **else**
        $w(e_{ij}) = 0$
        Insert $w(e_{ij})$ into the edge-weighted matrix
**Until:** Every edge is assigned a weight.
**Output:** The edge-weighted matrix.
```

Algorithm 1: Edge-weighted matrix generation.

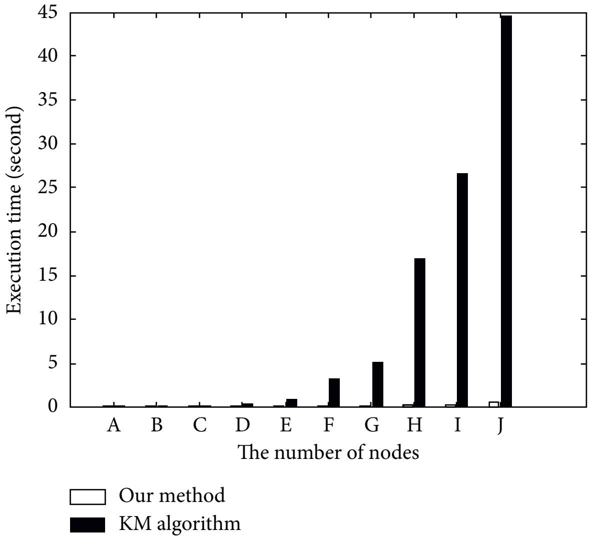


Figure 5: The execution time of our method on different sizes of applications.

(2) Prepare for augmenting path searching of equal subgraphs. Suppose that M represents a perfect match of the bipartite graph, the augmenting paths are searched to find the optimal matching which makes the sum of edges' weights of M maximum. However, the lack of required edges usually leads to the inefficiency of augmenting path search. To solve this problem, several improvements are taken as follows:

Firstly, adjust the label values of nodes. Let

$$\begin{aligned} & l(v_{ai}) - \eta, \\ & l(v_{bj}) + \eta, \\ & \eta = \min\left(l'(v_{ai}) + kl'(v_{aj}) - w(e_{ij})\right), \end{aligned} \tag{5}$$

where $v_{ai} \in V_A'$, $v_{bj} \in V_B'$, and $v_{aj} \in V_B'$. $l'(v_{ai})$ is the label value of v_{ai} where v_{ai} belongs to the searched augmenting path. $l'(v_{aj})$ is the label value of v_{aj} where v_{aj} has not been in the searched augmenting path yet. This change can make more nodes and edges meet the condition $l(v_{ai}) + l(v_{bj}) = w(e_{ij})$. Secondly, define an array named "*slack*"; let $slack[i] = \infty$ before the augmenting path searching each time. When $w(e_{ij})$ is searched, let $slack[i] = \min(l'(v_{ai}) + l'(v_{bj}) - w(e_{ij}))$, where v_{ai} belongs to the searched augmenting path. v_{bj} has not been in the searched augmenting path yet. This change makes the algorithm never search the edge repetitively until an augmenting path is successfully found.

(3) Search the augmenting path by the Hungarian algorithm to find M.

(4) If M is not found, change the label values of accessed nodes as follows:

$$\begin{aligned} & l(v_{ai}) - \eta', \\ & l(v_{bj}) + \eta', \\ & \eta' = \min(slack[j]), \\ & slack[j] - = \eta', \end{aligned} \tag{6}$$

where $j \in V_B'$ and j has not been in the searched augmenting path yet.

(5) Execute steps (3)-(4) repetitively until the perfect matching of the equal subgraph is found.

According to the above steps, on the basis of the paths that have been searched, only the newly added edges are searched to ensure that each edge is searched once.

4.5. Malware Detection. When both sensitive permission-based subgraphs and common subgraph of each family are collected, our malicious feature set can be generated. The detection result of a test application can be obtained by matching its FCG with the feature set. The matching steps are as follows. Firstly, find the similar nodes of structural features' initial nodes in test application's FCG. Secondly, continue to match the initial nodes' adjacent nodes according to the nodes similarity calculation process which is introduced in Subsections 4.3 and 4.4. The FCG can be judged to contain a malicious structure if all nodes of a malicious structure feature are successfully matched. At the same time, the malware family to which the application belongs can be detected. Otherwise, the application is judged to be safe.

In order to promote execution speed, the family feature straining process is taken off-line.

5. Experiments and Evaluation

5.1. Dataset and Evaluation Metrics. To test the effectiveness of our method more comprehensively, three malware datasets and one benign dataset are collected. The first dataset (dataset 1) is Genome Project which was collected by Jiang and Zhou [33] in 2012. It contains 1247 malware samples of 49 malware families. This dataset involves many kinds of attack techniques such as repackaging, remote control, personal information stealing, and update attack techniques. The authors' experimental results showed that four famous commercial malware detection tools can only achieve unsatisfied detection rate on this dataset (20–79%).

The second dataset (dataset 2) is Drebin which contains 5560 malware samples of 179 families. Its malware samples were collected from mobile application markets in Russia, China, and so on [34].

These two datasets are chosen because of their high frequency of use in many former works [11, 21, 24, 35–37].

Malware dataset 3 contains malicious apps collected from the virus share [29] and Android malware dataset (AndMal) [38] which contains 713 malware of 42 families.

The benign apps are collected from Google Play Store, 360 application market [1, 33, 38], and so on. All of them are checked by frequently used antivirus softwares McAfee and Kaspersky to ensure their safety, as shown in Table 2.

Based on these datasets, the performance of detection methods is evaluated by True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN), accuracy, FPR, TPR, recall, and precision. They can be defined as follows.

The row of Table 3 means the actual type of applications. The column means the prediction type of the detection method.

$$\begin{aligned} \text{Accuracy} &= \frac{TP + TN}{TP + FN + TN + FP}, \\ \text{Precision} &= \frac{TP}{TP + FP}, \\ \text{Recall} &= \frac{TP}{TP + FN}, \\ FPR &= \frac{FP}{TN + FP}, \\ TPR &= \frac{TP}{TP + FN}. \end{aligned} \tag{7}$$

The following experiment is organized into four parts: firstly, the malware family identification ability of our

Table 2: Information of dataset.

Name	Sample number	Family/category number	Alias in this paper
Genome project	1247	49	Dataset 1
Drebin	5560	179	Dataset 2
AndMal	713	42	Dataset 3
Benign	4000	20	Benign dataset

Table 3: Matrix of metrics.

	Malware	Benign
Malware	TP	FN
Benign	FP	TN

method is compared with several former works in Subsection 5.2; secondly, the FPR of our method in malware family detection is evaluated in Subsection 5.3; thirdly, a 3-fold detection process is implemented to evaluate the detection ability of our method on unknown malware in Subsection 5.4; and finally, the runtime performance of our method is discussed in Subsection 5.5.

5.2. Compare with Graph-Based Detection Methods. As mentioned in Section 2, many research studies have applied their malware detection experiments on Genome dataset. Thus, the detection ability of our method is compared with three efficient former graph-based detection models and Androguard tool in this subsection. The first method is Dendroid [18]. Dendroid put a detailed analysis on CFGs of Android applications and designed a string feature extraction strategy. It successfully transformed the subgraph comparison into text mining. By comparing the similarity of string features, Dendroid can classify similar applications into the same malware family. The second one is "MIGDroid" [19], which is also a subgraph analysis-based detection method. Differently from Dendroid, MIGDroid calculated subgraphs' threat scores according to the sensitive features belonging to them. By evaluating the threat scores, MIGDroid can identify malware of each family. The third method is [23] (it is named "CFG-based method" in this experiment). This method improved the text mining model of Dendroid by adding machine learning classification into it.

Table 4 shows the detection rates among these former works and ours. The experimental results indicate that Androguard gets the worst detection rate because it is over-reliant on signatures in malware detection. Without relative signatures, it cannot identify malware even if the variants are very similar to the existing malware. MIGDroid is more easily influenced by the APIs. For example, the variants of "zHash" family are added a lot of garbage code. These garbage codes have close connections with the rest normal codes. Therefore, identifying which part of the garbage codes is benign or malicious is difficult for MIGDroid. Dendroid classifies malware into families based on CCs. However, the CCs' distribution is irregular. It means some families may have a large number of CCs, but some families may have few. At the same time, when families are similar to each other, the classification performance of Dendroid may decline, such as DroidKungFuX, BaseBridge, and AnserverBot. The CFG-based method has good performance on the detection of DroidKungFu family, but the average detection rate is lower than Dendroid. Our method has the highest average detection rate, which means it can locate the malicious structural feature with high efficiency. However, the accuracy of our method is still reduced when it is used to detect high similar families, such as DroidKungFu families.

5.3. The TPR and FPR Evaluation of Malware Family Classification. In addition to the detection rate, TPR and FPR are also important metrics. The TPR and FPR of our method among malware families are evaluated in two parts. Firstly, the identification result is evaluated by the 8 most closely related malware families. They are BaseBridge (BB), AnserverBot (AB), DroidKungFu (DK), GoldDream (GD), DroidDreamLight (DDL), Pjapps (PJ), DroidDream (DD), Plankton (PK), and Zsone (ZS). These families are chosen because they may cause higher FPR than other unrelated families [25].

Table 5 shows the confusion matrix of the family classification. The column represents actual families, and the row represents the identification results. Experimental result shows that the highest FPR is 3.2%. It only appears between BaseBridge and AnserverBot since they contain the virus variants of the same ancestor.

Secondly, 30 representative families of dataset 1 and dataset 3 are selected to evaluate their FPR and TPR in malware detection. Table 6 shows the FPR of our method on each family. The experimental result shows that the highest FPR is 3.8%, and the average FPR of these malware families is 1.1%.

5.4. Evaluation with Unknown Applications. In this subsection, the detection ability of our method on the unknown malware is evaluated. Thus, the experimental dataset is divided into two different parts: 80 percent random malware of each family (6016 samples with 221 families of dataset 1–dataset 3) are chosen as the training set. The rest applications (1504 malware and 4000 benign samples) are chosen as the unknown test set. In order to evaluate the stability of our detection method, this dataset division process is executed three times to get three different training sets and unknown test sets. The detection result of each time (named result 1, result 2, and result 3) is shown in Tables 7–9.

Tables 7–9 are the confusion matrixes of each detection result. The row is the actual type of applications, and the column is the prediction type of the detection method. Table 10 provides the metrics comparison among result 1 to result 3. It shows that the maximum difference of accuracy, recall, and precision is 1%, 0.1%, and 2.1%, respectively. This result indicates that the detection rate is stable, but it is still affected by different training samples.

5.5. Runtime Performance. In this subsection, the runtime performance of our method is evaluated. The execution efficiency of our method is mainly affected by the size of the

Table 4: Comparison of detection rates among malware families.

Family	Dendroid (%)	MIGDroid (%)	Androguard (%)	CFG-based method (%)	Ours (%)
ADRD	100	100	59.09	95.50	100
AnserverBot	96.70	100	0	98.93	96.80
BeanBot	100	100	0	62.5	100
Bgserv	100	100	0	88.9	100
DroidDream	100	100	93.75	93.75	100
DroidDreamLight	100	100	28.26	100	100
DroidKungFu1	88.24	97.06	0	97.05	100
DroidKungFu2	80	60	0	96.66	96.66
DroidKungFu3	92.56	99.35	0	100	100
DroidKungFu4	83.33	72.92	0	100	100
Geinimi	100	100	97.10	100	100
GoldDream	100	100	0	93.75	100
jSMSHider	100	100	0	100	100
Pjapps	100	72.40	41.38	97.78	98.27
Plankton	100	90	18.18	90.91	100
YZHC	100	100	95.45	95.45	100
Zsone	100	100	100	100	100
Asroot	100	50	0	37.50	100
BaseBridge	92.80	44.26	0	99.18	98.36
KMin	100	21.15	78.86	100	100
RogueSPPush	100	55.56	100	98.88	100
SndApps	100	0	100	100	100
zHash	100	0	0	90.91	100
Average	97.11	76.63	35.30	92.94	99.56

Table 5: The confusion matrix for classification of 8 families (%).

	BB	AB	DK	GD	DDL	PJ	DD	PK	ZS
BB	98.4	1.6	0	0	0	0	0	0	0
AB	3.2	96.8	0	0	0	0	0	0	0
DK	0	0	100	0	0	0	0	0	0
GD	0	0	0	100	0	0	0	0	0
DDL	0	0	0	0	100	0	0	0	0
PJ	0	0	0	0	0	100	0	0	0
DD	0	0	0	0	0	0	100	0	0
PK	0	0	0	0	0	0	0	100	0
ZS	0	0	0	0	0	0	0	0	100

Table 6: The FPR of 30 typical malware families of the dataset.

Family name	TPR/FPR (%)	Family name	TPR/FPR (%)	Family name	TPR/FPR (%)
DroidDream	100/1.5	RogueSPPush	100/2	Koodous	100/0
DroidDreamLight	100/1	SndApps	100/2	Mobidash	100/2.2
Geinimi	100/1.4	zHash	100/0	RansomBO	100/0
jSMSHider	100/0	Dowgin	100/3.8	Svpeng	100/0
Plankton	100/1.3	Edwin	100/0	Koler	100/2.4
YZHC	100/0	Feiwo	100/0	Pletor	100/0
Zsone	100/0	Gooligan	100/2.5	PornDroid	100/2.3
Asroot	100/0	Kemoge	100/0	Charger	100/3.7
Bgserv	100/0	Fakemart	100/3	Jisut	100/0
KMin	100/0	Jifake	100/3.1	MazarBot	100/0
Average	1.1				

Table 7: The confusion matrix of result 1.

	Malware	Benign
Malware	1475	29
Benign	40	3960

Table 8: The confusion matrix of result 2.

	Malware	Benign
Malware	1455	49
Benign	73	3927

Table 9: The confusion matrix of result 3.

	Malware	Benign
Malware	1459	45
Benign	68	3932

Table 10: Comparison of result 1–result 3.

Detection result	Accuracy	Recall	Precision
Result 1	0.987	0.968	0.973
Result 2	0.977	0.967	0.952
Result 3	0.979	0.97	0.955

Table 11: Information of the different sizes of applications.

Label	Average number of nodes	Average number of edges
A	180	340
B	550	1020
C	690	1800
D	1110	2800
E	1600	3730
F	2500	6800
G	2900	7000
H	4400	10300
I	5620	16030
J	8680	20700

FCGs. Thus, the test dataset of this experiment is established based on different sizes of applications' FCGs. Each application's FCG size is represented by the number of nodes and edges, which are divided into 10 levels, as shown in Table 11.

Figure 5 shows the execution time of our method on these applications. Experimental result shows that the execution time of our method rises from 0.003 seconds to 0.524 seconds, whereas the execution time of the KM algorithm rises from 0.006 seconds to 44.5 seconds. It indicates that the efficiency of our method has been improved obviously.

6. Conclusion

In this paper, a new structural feature-based Android malware detection method is introduced. The method can automatically extract static features from applications and generate FCGs and sensitive permissions. Then, sensitive permission-based subgraphs and the common subgraph of each malware family are constructed as malicious features. At last, unknown applications' safety can be identified by these malicious features. This method is proved useful from the following three aspects. First, the family detection rate of our method is evaluated by comparing with three former works and Androguard. Experimental results indicate that our method can get higher detection accuracy among many representative malware families of our dataset. Second, the detection ability of unknown malware is evaluated. Third, the runtime performance of our method is evaluated by different sizes of applications. All evaluation results indicate that our method can achieve good performance on different kinds of malware with various attack technologies.

Although our method is efficient, the experimental results also indicate that our method can be improved in many ways. Specifically, the next work can be taken in the following two directions: (1) research robust defense methods for malicious obfuscation technologies which can modify the code structures of malicious subgraphs and (2) design more efficient models to store the expanding structural features. Thus, more efficient graph analysis models or algorithms should be designed in future works. Moreover, more heuristic features are needed to cope with the rapid improvement of malicious attack technologies.

Data Availability

The data used to support this study are from previously reported studies [33], [34], and [38].

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Sichuan Science and Technology Program (grant no. 2021JDRC0075), Fundamental Research Funds for the Central Universities, Southwest Minzu University (grant no. 2020NZD02), and Chengdu Science and Technology Program (grant no. 2021-GH03-00001-HZ).

References

[1] "Google play store," 2019, https://play.google.com/store/.

[2] G. Kelly, "Report: 97% of mobile malware is on android. this is the easy way you stay safe," 2014, http://www.forbes.com/sites/gordonkelly/2014/03/24/report-97-of-mobile-malware-is-on-android-this-is-the-easy-way-you-stay-safe/.

[3] "Kaspersky report," 2015, https://www.kaspersky.com/about/press-releases?rel=1&sel=date.

[4] "Nokia threat intelligence report," 2019, https://pages.nokia.com/T003B6-Threat-Intelligence-Report-2019.html.

[5] V. Deepak and H. Guoning, "Efficient signature based malware detection on mobile devices," *Mobile Information Systems*, vol. 4, no. 1, pp. 33–49, 2014.

[6] R. H. Niazi, J. A. Shamsi, T. Waseem et al., "Signature-based detection of privilege-escalation attacks on Android," in *Proceedings of the Conference on Information Assurance and Cyber Security*, pp. 44–49, Rawalpindi, Pakistan, February 2016.

[7] İ. A. Doğru and M. önder, "AppPerm analyzer: malware detection system based on android permissions and permission groups," *International Journal of Software Engineering and Knowledge Engineering*, vol. 30, no. 4, pp. 427–450, 2020.

[8] M. Scalas, D. Maiorca, F. Mercaldo, C. A. Visaggio, F. Martinelli, and G. Giacinto, "On the effectiveness of system

API-related information for Android ransomware detection," *Computers & Security*, vol. 86, pp. 168–182, 2019.

[9] O. Yildiz and I. A. Doğru, "Permission-based android malware detection system using feature selection with genetic algorithm," *International Journal of Software Engineering and Knowledge Engineering*, vol. 29, no. 2, pp. 245–262, 2019.

[10] A. Bhattacharya, R. T. Goswami, and K. Mukherjee, "A feature selection technique based on rough set and improvised PSO algorithm (PSORS-FS) for permission based detection of Android malwares," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 7, pp. 1893–1907, 2019.

[11] M. Zhang, Y. Duan, H. Yin et al., "Semantics-aware Android malware classification using weighted contextual API dependency graphs," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1105–1116, Scottsdale, AZ, USA, November 2014.

[12] L. Taheri, A. F. A. Kadir, and A. H. Lashkari, "Extensible android malware detection and family classification using network-flows and API-calls," in *Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST)*, Chennai, India, October 2019.

[13] W. Huanran, H. Hui, and Z. Weizhe, "Demadroid: object reference graph-based malware detection in android," *Security and Communication Networks*, vol. 2018, pp. 1–16, Article ID 7064131, 2018.

[14] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen, and H. Bunke, "Approximation of graph edit distance based on Hausdorff matching," *Pattern Recognition*, vol. 48, no. 2, pp. 331–343, 2015.

[15] J. Crussell, C. Gibler, and H. Chen, "Attack of the clones: detecting cloned applications on android markets," *European Symposium on Research in Computer Security*, vol. 81, no. 13, pp. 2454–2456, 2012.

[16] M. Xu, L. Wu, S. Qi et al., "A similarity metric method of obfuscated malware using function-call graph," *Journal of Computer Virology and Hacking Techniques*, vol. 9, no. 1, pp. 35–47, 2013.

[17] "Soot: a java optimization framework," 2019, https://gitee.com/alexbill/soot.

[18] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and J. Blasco, "Dendroid: a text mining approach to analyzing and classifying code structures in Android malware families," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1104–1117, 2014.

[19] W. Hu, J. Tao, X. Ma et al., "MIGDroid: detecting APP-Repackaging Android malware via method invocation graph," in *Proceedings of the Conference on Computer Communication and Networks*, pp. 1–7, IEEE, Washington, DC, USA, August 2014.

[20] W. Niu, R. Cao, X. Zhang, K. Ding, K. Zhang, and T. Li, "OpCode-level function call graph based android malware classification using deep learning," *Sensors*, vol. 20, no. 13, p. 3645, 2020.

[21] T. Gao, W. Peng, D. Sisodia, T. K. Saha, F. Li, and M. Al Hasan, "Android malware detection via graphlet sampling," *IEEE Transactions on Mobile Computing*, vol. 18, no. 12, pp. 2754–2767, 2019.

[22] X. Sun, Y. Zhongyang, Z. Xin et al., "Detecting code reuse in android applications using component-based control flow graph," *IFIP Advances in Information & Communication Technology*, vol. 428, pp. 142–155, 2016.

[23] M. A. Atici, S. Sagiroglu, and I. A. Dogru, "Android malware analysis approach based on control flow graphs and machine learning algorithms," in *Proceedings of the 2016 4th International Symposium on Digital Forensic and Security (ISDFS)*, IEEE, Little Rock, AR, USA, May 2016.

[24] T. S. John, T. Thomas, and S. Emmanuel, "Graph convolutional networks for android malware detection with system call graphs," in *Proceedings of the 2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP)*, IEEE, Guwahati, India, April 2020.

[25] W. Zhang, H. Wang, H. He, and P. Liu, "DAMBA: detecting android malware by ORGB analysis," *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 55–69, 2020.

[26] P. Abdurrahman and T. Acarman, "Deep learning for effective Android malware detection using API call graph embeddings," *Soft Computing*, vol. 24, no. 2, pp. 1027–1043, 2020.

[27] S. Y. Yerima, M. K. Alzaylaee, and S. Sezer, "Machine learning-based dynamic analysis of Android apps with improved code coverage," *EURASIP Journal on Information Security*, vol. 4, 2019.

[28] Z. Lin, R. Wang, X. Jia et al., "Classifying android malware with dynamic behavior dependency graphs," in *Proceedings of the 2016 IEEE Trustcom/BigDataSE/I SPA*, IEEE, Tianjin, China, August 2016.

[29] L. Xu, D. Zhang, M. A. Alvarez et al., "Dynamic android malware classification using graph-based representations," in *Proceedings of the 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*, IEEE, Tianjin, China, August 2016.

[30] S. Hou, A. Saas, L. Chen et al., "Deep4MalDroid: a deep learning framework for android malware detection based on linux kernel system call graphs," in *Proceedings of the 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW)*, ACM, Omaha, NE, USA, October 2016.

[31] "Androguard project," 2016, http://code.google.com/p/androguard/.

[32] Y. Zhou and J. Kuang, "A sort method to enhance significant spectral components of test set," in *Proceedings of the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 2147–2151, Changsha, China, August 2016.

[33] X. Jiang and Y. Zhou, "Dissecting android malware: characterization and evolution," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 95–109, San Francisco, CA, USA, April 2012.

[34] D. Arp, M. Spreitzenbarth, M. Hübner et al., "DREBIN: effective and explainable detection of android malware in your pocket," in *Proceedings of the Network & Distributed System Security Symposium*, San Diego, CA, USA, February 2014.

[35] S. Arzt, S. Rasthofer, C. Fritz et al., "FlowDroid," *ACM SIGPLAN Notices*, vol. 49, no. 6, pp. 259–269, 2014.

[36] M. Grace, Y. Zhou, Z. Qiang et al., "RiskRanker: scalable and accurate zero-day android malware detection," in *Proceedings of the 10th International Conference on Mobile Systems, Applications and Services*, pp. 281–294, Istanbul, Turkey, June 2012.

[37] V. Rastogi, Y. Chen, and X. Jiang, "Catch me if you can: evaluating android anti-malware against transformation attacks," *IEEE Transactions on Information Forensics & Security*, vol. 9, no. 1, pp. 99–108, 2013.

[38] A. H. Lashkari, A. F. A. Kadir, L. Taheri et al., "Toward developing a systematic approach to generate Benchmark android malware datasets and classification," in *Proceedings of the 2018 International Carnahan Conference on Security Technology (ICCST)*, Montreal, Canada, October 2018.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5567991, 11 pages
https://doi.org/10.1155/2021/5567991

Research Article

Social Network Spam Detection Based on ALBERT and Combination of Bi-LSTM with Self-Attention

Guangxia Xu, Daiqi Zhou, and Jun Liu

School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

Correspondence should be addressed to Guangxia Xu; xugx@cqupt.edu.cn

Received 4 February 2021; Revised 15 March 2021; Accepted 24 March 2021; Published 8 April 2021

Academic Editor: Hao Peng

Social networks are full of spams and spammers. Although social network platforms have established a variety of strategies to prevent the spread of spam, strict information review mechanism has given birth to smarter spammers who disguise spam as text sent by ordinary users. In response to this, this paper proposes a spam detection method powered by the self-attention Bi-LSTM neural network model combined with ALBERT, a lightweight word vector model of BERT. We take advantage of ALBERT to transform social network text into word vectors and then input them to the Bi-LSTM layer. After feature extraction and combined with the information focus of the self-attention layer, the final feature vector is obtained. Finally, SoftMax classifier performs classification to obtain the result. We verify the excellence of the model with accuracy, precision, F_1-score, etc. The results show that the model has better performance than others.

1. Introduction

Online social network platforms (OSNs) provide users with convenient communication and interactive tools, which can instantly share various content related to life and work, including text, pictures, and videos. Because of the support of wireless communication and computer, OSNs have become very popular than before [1]. However, a great quantity active user and the convenient conditions for publishing content have attracted a lot of spammers on OSNs. The release of spam by spammers has caused great troubles to normal users and platforms. According to reports, spam that had little or limited impact in the past can now take advantage of OSNs to cause a huge distributed impact [2]. OSNs disclose all basic user information and provide follow-up functions, which enables spammers to easily and accurately send spam to potential target users and promote dissemination [3]. For example, a social platform message embedded with a URL may spread to thousands of users within a few minutes.

Spam is a kind of information actively sent by spammers, and its purpose is to deceive, spread lies, and advertise for profits [4]. Spam will cause problems such as resource occupation, extended communication time, and bandwidth waste [5]. A report showed that on most OSNs, the growth rate of spam exceeded the rate of ordinary reviews [6]. It also showed that 15% of spams contain links to malicious content, pornography, or malware. Although a lot of related research has been done, there are still a large number of spams on social networks. And the spammer that manufactures and sends spam will disguise spam by observing the platform filtering strategies. It shows that there are still deficiencies in these methods.

In this paper, we mainly study a spam detection model in the Sina Weibo social network. This model combines ALBERT and Bi-LSTM network based on self-attention mechanism. The contribution of this work can be summarized as follows:

(i) In view of the huge amount of social network tweet data, introducing the ALBERT model to embedding the text to improve the efficiency of model classification while ensuring the accuracy of the model.

(ii) Aiming at the situation where spammer hides the spam camouflage in the normal text, introducing the Bi-LSTM to the spam tweet recognition method, which can fully consider the context and semantics to capture the core of tweets text.

(iii) Because of the limited content of a single tweet (mostly less than or equal to 140 characters), which cannot provide a large amount of effective information, introducing the self-attention mechanism to the Bi-LSTM model to further obtain key words that affect the classification results of tweet characteristics.

The rest of this paper is organized as follows. We present background and related work in Sections 2 and 3, respectively. Section 4 introduces the architecture and details of the model. The implementation of the experiment is described in Section 5, and the results and analysis is discussed in Section 6. Section 7 concludes this paper and makes an outlook for future work.

2. Background

At present, there have been many related studies on the spam detection in social networks. Among them, part of the research has laid the foundation. Gupta et al. [7] conducted a security survey on pervasive online social networks, mainly exploring the trust management mechanism and anomaly detection mechanism of social network platforms, and raised current problems in anomaly detection, among which spam belongs to one. Branitskiy et al. [8] used machine learning methods to process social network data in order to explore the sensitivity of young generation to stimulating information on social networks. This research will help us distinguish between ordinary users and malicious users. Social network-based relationship graph methods are usually proposed when studying social networks, but the data are huge and not easy to calculate. Kolomeets et al. [9] proposed a reference architecture for storing and analysing graph data and provided visualization. Xu et al. [10] proposed a weighted algorithm for social network topology graphs, which can help researchers find smaller communities. In addition to the research on network topology, there is also a certain research foundation on the content of social network platforms. Jeong et al. [11] proposed a method to calculate text similarity-based edge weights. Experiments show that it can effectively find similar text data. Xu et al. [12] extracted audio and text from videos on social networks and used 3DCLS (3D Convolutional-Long Short-Term Memory) hybrid framework to analyse user emotions. In addition, there have been many studies for spam detections. After reviewing the relevant literature, it is found that the research can be classified to 2 types: traditional machine learning method and complex neural network method.

2.1. Traditional Machine Learning Method. Traditional machine learning method has been widely used in spam detection. These approaches build a classification model based on the characteristic attributes of the spam text. Most of these attributes include the number of URLs, key words, etc. [13]. Amayri and Bouguila [14] applied SVM to spam detection. They focused on the impact of SVM kernels. The results showed that string kernels are better than distance-based kernels. Vangelis et al. [15] made many different improvements to naive Bayes. They proposed five improved models based on Naive Bayes (NB), including multivariate Bernoulli NB, TF multinomial NB, boolean multinomial NB, flexible Bayes, and multivariate Gauss NB. The experiments show that flexible Bayesian and multinomial NB with Boolean attributes have a leading role. Soiraya et al. [16] took advantage of the J48 decision tree to analyse the Facebook message text. They detected the keywords, the average number of words, the length of the text, and the number of links contained in the information. The accuracy and recall rates of the research results were 61% and 63%, respectively. Johnson et al. [17] compared traditional machine learning algorithms with deep learning framework algorithms and found that random forests has better performance in text URL detection.

In addition to the classification model based on the single method, researchers also try to combine multiple methods to achieve better classification effect. The authors in [18] applied the Markov clustering (MCL) algorithm and allocated the probability of each node in the network by using the weighted graph as the input of the algorithm. In [19], a hybrid machine learning spam detection model is proposed based on support vector machine, and the experiments were carried out on four language datasets (Arabic, English, Spanish, and Korean). It showed that the accuracy of the model was better than other algorithms.

2.2. Complex Neural Network Method. In recent years, the model of deep neural network has shown strong ability in the field of natural language processing. It includes word representation learning, sentence and document representation, grammar analysis, machine translation, and sentiment classification. In the field of spam detection, many methods have been proposed. Tien and Nur [20] explored an artificial neural network language model to distinguish different users' short text writing styles, so as to distinguish and identify users, and then detect spam users. Ma et al. [21] established a microblog rumor detection system using LSTM and GRU. Ruan and Tan [22] introduced a technique based on three-layer back propagation neural network and feature construction based on concentration. In [23], a text classification model based on word2vec is constructed to solve the high-dimensional problem of traditional methods. In addition, LSTM is added to extract the key information of the text. Finally, this method is applied to the classification of patent texts. Experiments show that the accuracy of classification is 93.48%. Luan and Lin [24] proposed a text classification model called CNN-COIF-LSTM. The experimental results show that the combination of CNN and LSTM has higher accuracy without activation function or its variants. Recently, self-attention mechanism has attracted people's attention and achieved state-of-the-art results. Dong et al. [25] combined a self-interaction attention mechanism with label representation and used the Bert model to solve the problem of text feature extraction. In this method, joint word representation and label representation are proposed to improve the efficiency of the model. Experiments also prove the correctness of the method.

3. Related Work

3.1. BERT and ALBERT. BERT (bidirectional encoder representations from transformers) was proposed by Devlin et al. [26] of Google in 2018, and immediately it showed a strong ability in the NLP field. The structure of Bert is shown in Figure 1.

BERT uses transformer [27] structure as the main framework. It converts the distance of two words at any position into one, which effectively solves the long-term dependency problem in NLP. The training process of BERT includes two parts: MLM (masked language model) and NSP (next sentence prediction). In the first part of experiment, 15% of the words were randomly masked, 80% of the words were replaced by (mask), 10% were replaced by any other words, and 10% were kept in the original state. In the second part, the model randomly extracts two consecutive sentences, 50% of which retain the extracted two sentences, their relationship is labeled IsNext, the other 50% of the second sentence is randomly extracted from the corpus, and their relationship is labeled NotNext.

The structure of ALBERT refers to BERT, and it still uses transformer and GELU activation function. However, at the same time, there are some innovations compared with BERT, including the following three points:

(1) Factored embedding parameterization: it reduces the word embedding dimension of embedding layer and adds a project layer between word embedding and hiding layer. Suppose the size of thesaurus is L, H represents the dimension of hidden layer, and the dimension of word embedding is V.

The calculation formula of the parameters of the BERT model is

$$P_{\text{bert}} = L \times H. \tag{1}$$

The calculation formula of the parameters of the ALBERT model is

$$P_{\text{albert}} = L \times V + V \times H. \tag{2}$$

In the ALBERT model, the dimension of word embedding is the same as the hidden layer. When V is large and V is far less than H, the number of parameters will be reduced after factorization of word embedding.

(2) Cross-layer parameter sharing: the parameters in each layer of transformer are relatively independent, including self-attention and full connection, which will lead to a significant increase of parameters when the layers increase. In order to decrease the number of parameters and promote the stability of the model, ALBERT tries to share all the parameters.

(3) Inter-sentence coherence loss: ALBERT has changed the NSP of BERT into a sentence order prediction (SOP). In SOP, the construction of positive example is consistent with NSP, but the negative example is to reverse the two sentences.

(4) Dropout was canceled.

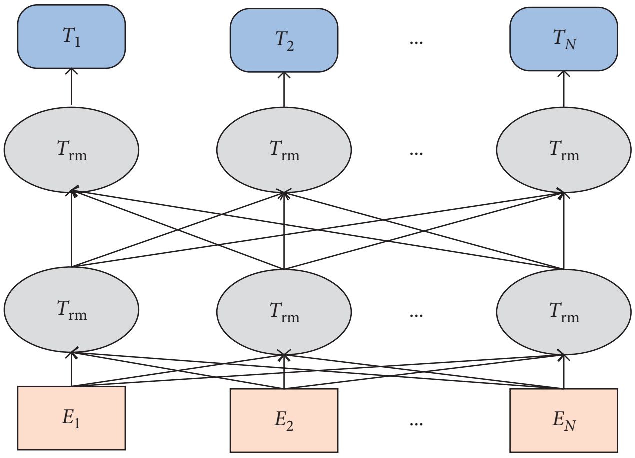


Figure 1: The structure of BERT.

3.2. LSTM and Bi-LSTM. Although RNN can support information persistence, it cannot achieve significant effect for some complex scenarios. For example, we try to predict the following: "I grew up in Sichuan I say "xxx" fluently". The RNN will give a name of language for field in, but the answer is wrong most of the time because the word "Sichuan" that is helpful for prediction is too far away. However, in the case of increasing the interval, we will not be able to learn the information between them. LSTM, a special RNN, was proposed by Hochreiter et al. [28] in 1996, and it can stably learn long-term dependent information. Having a chained network module is a feature of all RNNs. In a normal RNN, this repeating module has a very simple structure, such as a tanh layer. There is the same structure in LSTM, but the internal of the structure is different from RNN. The difference is that it has four modules that play different roles.

The structure of LSTM includes input gate i_t, forgetting gate f_t, output gate o_t, and cell state update vector c_t. The structure of LSTM is shown in Figure 2.

The forgetting gate at LSTM selects what information to discard from the cellular state. The formula of this part is as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \tag{3}$$

where W_f and b_f represent the weight matrix and bias matrix of forgetting gate, respectively. σ is the activation function, h_{t-1} represents historical information, and x_t is based on the current input of new information to determine which old information to forget. In this formula, the output h_{t-1} of the previous stage is calculated and combined with x_t. The formula will output a value between 0 and 1. 0 indicates that the old information is completely discarded, and 1 means that the old information is completely retained.

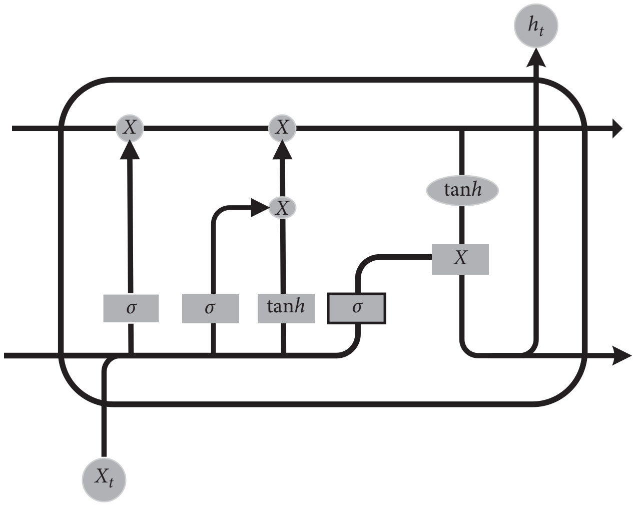


Figure 2: The structure of LSTM.

Choosing what sort of information to store in cell is the main responsibility of input gate. The input gate contains two parts. In the first part, the sigmoid layer (input gate layer) decides what value to update. In the second part, the tanh layer will build a new candidate value vector $\widetilde{C_t}$ and put it in the state. After the coefficients of the input gate and the forget gate are obtained, the current cell state is updated. The formula is as follows:

$$\begin{aligned} i_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \\ \widetilde{C_t} &= \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right), \\ C_t &= f_t * C_{t-1} + i_t * \sim C_t, \end{aligned} \tag{4}$$

where W_i and b_i, respectively, represent the weight of the input gate and tanh is the activation function.

The output gate determines what information will be output at the current stage. In the first step, it calculates a sigmoid layer to decide for which section of the cell state will be output. Then, we will get a value between −1 and 1 through the tanh layer and multiply it with the value of the first step output. Finally, we only output the part that we decided. The formula is as follows:

$$\begin{aligned} O_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right), \\ h_t &= o_t * \tanh\left(C_t\right), \end{aligned} \tag{5}$$

where W_o and b_o represent the weight matrix and the offset matrix of the input gate, respectively.

Bi-LSTM is a combination of a forward LSTM and a backward LSTM. The common forward LSTM has a sequence of information processing. Usually, it only considers the preceding text and ignores the following, so it cannot synthesize the context information to output. The Bi-LSTM structure can obtain enough context information, and both models have a common output layer, as shown in Figure 3.

3.3. Self-Attention Mechanism. In the process of spam detection, we usually faced the situation about the number of texts is limited, especially for the situation that the content information of different users' tweets varies greatly, it is difficult to obtain more effective semantic information. However, through the comparison, it is found that some key words in tweets can help to identify the types of tweets more quickly. For example, words such as "promotion" and "discount" can help to quickly identify the advertising intention of the product tweeted by spammers. At the same time, different words play different roles in classification. Extracting key words helps optimize the feature extraction process. The introduction of attention mechanism can increase the efficiency and improve the classification accuracy. Compared with the attention mechanism, self-attention only computes attention within the sequence, looking for the internal connection of the sequence. The calculation formula is as follows:

$$\text{Attention}\,(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{6}$$

where Q, K, V are three matrices obtained from the same input and different parameters. First, calculate the multiplication of Q, K matrix and divide by $\sqrt{d_k}$. Finally, the softmax operation is used to normalize the result into probability distribution and multiply it by matrix V to get the result. The structure of self-attention is shown in Figure 4.

4. The Proposed Model

In traditional spam detection tasks, it only needs to count the malicious words. However, with the continuous optimization of spam, spammer replaces malicious words with other words, but it can transmit the same information. We need a model which can better understand the sentence and pay attention to the order of words in the sentence, and the information expressed after they are related. ALBERT and Bi-LSTM can better understand the short text information and the association information between words. In the model, we propose a hybrid model with ALBERT, Bi-LSTM, and Self-attention. The model uses ALBERT to extract and

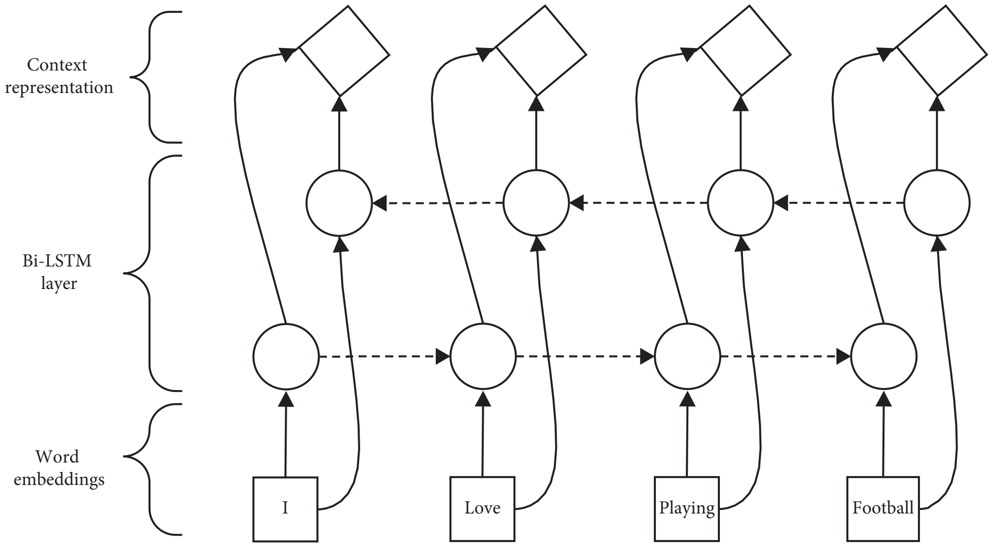


Figure 3: The structure of Bi-LSTM.

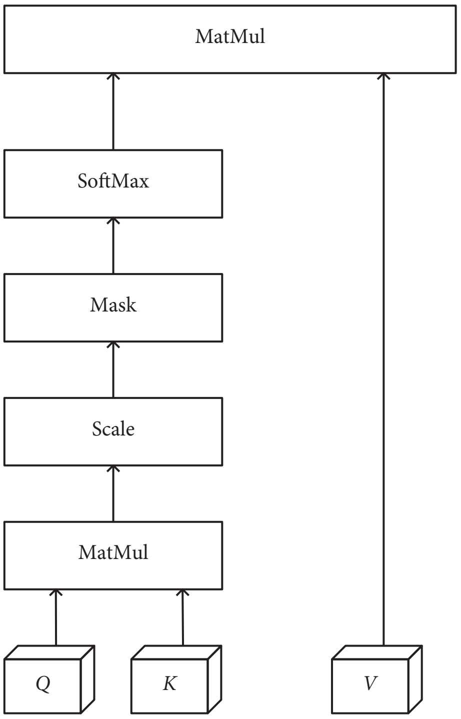


Figure 4: The structure of self-attention.

understand the semantic features of the original text for the first time. Self-attention is combined with Bi-LSTM to detect spam. Figure 5 shows the framework and steps of the model.

4.1. Embedding Layer. Before passing the data into the model, we need to perform some preprocessing operations on the data. For example, remove some stop words and delete emoticons. Then, the embedding layer will serialize the input preprocessed data and convert each word in the text data into a corresponding number in the dictionary. The feature representation will be output through the multilayer bidirectional transformer encoder. The formula is as follows:

$$T = (T_1, T_2, \ldots, T_{N-1}, T_N), \tag{7}$$

where T_i represents the feature vector of the i word in the text.

4.2. Bi-LSTM Self-Attention Layer. The ALBERT model has serialized the data and tried to understand the relationship between words. However, to understand a short text is a tough work for ALBERT. We use a combination of self-attention and Bi-LSTM to perform spam detection. In this layer, the text feature data output by the ALBERT layer will be trained, and the text features will be input into the forward LSTM and the backward LSTM, respectively; then, two text vector representations will be obtained, which will be calculated together to get the final output. Finally, we perform SoftMax normalization on the output result to get result.

5. Experiment

5.1. Dataset. There are two datasets for our experiment. One of the datasets is microblogPCU. It comes from the uci dataset website, which contains the user's basic attribute information (such as gender, number of fans, and number of followers) and the content posted by the user. In this paper, we extracted part of the posted content data. In order to simulate the real social network environment, we set the ratio of the number of spam and nonspam to 2 : 8. There are a total of 2000 data, including 1600 nonspam and 400 spam.

Another dataset is a self-collected Weibo dataset by us. The dataset contains the basic information and tweets of 985 users who have been labeled, of which 403 are marked spammers and 582 are nonspammers. It contains 95,385 Weibo twitters in total. We randomly selected 1000 Weibo

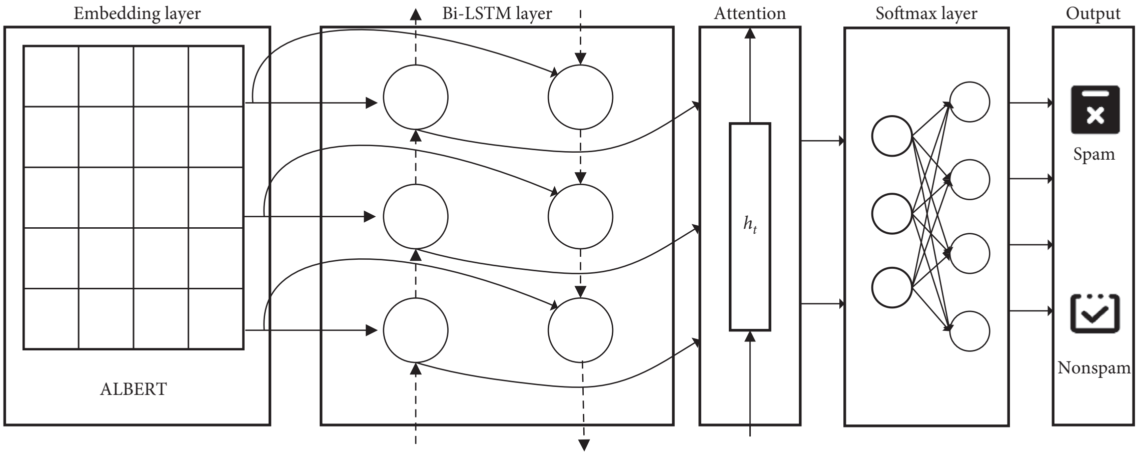


Figure 5: Spam detection model based on ALBERT and self-attention Bi-LSTM.

contents from spammer and nonspammer, respectively, checked, and labeled each one.

5.2. Input Preparation. The input to Bi-LSTM network is the user's Weibo text, but the length of the text is inconsistent. However, the network we built can only accept fixed length text, so we first analysed the length of the input network text, as shown in Table 1. We take 75% of the length quantile 80 as the maximum length value of the input model. For texts with length less than 80, we perform filling operation, and for texts with length greater than 80, we clear the subsequent text. In this way, the user's Weibo content will be retained and will not produce excessively filled meaningless content. And the model will also get enough text to extract semantic information. The processed text is then entered into the model.

5.3. Evaluation Metrics. We calculate precision, recall, F_1-score, and accuracy to judge the performance of the model. Accuracy is the probability of spam in the selected spam text, recall rate is the probability of correctly predicting as spam, F_1-score is the harmonic mean of precision and recall, and accuracy is the rate that the predicted correct text accounts for the total text. The formula is as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$
$$\text{recall} = \text{TPTP} + \text{FN}, \tag{8}$$
$$F_1 - \text{score} = 2 \times \text{precision} \times \text{recall}\,\text{precision} + \text{recall}.$$

The relationships among TP, FP, FN, and TN are given in Table 2.

Table 1: Analysis of user tweet length.

Measurement method	Value
Mean	58.5
Std	36.6
Min	7
25% of text length quantiles	27
50% of text length quantiles	54
75% of text length quantiles	**80**
Max	157

Table 2: Confusion matrix.

		Predicted	
		Spam	Ham
Actual	Spam	TP	FN
	Ham	FP	TN

Table 3: The results of model experiments (weiboData).

Model	Precision	Recall	F_1
AB + LR	0.868	0.829	0.848
AB + NB	0.810	0.814	0.811
AB + SVM	0.846	0.835	0.840
W2V + Bi-LSTM	0.844	0.841	0.842
AB + Bi-LSTM	0.889	0.862	0.875
AB + SA Bi-LSTM	**0.903**	**0.863**	**0.882**

6. Results and Analysis

In this section, we use 2 datasets to verify the effectiveness of the model and compared it with other approaches. The experimental results are presented in the form of Table 3 and Figure 6 for dataset weiboData and Table 4 and Figure 7 for dataset microblogPCU. In the experiment, we use logistic regression, naive Bayes, and SVM combined with ALBERT to construct the spam detection model, respectively, and prove the superiority of LSTM neural network in this task. The results show that all the methods of constructing neural networks using LSTM performance better than the traditional methods of machine learning. From Tables 3 and 4, it shows the performance of the microblogPCU dataset is better than the weiboData dataset, which may be related to the ratio of spam to nonspam in the dataset.

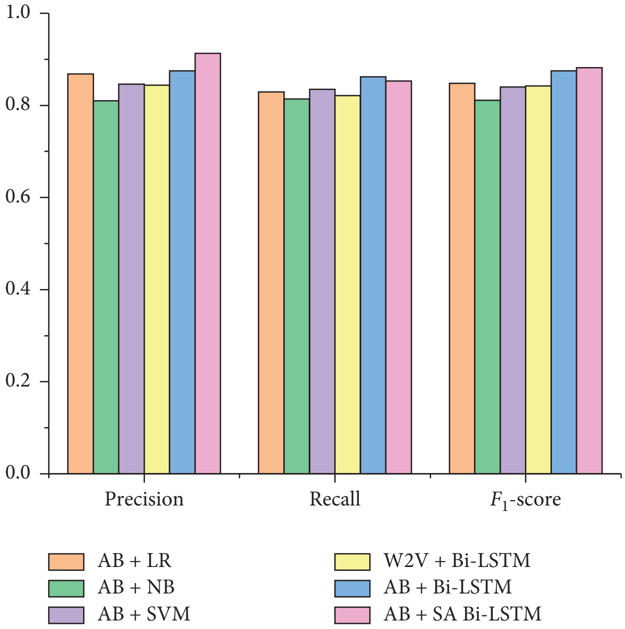


Figure 6: Comparison of model experiment results (weiboData).

Table 4: The results of model experiments (microblogPCU).

Model	Precision	Recall	F_1
AB + LR	0.865	0.821	0.842
AB + NB	0.817	0.826	0.821
AB + SVM	0.861	0.842	0.850
W2V + Bi-LSTM	0.850	0.853	0.851
AB + Bi-LSTM	0.899	0.870	0.884
AB + SA Bi-LSTM	**0.912**	**0.891**	**0.901**

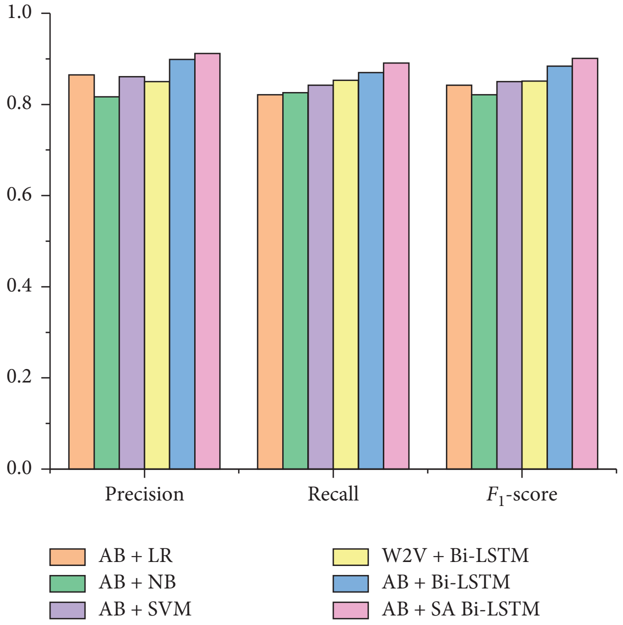


Figure 7: Comparison of model experiment results (microblogPCU).

To select the leading model in word embedding, word2vec (W2V) and Bi-LSTM are used to build the model. In Figures 6 and 7, it can be seen that ALBERT + Bi-LSTM is 2%–4% ahead of word2vec + Bi-LSTM in precision and F_1-score.

Finally, the effectiveness of self-attention is proved by comparing the AB + Bi-LSTM model and others. The experimental results are shown in Table 3. AB + SA Bi-LSTM is all ahead of other comparison models in the results and is 1% ~ 2% ahead of AB + Bi-LSTM in precision.

Figures 8 and 9 show the performance comparison of the word2vec (W2V) Bi-LSTM, ALBERT (AB) Bi-LSTM, and ALBERT + self-attention (AB + SA) Bi-LSTM. When the dataset size is the same and the parameters are set to the same value, the accuracy of the model rises with the increase of epoch. Figure 8 shows that the accuracy of W2V Bi-LSTM increases steadily with rounds before the epoch 10, fluctuates between the epoch 10–28. The accuracy of AB Bi-LSTM has been fluctuating and rising. The accuracy of the AB + SA LSTM model also increases with the increase of epoch and is stable higher than the values of the other two models after epoch 20. It can be seen from Figure 8 that the performance of AB + SA Bi-LSTM is not excellent when there are fewer rounds. This may be related to the self-attention mechanism requiring more data features for reference. It is noticed that along with the increase of training epoch, the model we proposed gradually shows its superiority. Figure 9 shows that the accuracy of the microblogPCU dataset with the increase of epochs is close to the same as the weiboData dataset. The difference is that the AB + SA Bi-LSTM model leads the other two models when the epoch is 25 on the microblogPCU.

Figures 10 and 11 show the accuracy of the model as the amount of data raises, assuming that the training epoch is

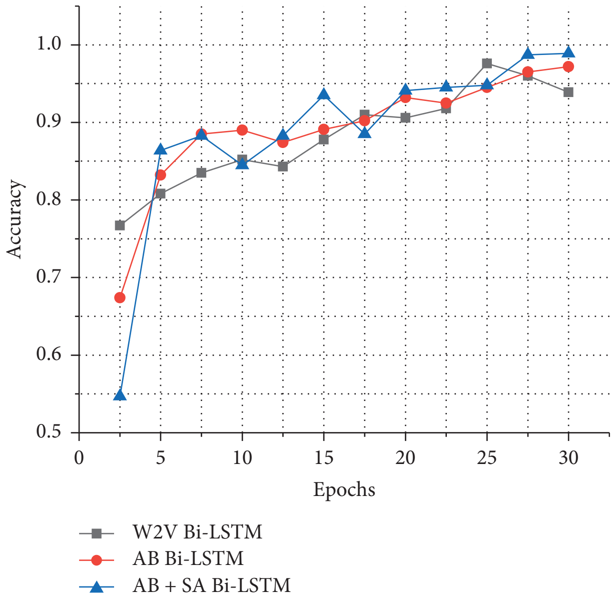


Figure 8: Accuracy as the number of epochs increases (weiboData).

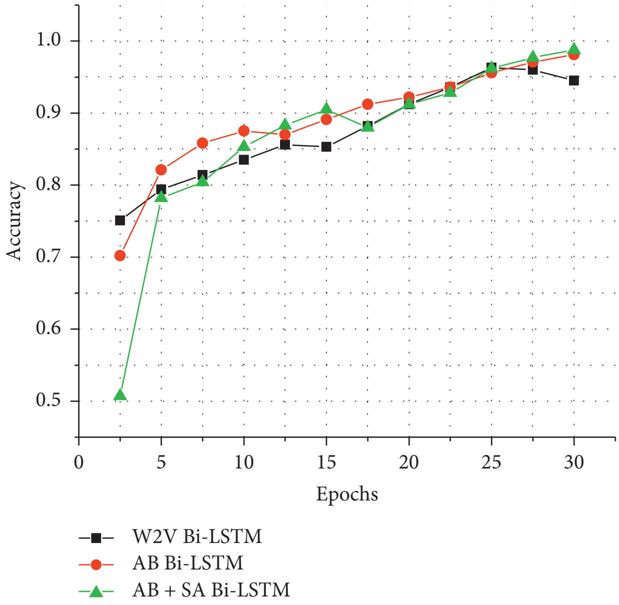


Figure 9: Accuracy as the number of epochs increases (microblogPCU).

fixed to a uniform size of 30. The figures show that the accuracy increases with the data size increasing. Figure 10 shows that the AB + SA Bi-LSTM model leads the other models at 1700 data volume, which shows that our proposed model has better performance when the data volume is larger. Figure 11 shows that the AB + SA Bi-LSTM model on the microblogPCU dataset always ahead of the other two models, which proves that the model is more sensitive to the quality of the dataset and outperforms other models on a better-quality dataset.

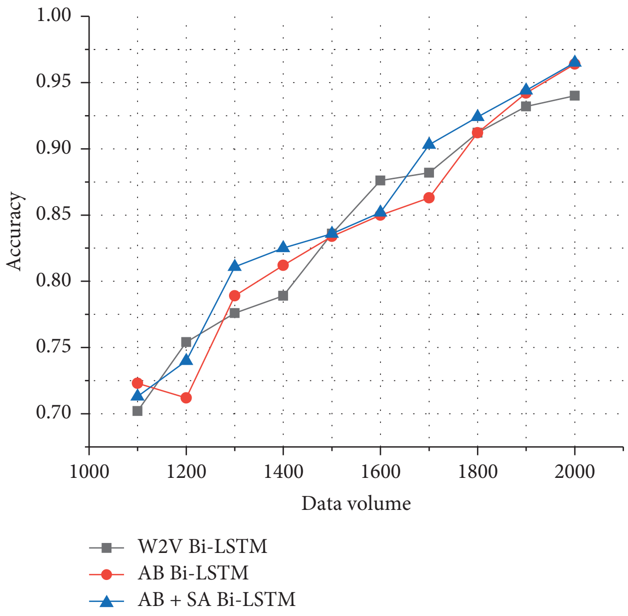


Figure 10: Accuracy as the data volume increases (weiboData).

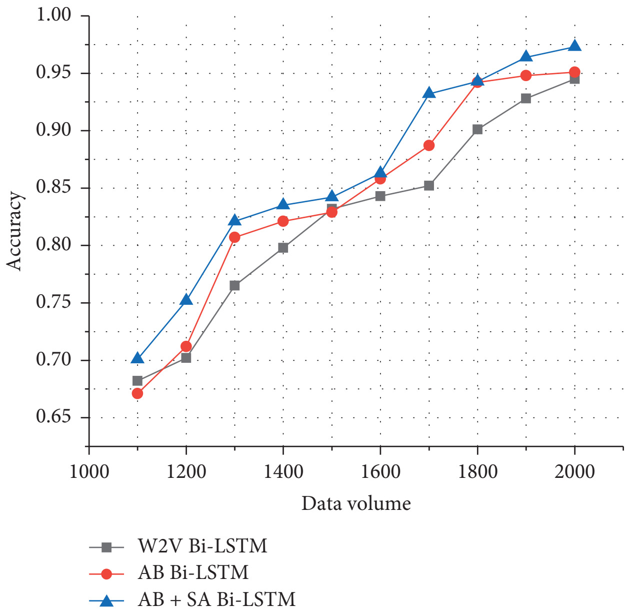


Figure 11: Accuracy as the data volume increases (microblogPCU).

7. Conclusion and Future Work

The detection of social network spam is a difficult task. The main problem in this field is that spammers have been upgrading and iterating spam text based on detection strategies, resulting in a decrease in the accuracy of detection. In response to this problem, we propose a detection method that fully considers the contextual information of text and, at the same time, take advantage of the self-attention mechanism for the shortness of text. We compared the difference between the machine learning methods with Bi-LSTM. Experiments show that Bi-LSTM performs better on spam detection task. At the same time, we also compared the effect of word2vec and ALBERT applied to the model. The experimental results show that ALBERT performs better because it considers more context information. We also proved the effectiveness of the self-attention mechanism in the model through comparative experiments. In general, according to the experimental results, our proposed model is 1% to 4% ahead of other models.

However, at the same time, the addition of the self-attention mechanism increases the computational time and computational resources required by the model. In the future, we will continue to explore how to improve the efficiency of detection with the addition of a self-attention mechanism.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation (grant nos. 61772099, 61772098, and 61802039); the Science and Technology Innovation Leadership Support Program of Chongqing (grant no. CSTCCXLJRC201917); the University Outstanding Achievements Transformation Funding Project of Chongqing (grant no. KJZH17116); the Innovation and Entrepreneurship Demonstration Team Cultivation Plan of Chongqing (grant nos. CSTC2017kjrc-cxcytd0063); and the Technology Innovation and Application Development Projects of Chongqing under grant nos. CSTC2019jscx-fxydX0086, cstc2019jscx-fxydX0089, and cstc2018jscx-mszd0132.

References

[1] G. Jain and M. Sharma, "Social media: a review," *Advances in Intelligent Systems and Computing*, vol. 433, pp. 387–395, 2016.

[2] J. f. Alqatawna, A. Madain, and A. M. Al-Zoubi, "Online social networks security: threats, attacks, and future directions," *Social Media Shaping E-Publishing and Academia*, pp. 121–132, 2017.

[3] S.-M. Al-Sayyed, W. C. Ao, and P.-Y. Chen, "On modeling malware propagation in generalized social networks," *IEEE Communications Letters*, vol. 15, no. 1, pp. 25–27, 2011.

[4] J. Chen and L. Bing, "Opinion spam and analysis," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 219–230, Palo Alto, CA, USA, 2008.

[5] H. Atefeh, M. Tavakoli, N. Salim et al., "Detection of review spam: a survey," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3634–3642, 2015.

[6] Nexgate, *State of Social Media Spam Report*, https://go.proofpoint.com/%20nexgate-social-media-spam-research-report, 2013.

[7] T. Gupta, G. Choudhary, and V. Sharma, "A survey on the security of pervasive online social networks (POSNs)," *Journal of Internet Services and Information Security*, vol. 8, no. 2, pp. 48–86, 2018.

[8] A. Branitskiy, D. Levshun, N. Krasilnikova et al., "Determination of young generation's sensitivity to the destructive stimuli based on the information in social networks," *Journal of Internet Services and Information Security*, vol. 9, no. 3, pp. 1–20, 2019.

[9] M. Kolomeets, A. Benachour, D. E. Baz et al., "Reference architecture for social networks graph analysis," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 10, no. 4, pp. 109–125, 2019.

[10] G. Xu, X. Wu, and J. Liu, "A community detection method based on local optimization in social networks," *IEEE Network*, vol. 34, no. 4, pp. 42–48, 2020.

[11] S. Liu, J. H. Yim, H. J. Lee et al., "Semantic similarity calculation method using information contents-based edge weighting," *Journal of Internet Services and Information Security*, vol. 7, no. 1, pp. 40–53, 2017.

[12] G. Xu, W. Li, and J. Liu, "A social emotion classification approach using multi-model fusion," *Future Generation Computer Systems*, vol. 102, pp. 347–356, 2020.

[13] B. Enrico and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artificial Intelligence Review*, vol. 29, pp. 63–92, 2008.

[14] O. Amayri and N. Bouguila, "A study of spam filtering using support vector machines," *Artificial Intelligence Review*, vol. 34, no. 1, pp. 73–108, 2010.

[15] M. Vangelis, L. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?" in *Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)*, Mountain View, CA, USA, July 2006.

[16] M. Soiraya, S. Thanalerdmongkol, and C. Chantrapornchai, "Using a data mining approach: spam detection on Facebook," *International Journal of Computer Applications*, vol. 58, no. 13, pp. 26–31, 2012.

[17] C. Johnson, B. Khadka, R. B. Basnet et al., "Towards detecting and classifying malicious URLs using deep learning," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 11, no. 4, pp. 31–48, 2020.

[18] F. Ahmed and M. Abulaish, "An MCL-based approach for spam profile detection in online social networks," in *Proceedings of the 2012 IEEE 11th International Conference on IEEE Trust, Security and Privacy in Computing and Communications (TrustCom)*, Liverpool, UK, 2012.

[19] A. Zoubi, H. Faris, J. Alqatawna et al., "Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts," *Knowledge Based Systems*, vol. 153, pp. 91–104, 2018.

[20] P. Tien and Z. Nur, "User identification via neural network based language models," *International Journal of Network Management*, vol. 29, no. 3, 2018.

[21] J. Ma, W. Gao, P. Mitra et al., "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Palo Alto, CA, USA, July 2016.

[22] G. Ruan and Y. Tan, "A three-layer back-propagation neural network for spam detection using artificial immune concentration," *Soft Computing*, vol. 14, no. 2, pp. 139–150, 2010.

[23] L. Z. Xiao, G. Z. Wang, and Y. Zuo, "Research on patent text classification based on Word2Vec and LSTM," in *Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, December 2018.

[24] Y. D. Luan and S. F. Lin, "Research on text classification based on CNN and LSTM," in *Proceedings of the 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China, March 2019.

[25] Y. Dong, P. Liu, Z. Zhu, and Q. Zhang, "A fusion model-based label embedding and self-interaction attention for text classification," *IEEE Access*, vol. 8, pp. 30548–30559, 2020.

[26] D. Wang, M. W. Chang, L. Kenton et al., “BERT: pre-training of deep bidirectional transformers for language understanding,” arXiv, 2017.

[27] V. Ashish, S. Noam, P. Niki et al., “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.

[28] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 7, pp. 1735–1780, 1997.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5582921, 12 pages
https://doi.org/10.1155/2021/5582921

Research Article

An Attribute-Based Access Control Policy Retrieval Method Based on Binary Sequence

Ruijie Pan, Gaocai Wang, and Man Wu

School of Computer and Electronic and Information, Guangxi University, Nanning, China

Correspondence should be addressed to Gaocai Wang; wanggcgx@163.com

Received 20 January 2021; Revised 8 February 2021; Accepted 21 March 2021; Published 5 April 2021

Academic Editor: Zhe-Li Liu

With the widespread application of new technologies, fine-grained authorization requires a large number of access control policies. However, the existing policy retrieval method applied to a large-scale policy environment has the problem of low retrieval efficiency. Therefore, this paper proposes an attribute access control policy retrieval method based on the binary sequence. This method uses binary identification and binary code to express access control requests and policies. When the policy is retrieved, the appropriate group is selected through the logical operation of the access control request and the policy binary identification. Within the group, the binary code of the access control request is matched with the binary code of all rules to find suitable rules, thereby reducing the number of matching attribute-value pairs in the rule and improving the efficiency of policy retrieval. Experimental results show that the policy retrieval method proposed in this paper has higher retrieval efficiency.

1. Introduction

The rise and development of new technologies such as cloud computing and the Internet of Things provide us with convenient data sharing, integrated computing, and other services and improve the efficiency of data processing, making full use of computing and storage resources. IoT devices increasingly prefer to synchronize all data resources to the cloud [1]. These resources contain a large amount of user privacy data, but the protection of this information by relevant agencies is not satisfactory [2]. Once such private information is leaked, it may cause immeasurable losses to individuals and organizations. For example, the private information of nearly 100,000 Westpac customers was leaked, and the data of 49 million Instagram users was exposed. These are all due to illegal access by illegal users and unauthorized access to resources by legitimate users. In recent years, as an effective means to prevent users from illegally accessing resources, access control technology has attracted the attention of domestic and foreign researchers. The object resources are protected by standardizing and restricting access to the requestor by verifying the visitor's identity information and establishing appropriate policies. This ensures that legitimate users can access and use resource services in complex network environments, while preventing illegal users from stealing and abusing resources and illegal access by legitimate users.

As the number of users increases and the access environment changes dynamically, discretionary access control, mandatory access control, and role-based access control [3] are no longer suitable for the current dynamic and real-time network environment due to their lack of dynamics and fine-graininess. However, because of its fine-grained and flexible nature, attribute-based access control (ABAC) [4] is an ideal access control solution for application scenarios such as cloud computing and the Internet of Things. Although the existing ABAC model has a relatively large advantage in the expression and formulation of security policies, the application of the ABAC model to a computing environment with a huge amount of information will have the problem of low retrieval efficiency. According to the related theories of ABAC and XACML (extensible Access Control Markup Language), when a user initiates a request to access object resources, the policy decision point will call related attributes to analyze and match all policies. This means that the evaluation of XACML-based access control policies needs to traverse all policies to make judgments, and attribute matching requires comparing all attribute information of the access control request with the attribute information of the rules in the policy set. All the above information matching includes string matching and numerical comparison. Only when all the attribute information of this rule matches the attributed-based access request (AAR) will the subject be granted the privilege to access the object. In addition, when the last attribute of the rule is detected, the rule is found to be inappropriate. If there are too many such rules, it will cause a waste of retrieval time.

This paper proposes a method of attribute access control policy retrieval based on binary sequence to improve the efficiency of policy retrieval. The rest of this paper is organized as follows: Section 1 describes the related research based on attribute access control. Section 2 introduces the related theories of ABAC. Section 3 describes the retrieval method of attribute access control policy based on the binary sequence. Section 4 focuses on the experiments and analysis. Finally, Section 5 presents the conclusion.

2. Related Work

ABAC is an access control model that allows or denies user access based on the access control policy formulated by the administrator. The successful deployment of an attribute-based access control model requires two key parts. The first is a well-defined policy description language, and the second requires an effective policy retrieval method for attribute-based access control. A good policy description language helps prevent cloud platform resource data leakage and unauthorized access, and an effective policy retrieval method can ensure that access control requests are responded to in a time when they arrive. Regarding how to define a good access control authorization policy description language, scholars have studied different access control models and application scenarios and proposed a universal access control policy language XACML [5], a graph-based application for web services policy visual description language [6], and policy description language RestPL [7] suitable for RESTful interface and so on. Among them, the most commonly used is XACML, which is suitable for ABAC models, but in actual cloud deployment, each enterprise prefers its own policy description language. In order to make the policy language suitable for different access control models, Luo et al. [8] proposed a metamodel-based access control policy description language PML and implementation mechanism PML-EM. The policy description language supports multiple access control models such as RBAC and ABAC and has better flexibility. Matthew et al. [9] proposed a rule-based ABAC policy mining algorithm for the moderate problem of privilege. This algorithm can generate the least privilege ABAC policies that balance between underprivilege and overprivilege assignment errors.

For some large organizations, the scale of ABAC policies is getting bigger and bigger, and the number of users making access requests at the same time is also increasing. Therefore, effective retrieval of these policies is essential for real-time response to user access control requests and directly affects user experience. On how to effectively retrieve these policies, researchers have achieved some results. These research work can be roughly divided into two categories. The first category is an improvement of the policy retrieval method based on the XACML standard for the policy access control language, and the second category is an improved policy retrieval method based on the next-generation access control NGAC [10] standard.

Since XACML's policy description language was proposed in 2003, there have been many studies on how to improve the efficiency of policy retrieval. In the retrieval process of the traditional policy retrieval method, when there is an access control request, all policies need to be traversed. If there are policy redundancy and conflicts, XACML supports four combination algorithms which are (1) permit-override, (2) deny-overrides, (3) first applicable, and (4) only-one-applicable [11]. Li et al. [12] proposed a retrieval method with a priority policy for access control. This method establishes an associated policy table for each access object, adopts the optimization principle of space for time, and has high retrieval efficiency. However, when the number of access objects is huge, this method needs to establish a large number of policy tables, and it is relatively difficult to establish and maintain the policy tables. Zhou et al. [13] proposed a policy retrieval method based on a prefix-based tag, which adds binary prefixes based on policy attributes to narrow the scope of policy retrieval. This retrieval method does improve retrieval efficiency. However, when the access control policy matches the attribute name of the access request, this prefix calculation causes a waste of time. In response to this problem. Liu et al. [14] added binary identification based on the attribute access control policy and then introduced a policy decision tree at the attribute value level to improve retrieval efficiency. This method has good scalability, but it has certain limitations when there are many attributes. Huang et al. [15] conducted research on the basis of [13], grouping according to the first three subject attribute values of the policy prefix and further narrowing the search scope of the policy to improve retrieval efficiency.

It is mentioned in the next-generation access control standard [16] that the policy decision point is responsible for retrieving the rules that meet the access control request in the access control policy. When performing policy retrieval, it is necessary to compare the attribute information of the access control request with the attribute information of each rule in the access control policy until a matching rule is found. This is undoubtedly time-consuming. To this end, Nath et al. [10] proposed a data structure called PolTree, which uses two variants, PolTree and N-PolTree, to store ABAC policies and reduce the number of attribute-value pair comparisons during policy retrieval. Therefore, the retrieval efficiency of the policy is improved.

Most of the policy retrieval methods mentioned above need to match the attribute value of the access control request with the corresponding attribute value of each access control rule when performing the policy retrieval. If every time the last attribute value of the rule is detected, it is found that it does not match the access control request, which will cause a waste of retrieval time. Therefore, this paper considers the processing of rule attributes and attribute values and proposes a new policy retrieval method. This method performs binary identification on the policies, groups them according to the binary identification, forms several policy groups, and performs binary coding on the policies. When accessing, first, we perform binary identification and encoding on the access control request. Then, through the logical operation of AAR and policy, the appropriate group is selected to filter out a large number of irrelevant policies. Finally, the binary code value of the access control request is logically operated with the binary code value of the rule in the group to avoid a single match of the attribute information in the policy and improve the retrieval efficiency.

3. ABAC Model

3.1. ABAC Policy Model. ABAC is an access control model that decides whether to grant the subject access to the object based on whether the attribute information of the subject, object, environment, and privilege attributes conforms to the established access control policy. We refer to [4, 17] to make the following definitions of ABAC.

Definition 1 (ABAC). ABAC can be abstracted as a quadruple (S, O, E, P). The Subject (S) represents a person or some nonhuman entity (a device that can initiate an access request), $S = \{s_1, s_2, s_3,...,s_n\}$. Objects ($O$) are usually resources that are requested to be accessed by the subject $O = \{o_1, o_2,o_3,. . .,o_m\}$. The environment ($E$) represents the context state when the visit occurs, which includes the location and time when the subject initiates the visit, $E = \{e_1, e_2, e_3,. . ..e_j\}$. Privilege (P) indicates the operations that the subject will perform on the object, including read, write, and modify. $P = \{p_1, p_2, p_3, . . .,p_k\}$. Among them, n, m, j, and k are all greater than or equal to 1.

Definition 2 (attribute). Attributes are used to describe the characteristics of the subject, object, environment, and privilege. Attributes are represented by attribute-value pairs. We use SA, EA, OA, and PA to represent the subject attribute, object attribute, environment attribute, and privilege attribute, respectively. Attr(SA), Attr(OA), Attr(EA), and Attr(PA) represent the value range of subject attribute, object attribute, environment attribute, and privilege attribute, respectively. Avsa, Avoa, Avea, and Avpa represent the attribute-value pairs of the subject, object, environment, and privilege, respectively, namely, two-tuples (SA, attr(SA)), (OA, attr(EA)), and (PA, attr(PA)).

Definition 3 (policy). ABAC policies are represented by attributes, and attribute values can be divided into discrete and continuous types. The access control policy is defined as (permit, deny) ← (Avsa, Avoa, Avea, Avpa).

Access control rules stipulate the attribute information that users must have when they want to access a particular resource. It is the basic unit of policy and the smallest policy execution unit $p = \{r_1,r_2,r_3,...,r_k\}$.

3.2. ABAC Process. The access control mechanism is composed of four key parts, which are mainly responsible for the retrieval and management of policies, and the management of attributes. They cooperate with each other to complete the authorization process for access control requests. The four key service sites shown in Figure 1 are Policy Administration Point (PAP), Policy Information Point (PIP), Policy Decision Point (PDP), and Policy Enforcement Point (PEP). According to the types of operations performed by the ABAC system, it can be divided into two stages: the preparation stage and the execution stage [17]. The dotted line in Figure 1 is an extension of the original access control model. BI is binary identification; BC is binary coding. BI and BC are both for access control requests and policies. The functions of these two modules will be described in Section 3.

4. Attribute Access Control Policy Retrieval Method Based on Binary Sequence

4.1. Build Policy and Attribute AARs Based on Binary Identification. Binary identification of ABAC policies and AARs is as follows. This method counts all the attributes that appear in the access control policy set and arranges the attributes in a specific order of subject, object, environment, and privilege. Moreover, the values of the subject, object, environment, and privilege attributes are also arranged in a certain order to ensure the validity and uniqueness of the binary identification. Attributes are divided into category attributes and noncategory attributes [18]. Category attributes refer to some decision information attribute values, usually including {permit, deny}; each rule has only one decision attribute. Noncategory attributes refer to the attribute information that needs to be measured to make a decision, including subject attributes, object attributes, environment attributes, and privilege attributes. Binary identification and encoding are for noncategory attributes.

On cloud computing and Internet of Things platforms, users request services through their respective terminal devices [19]. Therefore, we can construct the attribute information shown in Table 1 and use attribute information to construct ABAC policies (as shown in Table 2). The dimension of the binary identifier is the number of nondirectory attributes; that is, the total number of all attributes in the access control policy is the number of bits of the binary identifier. For example, the maximum number of noncategory attributes in a single rule in Table 2 is 6, so 6 binary bits are used to encode the policy.

As mentioned in Section 2, ABAC's access control process is divided into a preparation phase and an execution phase. In the preparation phase, PAP needs to perform binary identification for each rule in the policy set. When performing a policy retrieval, only binary identification of the access control request is required, and then, a logical AND operation is performed with the binary identification of the policy. When the operation result is consistent with the binary identifier of the access control rule, other attributes of the rule are checked; that is, it is judged whether the attribute of the access control request conforms to the attribute information of the rule. In order to further reduce the policy retrieval time, grouping is carried out according to the binary identification of ABAC policy. As shown in Figure 2,

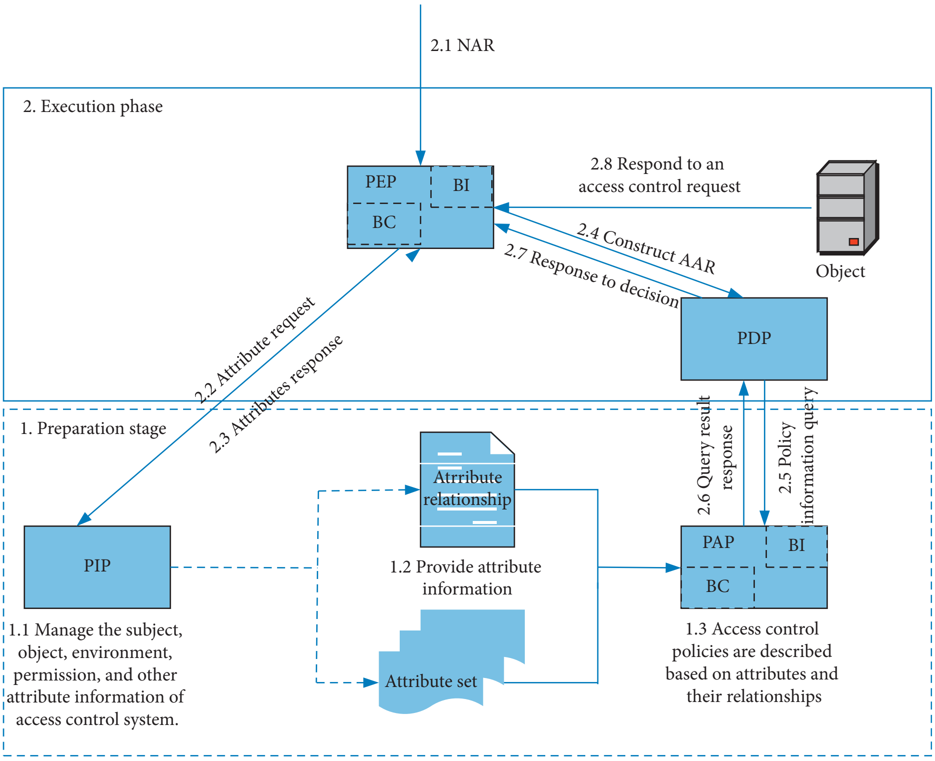

2.1 NAR
2. Execution phase
PEP
BI
BC
2.8 Respond to an access control request
Object
2.4 Construct AAR
2.7 Response to decision
PDP
2.2 Attribute request
2.3 Attributes response
1. Preparation stage
Atrribute relationship
1.2 Provide attribute information
2.6 Query result response
2.5 Policy information query
PIP
PAP
BI
BC
1.1 Manage the subject, object, environment, permission, and other attribute information of access control system.
Attribute set
1.3 Access control policies are described based on attributes and their relationships

Figure 1: ABAC framework.

Table 1: Attribute combination.

SA		OA		EA	PA
SA_Role	SA_trust	OA_type	OA_trust	EA_Network	PA_permission
Student	Low	Personal	Low	Home	Delete
Teacher	Middle	Common	Middle	Work	Read
Admin	High		High	Public	Write

Table 2: Policy set.

Rule	attr(SA)	Attr(OA)	Attr(EA)	Attr(P)	Decide	Binary identification
R_1	Student ˅ low	Personal ˅low	Home	Delete	Permit	111111
R_2	Low	Personal ˅ low	Public	Delete	Deny	011111
R_3	Student	Low	Work	Delete	Deny	100111
R_4	Student	Low	Home	Delete	Permit	100111

the root node represents all rule information of the policy set. Each node except the root node of the first level represents a group. Rules with the same binary identifier are in the same group, and rules with different identifiers are in different groups. The number of rule groups is determined by the complexity of the policy. If the complexity is n, the rule set can be divided into 2^n-1 groups at most.

The attributes of the access control rule represent the requirements for the access control request; that is, to successfully match the access control rule, the access request needs to have the attributes and attribute values required by the rule. This means that the matching of access control requests and rules is not a completely consistent match. In fact, the access control policy only detects the attributes required by the rules in the access request, and the additional attributes in the access request do not affect the success of its matching. Assuming that an access control request is (SA_Role = "student," SA_trust = "low," OA_type = "personal," OA_trust = "low,"t EA_Network = "public," PA_permisssion = "delete"), according to the matching rules, R_2 in Table 2 matches the access control request.

4.2. Binary Encoding of Attribute Information. Binary identification and grouping of ABAC policies can limit the scope of the policy to several groups that conform to the binary identification. A large number of irrelevant rules can be filtered out, thereby shortening the time for policy retrieval. However, when retrieving in this group, it is necessary to match the value of each attribute of each rule

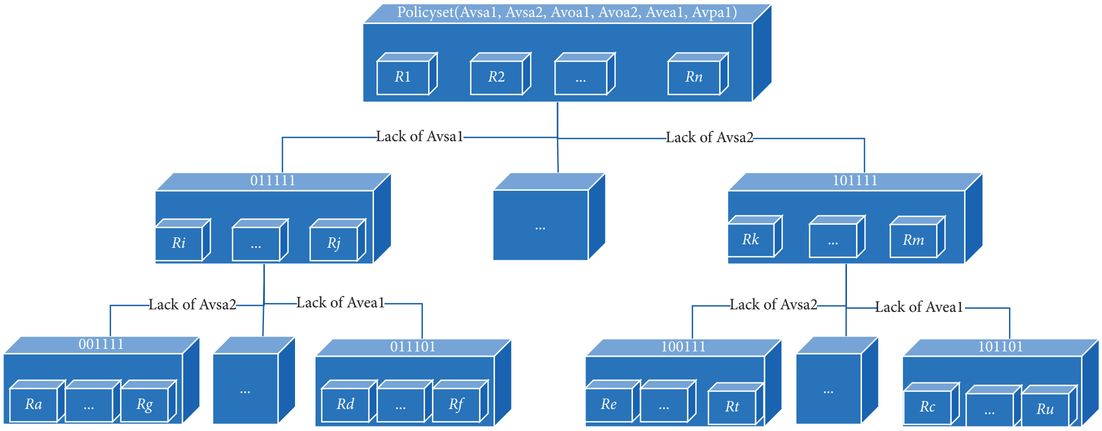


FIGURE 2: Policy set grouping.

with the corresponding value of the access control request and check whether the access control request satisfies the rule. The worst case is that every time the last attribute of the rule is detected, the rule does not match, which will cause a waste of time. If there are too many such rules, it will take too long to retrieve the policy. For this reason, we propose to binary code the access control policy of each group. In this way, only the binary code based on the attribute access control request and the binary code based on the attribute access control policy need to be logically operated to make an authorization decision, and there is no need to judge each attribute value of each rule in the policy set. Therefore, this retrieval method saves the time of policy retrieval and improves the efficiency of policy retrieval.

The attribute value based on attribute access control consists of two types, namely, discrete attribute value and continuous attribute value. For example, the value range of the attribute value of the subject attribute SA_Role mentioned above is {student, teacher, admin}, which is a discrete attribute value. In a specific access control environment, it is necessary to specify users to access resources in a certain interval, which requires setting continuous attributes. The process of mapping discrete attribute values and continuous attribute values into a binary sequence is called binary encoding of attribute values.

4.2.1. *Discrete Attribute Value Coding.*

The values of discrete attributes are independent of each other, and the range of values can be represented by enumeration. We use dummy variable encoding to encode discrete attribute values, and dummy variable encoding uses a smaller dimension to represent the value of the attribute. If there are M types of discrete variables, the dummy variable coding can represent M possible values using only the M-1 dimension.

Definition 4 (attribute encoding dimension). Different subjects, objects, environments, and privilege entities have different attributes, and different attributes have different attribute values. NUM (s_i), NUM (o_i), NUM (e_i), and NUM (p_i) represent the number of attributes owned by s_i, o_i, e_i, and p_i, respectively. NumV (NUM(sa$_i$)), NumV(NUM(oa$_i$)), NumV(NUM(ea$_i$)), and NumV(NUM(pa$_i$)) represent the number of attribute values owned by subject attribute name sa$_i$, object attribute name oa$_i$, environment attribute name ea$_i$, and permission attribute name pa_i, respectively. The dimension $V_{s_i}V_{o_i}$, V_{e_i}, and V_{p_i}required to encode subject s_i, object o_i, environment e_i, and privilege p_i can be identified as

$$V_{s_i} = \sum_{i=0}^{n} \mathrm{Num}V\left(\mathrm{Num}\left(sa_i\right)\right) - \mathrm{Num}\left(s_i\right), \qquad (1)$$

$$v_{o_i} = \sum_{i=0}^{m} \mathrm{Num}V\left(\mathrm{Num}\left(oa_i\right)\right) - \mathrm{Num}\left(o_i\right), \qquad (2)$$

$$V_{e_i} = \sum_{i=0}^{c} \mathrm{Num}V\left(\mathrm{Num}\left(ea_i\right)\right) - \mathrm{Num}\left(e_i\right), \qquad (3)$$

$$V_{\mathrm{p_i}} = \sum_{i=0}^{d} \mathrm{Num}V\left(\mathrm{Num}\left(pa_i\right)\right) - \mathrm{Num}\left(p_i\right), \qquad (4)$$

where n, m, c, and d are the number of attributes contained in s_i, o_i, e_i, and p_i. The dimension of the binary identifier required by the encoding rule is $V_{\text{total}} = V_{s_i} + V_{o_i} + V_{e_i} + V_{p_i}$. Assuming that s_i has two attributes, sa$_1$ and sa$_2$, the value range of sa_1 is {sa_1v_0, sa_1v_1, sa_1v_2,ϕ}, and the value range of sa_2 is {sa_1v_0, sa_1v_1, ϕ}, and the dimension required to encode s_i is $4+3-2=5$.

Definition 5 (attribute encoding rules). Arrange the attribute names of all attributes in a specific order, and the attribute values contained in the attributes also need to be arranged in a specific order to form a large attribute array. If a specific attribute value appears in the rule, the corresponding position is set to 1.

If the subject, object, environment, and privilege attributes appear in a rule at the same time, the attributes of the rule are arranged in a specific order of subject, object, environment, and privilege attributes. If the attribute value ha$_i$v$_j$ appears in the rule, the Pth position of V_{total} is set to 1 according to

$$\mathrm{P} = \begin{cases} \mathrm{Num}V\left(\mathrm{Num}\left(sa_1\right)\right) + \ldots. + \mathrm{Num}V\left(\mathrm{Num}\left(sa_{i-1}\right)\right) + j \quad ha_i v_j \in SA, \\ V_{s_i} + \mathrm{Num}V\left(\mathrm{Num}\left(oa_1\right)\right) + \ldots. + \mathrm{Num}V\left(\mathrm{Num}\left(oa_{i-1}\right)\right) + j \quad ha_i v_j \in OA, \\ V_{\mathrm{s_i}} + V_{\mathrm{o_i}} + \mathrm{Num}V\left(\mathrm{Num}\left(\mathrm{e}a_1\right)\right)) + \ldots. + \mathrm{Num}V\left(\mathrm{Num}\left(\mathrm{e}a_{i-1}\right)\right) + j \quad ha_i v_j \in EA, \\ V_{\mathrm{s_i}} + V_{\mathrm{o_i}} + V_{\mathrm{e}_i} + \mathrm{Num}V\left(\mathrm{Num}\left(pa_1\right)\right) + \ldots. + \mathrm{Num}V\left(\mathrm{Num}\left(pa_{i-1}\right)\right) + j \quad ha_i v_j \in PA. \end{cases} \tag{5}$$

The encoding of subject s_{i} according to the encoding rules is shown in Table 3.

4.2.2. Continuous Attribute Coding. For continuous attribute values that cannot be enumerated, we first discretize them and then encode them. As shown in Figure 3, assuming that the attribute named sa_3 is a continuous attribute, the discretization process of sa_3 is as follows.

(1) Traverse all the rules, divide the value range of sa_3, remove duplicate attribute values, and arrange them in ascending order. Suppose we get{sa_3v_1, sa_3v_2,..., $\mathrm{sa}_3v_{\mathrm{n}}$}.

(2) Remove the smaller attribute value that obtains the same access privileges of the same object in the rule, and keep the largest attribute value. For example, if the attribute values sa_3v_1, sa_3v_2, and sa_3v_3 have read privilege for the same object resource content at the same time, sa_3v_1 and sa_3v_2 are deleted and sa_3v_3 is retained. Assume that the set of these largest attribute values is {sa_3v_3, sa_3v_9,..., $\mathrm{sa}_3v_{\mathrm{n}}$}.

(3) Set labels L_1, L_2,...,L_{m} with sa_3v_3, sa_3v_9,..., $\mathrm{sa}_3v_{\mathrm{n}}$ as the thresholds, respectively, where m is the number of elements in {sa_3v_3, sa_3v_9,..., $\mathrm{sa}_3v_{\mathrm{n}}$}. The value ranges of L_1, L_2,..., L_{m} are {0, sa_3v_3}, {sa_3v_3, sa_3v_9},..., {$\mathrm{sa}_3v_{\mathrm{g}}$, $\mathrm{sa}_3v_{\mathrm{n}}$}.

(4) Use L_1, L_2,...,L_{m} to replace the attribute value of sa_3 in the rule.

After the continuous attribute value is converted to the discrete attribute value, the encoding method is completely consistent with that of the discrete attribute, and the description will not be repeated here.

4.3. Analysis of Policy Retrieval Method Based on Binary Sequence. We make the following definitions for the retrieval rules of the binary sequence-based policy retrieval method.

Definition 6 (group selection principle). Perform a logical AND operation based on the binary identification of the attribute access control request and the group number of the rule grouping. If the calculation result is consistent with the group number, the group is a suitable group; otherwise,e the group is not suitable.

Definition 7 (rule selection principle). Perform logical AND operation based on the binary code of the attribute access control request and the rule binary code. If the result is consistent with the binary code of the rule, the rule is the rule to be found; otherwise, the access control request does not comply with the rule.

In order to further describe the policy retrieval method mentioned in this article, we take the policy set in Table 2 as an example, select the attribute values in Table 1 to construct the attribute-based access control request, aar_1 = {SA_role = student, SA_trust = low, OA_trust = low, EA_Network = -work, PA_permission = delete}, and complete the retrieval process.

In the access control preparation phase, the preparation work that PAP needs to complete is as follows: the results of the binary identification and coding of the rules in Table 2 according to the coding rules proposed in Section 4.2 are (R_1, 111111, 10010010100100100), (R_2, 011111, 00010010100001 100), (R_3, 100111,10000000100010100), and (R_4,100111, 10000000100100100). The group numbers of R_1 and R_2 are 111111 and 011111, respectively. The group number of R_3 and R_4 is 100111, and they are in the same group.

The access control execution phase is divided into two phases: processing the attribute-based access control request and retrieving the rule set. Processing of access control requests is as follows: the result of the binary identification of aar_1 according to the binary identification principle proposed in Section 4.1 is 110111; according to the encoding method mentioned in Section 4.2, the result of the binary encoding of aar_1 is (aar_1, 10010000100010100).

Retrieve the rule set: the binary identification of aar_1 and the group number AA are logically ANDed according to the principles defined in Definition 6, namely, 110111 and 111111 = 110111≠111111, indicating that aar_1 does not meet the attribute combination requirements of the group. Query the group with the group number 011111, that is, 110111 and 011111 = 010111≠011111, so aar_1 does not meet the attribute combination requirements of the group, and the group is filtered out. Query the rule group with the group number 100111, 110111, and 100111 = 100111, indicating that this group meets the requirements, which needs further retrieval. The result of logical AND operation between the binary code of the R3 rule and the binary code of the access control request is equal to the binary code of the rule, that is, 10000000100010100 and 10010000100010100 = 1000000010 0010100. According to Definition 5, this rule meets the requirements. Because R_3's decision attribute is deny, the decision result of the PDP will be to deny this access, and this policy retrieval ends. The specific policy retrieval process is shown in Figure 4.

In the preparation phase, PAP performs binary identification and coding on all access control policies and groups them according to the binary identification. When there is an access control request, the PEP obtains the attribute information from the PIP to construct the AAR and performs binary

Table 3: Binary encoding of attributes.

Subject attribute value range	Binary code
$\mathrm{attr}(\mathrm{sa}_1) = \{\mathrm{sa}_1 = sa_1 v_1\}^{\vee}\mathrm{attr}(\mathrm{sa}_2) = \{\mathrm{sa}_2 = sa_1 v_0\}$	01010
$\mathrm{attr}(\mathrm{sa}_1) = \{\mathrm{sa}_1 \neq sa_1 v_1\}^{\vee}\mathrm{attr}(\mathrm{sa}_2) = \{\mathrm{sa}_2 = \mathrm{sa}_2 v_0\}$	01101
$\mathrm{attr}(\mathrm{sa}_1) = \{\mathrm{sa}_1 = \{ sa_1 v_1, \mathrm{sa}_1 v_2 \}^{\vee}\mathrm{attr}(\mathrm{sa}_2) = \{\mathrm{sa}_2 = \mathrm{sa}_2 v_0\}$	01110
$\mathrm{attr}(\mathrm{sa}_1) = \{\mathrm{sa}_1 = sa_1 v_1\}$	00010
$\mathrm{attr}(\mathrm{sa}_2) = \{\mathrm{sa}_2 = \mathrm{sa}_2 v_0\}$	01000

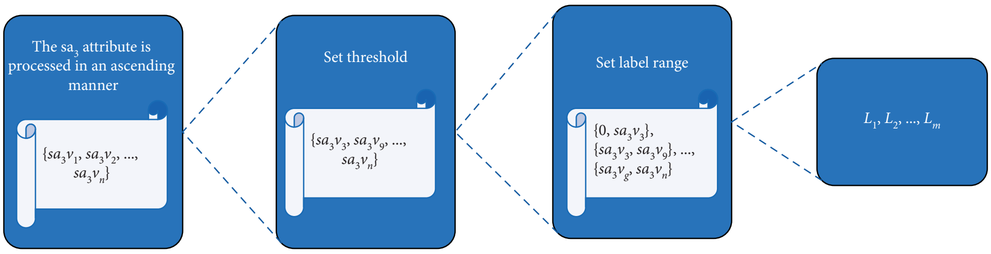

The sa_3 attribute is processed in an ascending manner
$\{sa_3v_1, sa_3v_2, \ldots, sa_3v_n\}$
Set threshold
$\{sa_3v_3, sa_3v_9, \ldots, sa_3v_n\}$
Set label range
$\{0, sa_3v_3\}$, $\{sa_3v_3, sa_3v_9\}, \ldots,$ $\{sa_3v_g, sa_3v_n\}$
$L_1, L_2, \ldots, L_m$

Figure 3: Continuous attribute discretization process.

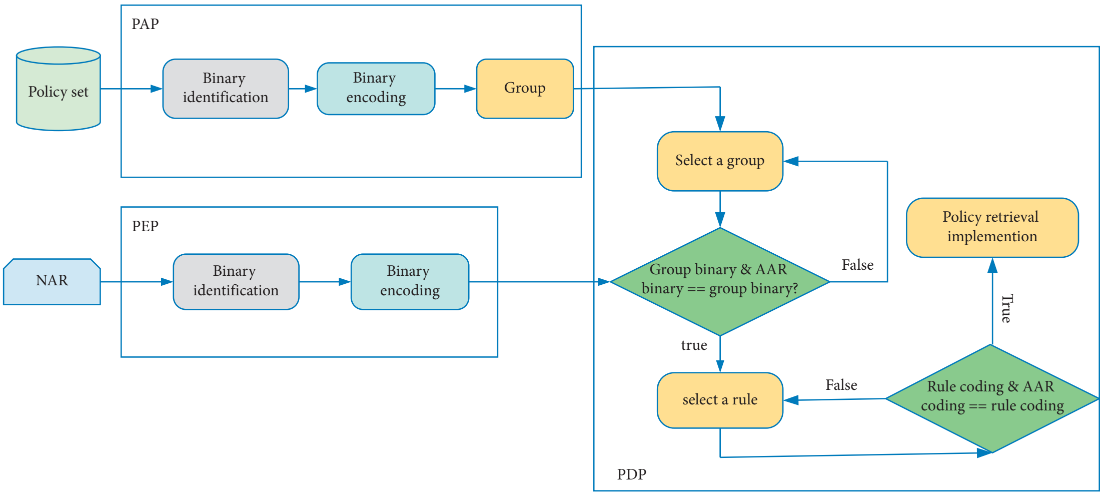


Figure 4: Policy retrieval mechanism based on the binary sequence.

identification and encoding of the AAR. The PDP selects a group from the rule group to make a logical AND operation of its binary identifier and the binary identifier of the access control request and judges whether the result is equal to the binary identifier of the AAR. If they are not equal, then the binary identification of the next group is judged. If they are equal, make the binary code of each rule in the group and the binary code of the access control request do a logical AND operation, and judge whether it is equal to the binary code of the rule. If they are equal, it indicates that the rule is the rule to be searched; otherwise, the binary code of the next rule in the group is logically operated.

The core Algorithm 1 of the attribute access control policy retrieval method based on binary sequence is as follows.

5. Experimental Results and Analysis

In order to evaluate the efficiency of the retrieval method based on the binary sequence policy, this paper uses the C++ language to write the test code in Qt Creator on the Win10 platform and uses MATLAB as the data analysis tool. Evaluating the retrieval method proposed in this paper requires a large number of ABAC policies and AARs. Unfortunately, due to the confidentiality of the access control policy, we cannot obtain real industry data. Therefore, referring to the attribute information mentioned above, a simple policy and AAR generator were implemented, and 4000 policies and 2000 AARs were generated. In order to ensure the validity of binary identification, we refer to the data classification method in [18] to standardize policies and access control requests. The experiment in this paper uses these data as the basis to test the efficiency of the binary sequence-based attribute access control strategy retrieval method in terms of policy preprocessing, policy evaluation time, and total strategy retrieval time. At the same time, in order to ensure the generality of the test results, the data obtained in the following experiments are the results of performing 10 experiments and averaging them. The experimental environment is as follows: CPU: 11th Gen Intel(R)Core(TM)i5-

```
Input: AAR
Output: the decision
//: Binary identification and encoding of AAR
Init(AAR.policy_groupbinary);
Init(AAR.policy_code);
for($i = 0$;$i <$ AAR.policy_groupbinary.size(); $i$++) do
  if(AAR[i]) exist then
    AAR.policy_groupbinary.setposition($p$) = 1;
  end if
end for
for($j = 0$; $j <$ AAR.policy_code.size(); $j$++) do
  if(AAR[j].attrvale) exit then
    $a$ = getposition(AAR[j].attrvale);
    AAR.policycode.setposition(a) = 1;
  end if
end for
//policy retrieval
for($k = 0$; $k <$ Group.size; $k$++) do//
if (Group[k].groupbinary&AAR.groupbinary) = = Group[k].groupbinary) then
  for($g = 0$; $g <$ Group[k].size; $g$++) do
    if (Group[k].ruleset[g].policycode&AAR.policy_code = = (Group[k].ruleset[g].policycode) then
      Dodecide();
      break;
    end if;
  continue;
  end for;
  continue;
end if
end for
```

Algorithm 1: Policy retrieval algorithm based on binary identification.

1135G7@2.40GHZ 2.42 GHz,RAM:16.00 GB; Qt Creator 3.3.0(opensource) based on Qt 5.4.0(MSVC 2010, 32 bit), MATLAB version: 8.6.0.267246 (R2015b).

SG-PRM and AG-PRM in the following experiments refer to the prefix-based policy retrieval method mentioned in [13] and the attribute grouping-based access control policy retrieval method mentioned in [15]. B–S-PRM refers to the strategy retrieval method proposed in this article. Experimental comparisons are based on the same access control policy and access control request. Policy complexity refers to the number of attribute key-value pairs contained in each rule in the policy. The complexity of the policy used in all experiments is 6.

5.1. Policy Preprocessing Time. The policy preprocessing time is completed in the preparation phase of the access control framework diagram in Figure 1, mainly deploying policy attributes. The policy preprocessing time of the policy retrieval method proposed in this paper refers to the time that PAP adds binary identification, grouping, and coding to each rule in the policy set. SG-PRM performs binary identification of the policy in the policy preprocessing stage. AG-PRM performs binary identification and grouping of policies in the policy preprocessing stage. It can be seen from Figure 5 that when the number of policies is set to 500, 1000, 1500, 2000, and 2500, the retrieval time of B–S-PRM, SG-PRM, and AG-PRM policies all increases with the increase of the number of policies. Among them, the policy preprocessing time of the AG-PRM and SG-PRM policy retrieval methods is relatively close and lower than that of the B–S-PRM policy retrieval method. This is because the

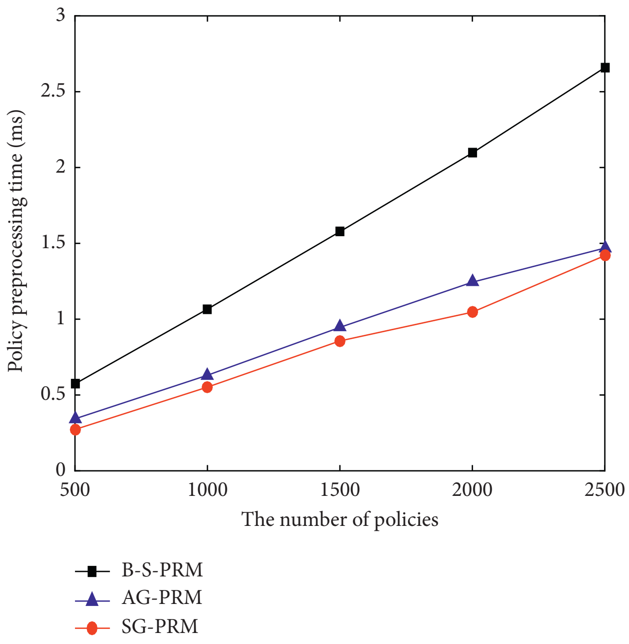
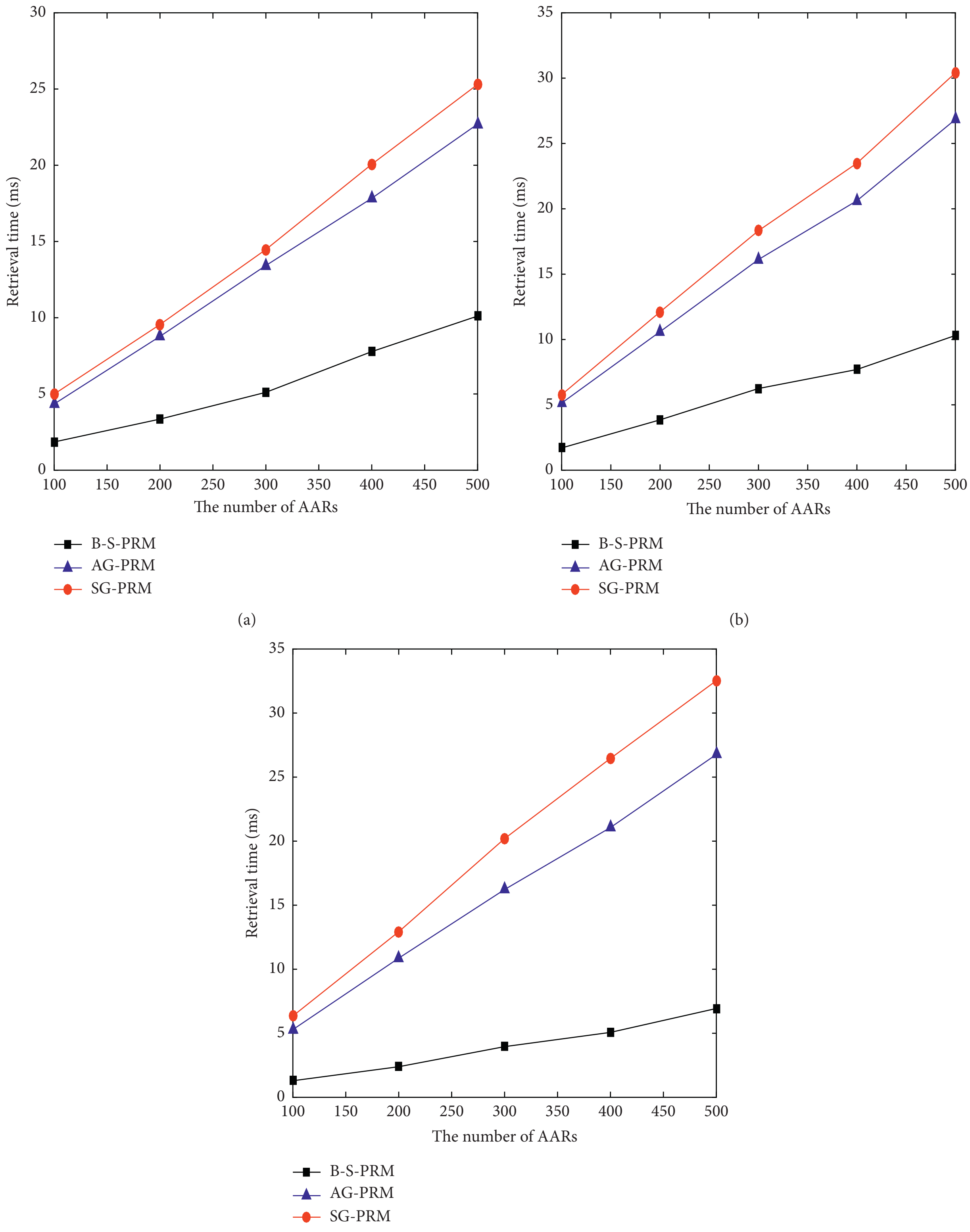


Figure 5: Policy preprocessing.



(a)

(b)

(c)

Figure 6: Changes in policy evaluation time. (a) The number of policies is 2000. (b) The number of policies is 2500. (c) The number of policies is 3000.

policy retrieval method proposed in this paper adds a binary encoding process to the first two policy retrieval methods. However, the preprocessing of the attribute-based access control strategy occurs in the preparation phase of the access control system. Therefore, the policy retrieval method in this article does not affect the user's real-time experience.

5.2. Policy Evaluation Time. Policy evaluation refers to the completion of the PDP in the execution phase in Figure 1. It mainly refers to the process of PDP retrieval in accordance with the rules of the AAR. Figures 6(a)–6(c) show the change of the policy evaluation time when the number of AARs increases under the condition of different numbers of policies. The policy complexity of this set of experiments is 6. The number of policies is 2000, 2500, and 3000, respectively. SG-PRM performs a logical OR operation between the prefix of the access control request and the prefix of the policy in the policy retrieval phase, and the result of the operation is consistent with the prefix of the policy and then matches whether the attribute information matches. AG-PRM performs attribute-based grouping of policy sets to reduce the scope of policy retrieval. It can be seen from Figures 6(a)-6(c) that under different access control policy conditions, as the number of access control requests increases, the policy evaluation time of these three policy retrieval methods gradually increases. Among them, the policy evaluation time of the strategy retrieval method (B–S-PRM) proposed in this paper shows a downward trend as the number of policies increases. When the number of strategies is 3000, the evaluation time of the BS-PRM strategy is about half of that when the number of strategies is 2000, and the fluctuation range of the policy evaluation time is gradually reduced, while the policy evaluation time of the other two policy retrieval methods has not changed significantly. The strategy evaluation time of SG-PRM fluctuates between 4 and 32 ms, the policy evaluation time of AG-PRM fluctuates between 4 and 27 ms, and the B–S-PRM fluctuates between 1 and 11. It can be seen that the retrieval time of the policy retrieval method proposed in this paper is more stable, and the policy retrieval efficiency of B–S-PRM is about 3 times that of SG-PRM and AG-PRM. This shows that the policy retrieval method proposed in this paper can be applied to high policy environments.

5.3. Total Retrieval Time. The total retrieval time based on the attribute access control policy includes the time for PEP to process AAR and the time for PDP to evaluate the policy. As shown in Figures 7 and 8 , the total number of policies is 3000. The complexity of this set of experimental policies is 6. With the increase in the number of access control requests, the processing time of the three policy retrieval methods for access control requests has gradually increased and is close to linear growth. The processing time of the policy is all milliseconds. Among them, the policy retrieval method proposed in this paper takes about twice the processing time of access control requests than SG-PRM and AG-PRM. This is because, in the process of access control, SG-PRM adds a prefix to the access control request, BS-PRM needs to binary code and identify the policy, and AG-PRM only needs to add the group identification and prefix identification to the access control request. However, it can be seen from Figure 8 that although the strategy retrieval method proposed in this article has a slower processing time for AAR in PEP, its total retrieval time has dropped significantly. Among them, the policy retrieval efficiency of B–S-PRM is about 4 times that of SG-PRM and 3.5 times that of AG-PRM. As the number of access control requests increases, the policy retrieval method proposed in this paper has more advantages and the strategy retrieval time becomes more stable. Because when the policy increases, unlike the other two policy retrieval methods, the attribute access control policy retrieval method

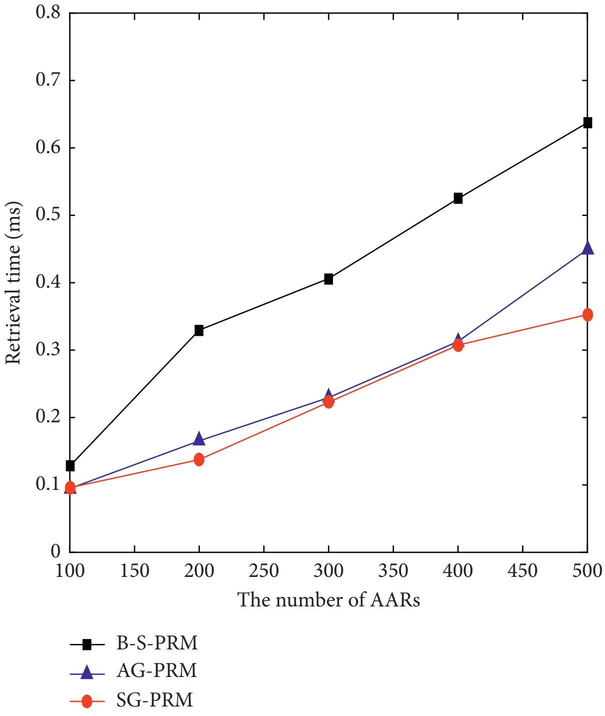


Figure 7: The processing time of the AARs.

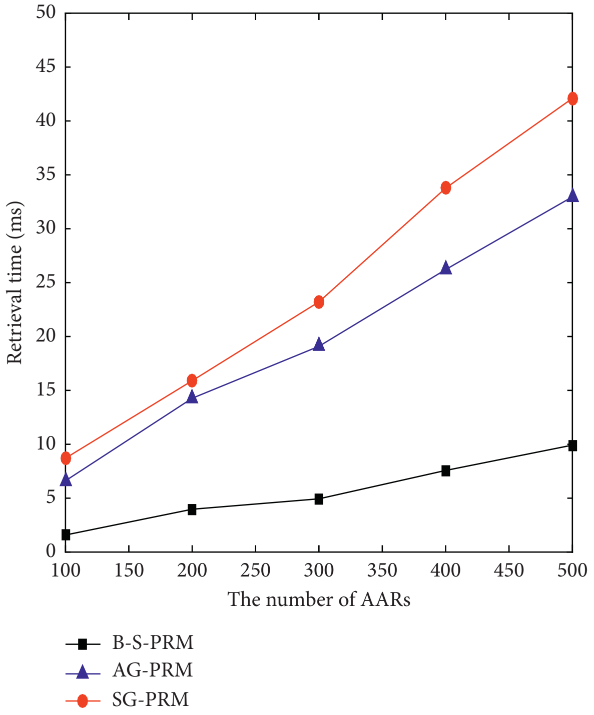


Figure 8: The total retrieval time.

based on binary sequence does not traverse the attribute-value pairs in the rule but chooses the appropriate grouping to perform logical operations on the binary code in the group. Therefore, the retrieval efficiency is greatly improved.

To sum up, although the policy retrieval method proposed in this paper takes longer than the other two policy retrieval methods in policy preprocessing time, the policy retrieval time is significantly shortened and the retrieval efficiency advantage is obvious. With the increase of the policy scale, the retrieval time of the policy retrieval method proposed in this paper has not changed significantly. In the deployment of attribute-based access control in cloud computing and Internet of Things environments, when a user's access control request comes, this method can shorten the user's waiting time and respond to the user's access control request in real time.

6. Conclusions

With the rise of cloud computing and Internet of Things technologies, today's computing environment is becoming more and more massive and dynamic, with a huge scale of users and resources, and real-time changes in the access environment of subject and object. ABAC is widely used as an effective technology to protect object resources. Users want to get the privilege to access object resources in a relatively short time. However, the existing retrieval methods based on attribute access control policies have certain deficiencies when applied to large-scale data. To solve this problem, this paper proposes an attribute access control policy retrieval method based on the binary sequence. This method uses one-hot and dummy variable encoding methods to perform binary identification and encoding of AAR and policies and group the policies according to the binary identification. AAR's binary identification and group binary identification perform a logical AND operation to select groups that meet the requirements and filter out a large number of irrelevant policies. Then, the binary code of the AAR and the binary code of the rules in the group do logical operations to select the appropriate rules, which reduces the process of matching the attributes of the rules in the policy set with the AAR attributes. The experimental results also verify that the policy retrieval method proposed in this paper has lower policy retrieval time and higher retrieval efficiency. [20]

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under Grant no. 62062007.

References

[1] T. Wang, J. Zhou, A. Liu, M. Z. A. Bhuiyan, G. Wang, and W. Jia, "Fog-based computing and storage offloading for data synchronization in IoT," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4272–4282, 2019.

[2] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, "A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4831–4843, 2019.

[3] K. R. S. Yaira, A. D. Steven, and S. B. Mohammed, "A service-based RBAC & MAC approach incorporated into the FHIR standard," *Digital Communications and Networks*, vol. 5, no. 4, pp. 214–225, 2019.

[4] D. Servos and S. L. Osborn, "Current research and open problems in attribute-based access control," *ACM Computing Surveys*, vol. 49, no. 4, pp. 1–45, 2017.

[5] X. Oasis, *eXtensible Access Control Markup Language XACML Version 3.0*, OASIS Standard, Burlington, MA, USA, 2013.

[6] D. Nabil, H. Slimani, H. Nacer et al., "ABAC conceptual graph model for composite web services," in *Proceedings of the 2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, pp. 36–41, Marrakech, Morocco, 2018.

[7] Y. Luo, Q. N. Shen, and Z. H. Wu, "A novel access control specification language and its permission classification," *Journal of Computers*, vol. 41, no. 6, pp. 969–986, 2018.

[8] Y. Lu, Q. N. Shen, and Z. H. Wu, "Access control policy specification language based on metamodel," *Journal of Software*, vol. 31, no. 2, pp. 439–454, 2020.

[9] M. W. Sander and Y. Chuan, "Mining least privilege attribute based access control policies," in *Proceedings of the 2019 Annual Computer Security Applications Conference (ACSAC '19)*, pp. 404–416, Tucson, AZ, USA, 2019.

[10] R. Nath, S. Das, and S. Sural, "PolTree: a data structure for making efficient access decisions in ABAC," in *Proceedings of the 24th ACM Symposium, ACM*, pp. 25–35, Toronto, Canada, 2019.

[11] M. Mejri, H. Yahyaoui, M. Azzam et al., "A rewriting system for the assessment of XACML policies relationship," *Computers & Security*, vol. 97, pp. 1–12, 2020.

[12] H. H. Li, M. F. Dong, and F. Fan, *A Retrieval Method Constructed By Access Control With Priority Policy*, China, 2018.

[13] J. S. Zhou and Y. S. Zhang, "Policy retrieval method based on sign," *Computer Engineering and Design*, vol. 36, no. 11, pp. 2943–2947, 2015.

[14] M. P. Liu, Y. Cheng, H. Li et al., "An Efficient attribute-based access control (ABAC) policy retrieval method based on attribute and value levels in multimedia networks," *Sensors*, vol. 20, no. 6, pp. 1–15, 2020.

[15] M. R. Huang and B. Ou, "Access control policy retrieval method based on attribute grouping," *Application Research of Computers*, vol. 37, no. 10, pp. 1–7, 2019.

[16] V. Hu, D. F. Ferraiolo, D. R. Kuhn, R. N. Kacker, and Y. Lei, "Implementing and managing policy rules in attribute based access control," in *Proceedings of the 2015 IEEE 16th International Conference on Information Reuse and Integration*, pp. 518–525, San Francisco, CA, USA, 2015.

[17] L. Fang, L. H. Yin, Y. C. Guo et al., "A survey of technologies in attribute-based access control scheme," *Chinese Journal of Computers*, vol. 40, no. 7, pp. 1681–1698, 2017.

[18] R. A. Shaikh, K. Adi, and L. Logrippo, "A data classification method for inconsistency and incompleteness detection in access control policy sets," *International Journal of Information Security*, vol. 16, no. 1, pp. 91–113, 2017.

[19] T. Wang, M. Z. A. Bhuiyan, G. Wang, L. Qi, J. Wu, and T. Hayajneh, "Preserving balance between privacy and data integrity in edge-assisted Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2679–2689, 2020.

[20] Z. Y. Zhao and L. Sun, "Attribute-based access control with dynamic trust in a hybrid cloud computing environment," in *Proceedings of the International Conference on Cryptography, Security and Privacy*, pp. 112–118, ACM, New York, NY, USA, 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5542543, 13 pages
https://doi.org/10.1155/2021/5542543

Research Article

A Security Log Analysis Scheme Using Deep Learning Algorithm for IDSs in Social Network

Ming Zhong, Yajin Zhou, and Gang Chen

Zhejiang University, Computer Science and Technology College, 310018 No. 38 Zheda Road, Xihu District, Hangzhou City, Zhejiang Province, China

Correspondence should be addressed to Ming Zhong; tracym@zju.edu.cn

Received 5 February 2021; Revised 20 February 2021; Accepted 13 March 2021; Published 23 March 2021

Academic Editor: Hao Peng

Due to the complexity of the social network server system, various system abnormalities may occur and in turn will lead to subsequent system failures and information losses. Thus, to monitor the system state and detect the system abnormalities are of great importance. As the system log contains valuable information and records the system operating status and users' behaviors, log data in system abnormality detection and diagnosis can ensure system availability and reliability. This paper discloses a log analysis method based on deep learning for an intrusion detection system, which includes the following steps: preprocess the acquired logs of different types in the target system; perform log analysis on the preprocessed logs using a clustering-based method; then, encode the parsed log events into digital feature vectors; use LSTM-based neural network and log collect-based clustering methods to learn the encoded logs to form warning information; lastly, trace the source of the warning information to the corresponding component to determine the point of intrusion. The paper finally implements the proposed intrusion detection method in the server system, thereby improving the system's security status.

1. Introduction

The development and rise of social networks have changed our way of life, realized the interconnection between people, accelerated information dissemination speed, and changed social communication. While enjoying the convenience that social networks bring to our lives, we also need to protect information. If there is a social network security issue, our personal information will be stolen. The data stored in social network servers and the services provided are themselves potential targets for various attacks. Due to their diversity and particularity, these attacks may have disastrous consequences. In this context, social network server security has become a major challenge, and research in this area is also increasing. The development of various tools and mechanisms ensures that the safety level is improved and meets modern life requirements. These methods include the IDS (intrusion detection system). The IDS is a tool used to detect network attack attempts and is used to identify abnormal activities and behaviors designed to interfere with the normal operation of the system [1].

Various logs are generated during the operation of the host system. The logs record the status of the computer during operation and various operations performed by the system. They are an excellent resource of information for online data monitoring and anomaly data detection. Therefore, the audit of system logs can be used as the host, an essential means of anomaly detection. There are already various security audit systems in the market, such as log audit systems and IDS. These systems can implement log collection, auditing, and abnormal behavior mining functions [2]. However, in the actual use, due to the differences in logs and the single backwardness of log auditing methods, these systems are often only suitable for specific types of hosts, and the abnormal behaviors that can be detected are not comprehensive and accurate.

Data mining capabilities have been further improved with AI and machine learning development, and research results based on machine learning into log auditing continue to appear. However, for different host systems and processes, the types of logs generated are inconsistent. Traditional machine learning detection methods need to use different feature extraction methods for different types of logs, and users must have professional background knowledge to better extract logs' characteristic information. In reality, the types and grammar of logs are constantly updated, and a certain method cannot be directly applied to multiple systems, and it takes much workforce to update and match. The relatively advanced one is to use deep learning for anomaly detection. The increasing amount of system log data is sufficient for the deep learning model to perform learning processing. When the parameters are appropriate, there is almost no need to extract features manually, and the deep learning model can complete log detection well.

For this purpose, in this paper, we propose a log analysis method based on deep learning and apply it in the IDS to ensure the server's security. Figure 1 shows the scenario of attackers and defenders using the IDS. The attackers on the Internet will use various ways to intrude the social network server and then leak the users' privacy. The IDS monitors the operating conditions of the social network servers in accordance with certain security policies and discovers as much as possible various attack attempts, attack behaviors, or attack results to ensure the confidentiality, integrity, and availability of network system resources [3]. The proposed method in this paper will ensure the users' privacy for the social network. This paper's main work is to form a log parser, build the feature extraction neural network, and finally implement the IDS.

The rest of this paper is organized as follows. Section 2 summarizes the related work and emphasizes log analysis and AI methods. Section 3 proposes the deep learning methods for the IDS. Section 4 shows the experiments and results. Finally, Section 5 concludes the paper.

2. Related Works

2.1. Social Networks' Security. The social network is a platform for human beings to transmit information and conduct social conversations on the Internet. Many researchers show that, with the rapid development of social networks, the way of information exchange between Internet users has also undergone tremendous changes, and social networks play a more unceasingly important role in people's everyday life [4]. People can follow the latest developments of friends and celebrities on Sina Weibo or get other people's attention by sharing information on WeChat Moments. Social networks provide users with the following main functions:

(1) Creating and sharing user information which is public or semipublic within a certain scope

(2) Providing a list of users who can be contacted and provide users with a communication platform

(3) Online chatting, making friends, video sharing, blogs, online communities, music sharing, adding comments, etc.

(4) Providing open interfaces for application plug-in development

Simultaneously, social networks face increasingly serious security threats and have become key targets of attacks on the Internet. In order to actualize the purpose of information exchange and public communication, people have the will and capacity to put their information in public on various social network websites [5]. Without privacy protection and corresponding technical awareness, the safe use of social network websites will not be guaranteed. Coupled with the lack of relevant safety technology, laws, and regulations, safety hazards have gradually become prominent. We must attach great importance to the security of social networks [6].

Statistics and research on international relevant state departments of social networking security issues show as an application in the form of a computer network, security issues, social networks, and traditional computer network security problems have many similarities [7]. However, due to the openness of social networks, in addition to various traditional network information security threats, such as viruses, vulnerabilities, and Trojan horses, social networks also need to face many attacks against their characteristics [8]. It can be divided into three categories: user privacy and data security risks associated with leak triggered, the proliferation of spam and security risks caused by cyberattacks, Internet rumors and public opinion network security threats, and other security risks caused [9].

The security solutions of social networking sites can be implemented through appropriate technical means and management measures, and the effects of different technologies or management on security protection are minimal. The technical methods are as follows [10]. 1. Identity the authentication scheme. 2. Social network identity encryption scheme. 3. Data security management. 4. They grouped data access control. The construction of social network security management measures includes the following aspects: 1. strengthen the construction of website security; 2. strengthen users' awareness of data protection; 3. strengthen supervision of third parties; 4. strengthen legal supervision.

2.2. Deep Learning Algorithm for Security. Developments in information technology have required newer and better methods to analyze how these information systems work. Various machine learning methods study the principles of the device. Deep learning is treated as advanced technology and widely deployed in multiple embranchments, including pattern recognition, natural language processing, and network security. Due to the corresponding increment in the production of data, the ordinary machine learning methods deployed in the field of cybersecurity are gradually unable to work for intrusion detection in the network systems [11]. Therefore, the use of deep learning methods for big data analysis is an innovation that attempts to study the patterns of network connections to detect unauthorized access to computer networks.

Primarily, the system operates according to known principles, including two machine learning methods:

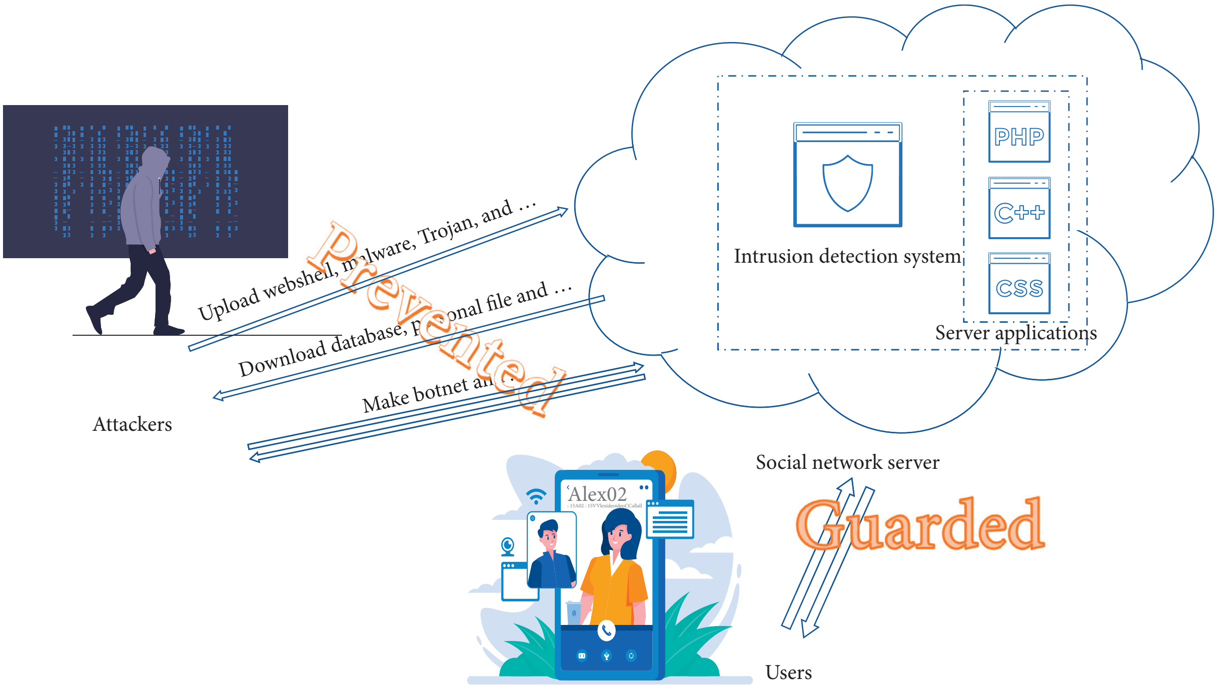

Attackers
Upload webshell, malware, Trojan, and …
Download database, personal file and …
Make botnet and …
Prevented
Intrusion detection system
PHP
C++
CSS
Server applications
Social network server
Alex02
Guarded
Users

Figure 1: The scenario of attackers and defenders using the IDS in the social network server.

probabilistic and deterministic [12]. The deterministic machine learning method employs a small sample dataset and analyzes them to find any regular pattern deviation. IT experts will then evaluate this information and develop models for classifying the data and processing the results. Usually, the information in the model is compared with the baseline. Therefore, any abnormal data beyond the average level is deemed to be invasive [13].

Besides, the probabilistic method of machine learning takes another big progress because it can evaluate the patterns involved in the evaluation, and these patterns may not be able to get rid of deterministic analysis [14]. The whole system depends on the cluster detecting any strange characters related to the data. The system depends on the unsupervised operation, where the whole system operates independently, generating the map, and ultimately analyzes any abnormal behavior by the same machine, so this method is more effective because the evaluation is conclusive. To be precise, 90% of the problems can be solved by conservative estimate [15].

Deep learning has also recently gained attention because of its advantages. This method is dynamic because the system is predictable and can adapt to generated data. It is worth noting that this method uses the output of the top-down method and uses it as the input of the bottom-up method [16]. The model also uses linear models to extract features, and these linear models are used as essential functions of layers. These layers depend on each other to form a more in-depth system architecture [17].

2.3. Security Application for the Intrusion Detection System. Under the constant development of web technology, the web is used more and more widely. The frequency with which malicious hackers attack a website is usually proportionate to the websites' value. Even if the website's value is comparatively insignificant, it will face malicious test attacks by "script kiddies" or tests on various large-scale vulnerability scanners, just like a saying in the security industry: "there are only two kinds of people in the world, one knows that they have been hacked, and the other I do not know if I was hacked" [18].

At this time, the log analysis of the website is critical. As the role of the management manager, operation staff, and maintenance technician of the website, if they are not on to the real-time security status of the server, they will become the type of "unknown whether they were hacked" and result in losses. Of course, there is another situation where hacking caused economic losses [19]. At this time, we will also conduct log analysis and other emergency measures to try to recover the loss. In short, two of the most direct and most apparent purposes are to implement security log analysis. Self-security events occur on the server to understand, followed by an emergency analysis and evidence collection [20].

The development of deep learning technology has promoted the progress of intrusion detection research. It can use the hierarchical structure to achieve unsupervised functional learning and data pattern classification. The feature extractors and classifiers can be integrated into a framework without the need for security experts to extract features [21] manually. Deep learning methods can efficaciously deal with traffic data of the large-scale network. Compared with shallow traditional machine learning methods, it has higher efficiency and detection rate, but its training process is more complex, and the model's interpretability is poor.

At present, the IDS based on deep learning solves the following problems: (1) the abnormal traffic data in the network is far less than the category imbalance problem of normal traffic data [22]; (2) the network data volume is large, the feature dimension increases, and the shallow machine learning technology is challenging to match. Massive high-dimensional data is detected, and it is challenging to extract nonlinear features from the data [23]; (3) improve the algorithm itself.

For the above problems, the current work is summarized as follows: (1) if an imbalanced dataset is used to build an intrusion detection model, it will seriously impact detection accuracy. Two methods are usually used to solve the imbalance problem in the data. They are a solution at the algorithm level (cost function method) and a solution at the data level (undersampling and oversampling methods) [24]. Some studies have used these two methods to deal with the imbalance of data categories in the IDS, but most research works have not considered the data imbalance. With the emergence of GANs with strong generation capabilities in recent years, new ideas have been provided to solve data category imbalance. GAN can solve this problem by generating new data. Therefore, the problem of data category imbalance is still currently studied as one of the hot issues [25]. (2) Most deep learning algorithms such as AE, DBM, DBN, LSTM, and CNN [26] have been used to solve this problem. In the deep learning comparison experiments cited in the article, it can be found that the detection accuracy of deep learning methods is usually excellent Traditional machine learning techniques, such as naive Bayes, decision tree, random forest, and support vector machine [27], reflect the effectiveness of deep learning detection methods [28]. However, some evaluation indicators such as the detection accuracy of binary classification and multiclassification problems based on deep learning IDS research still need to be improved [29]. For example, based on the accuracy of the multiclassification problem on the KDDTest + test set in the NSL-KDD dataset, the accuracy is roughly in the range of 79%–85%, while the accuracy of the multiclass problem on the KDDTest-21 test set is roughly 60%–69% [30]. Within the scope, there is still room for improvement. (3) Some studies have improved the deep learning algorithm to improve the model detection ability, such as using the SVM to replace the softmax function to improve the detection ability of the GRU model [31] and using a genetic algorithm to optimize the network structure of DBN [32].

3. System Description and Methodology

3.1. System Architecture. Each log line is generated by the output statement of the source code. For example, the log print statement of a process is printf (“accepted password for %s from %s port %d ssh,” user, host, and port), and then, during the running of the program, it may generate a log sentence: “17 Nov 2020 19 : 12 : 48 combo sshd (pam) [1241]: accepted password for root from 10.21.234.33 port 8888 ssh.” The logs printed by a source code are of the same type, and their codes are called log keys.

Adjacent or similar logs have a high degree of relevance. A log often depends on the previous logs. When a log breaks this logic, it means that an abnormal log execution path has occurred. Thus, we can regard the log execution sequence as a multiclassification problem [33]. The total number of log keys is certain. We regard it as K. In the training phase, we input the typical log execution sequence to generate a multiclassifier model. In the testing phase, we input the history of the most recent log key, and the output is the probability distribution of a log key. When the error between the sequence prediction result and the actual result is large, we can consider the log to be abnormal.

Figure 2 describes the procedure of the IDS. The input is a sequence of log data, and the output is the probability value. The log parser will parse the log data into the appropriate form and feed it to the neural network.

3.2. Principle of Log Parser. Logs can be divided into log keys and log parameters. We must first separate the two and parse the logs into the structure. The complete process of parsing logs is as follows (Algorithm 1).

3.3. Handling Feature Extraction. After the log parsing is completed, we have obtained the structured log of the system, but, at this time, the log key is only in the form of a string, and the parameter list elements are also strings, which cannot be directly used as the input of the deep learning model, so we need to use it characterized as a feature vector in the digital form. The feature extraction process converts the string into quantifiable numbers, thereby constructing a feature vector matrix. We use two different characterization methods for log keys and parameters due to their different formation methods and expressive meanings. Figure 3 shows the flow chart of the feature extraction.

3.3.1. Log Key Encoding. Since the log is output by the program’s code or process, the code is constant, so the type of the output log is also constant, and the number is often not very large. Therefore, for the log key, it is directly coded using sequential numbers. For example, for log keys $K1$, $K2$, and $K3$, we directly characterize them as 1, 2, and 3.

3.3.2. Log Parameter Encoding. Unlike the log type, the parameter value is not generated by the template but dynamically generated according to the actual situation during the system operation, so it often has great uncertainty. The string type of the parameter value will be many and various. In many cases, the direct use of simple integer permutation codes will cause the linear length to be too large.

My approach is to extract all parameter lists, perform parameter preprocessing, and remove all punctuation marks and special characters. Because these characters are not used as criteria for parameter abnormalities, they may affect characters’ accuracy. Then, all the parameter strings are recomposed to form a token list, the tokenizer module under the deep learning library Keras is used to process the strings, and the fit_on_texts method is used to learn the text dictionary, which is the mapping relationship between the corresponding words and numbers. Statistics such as word

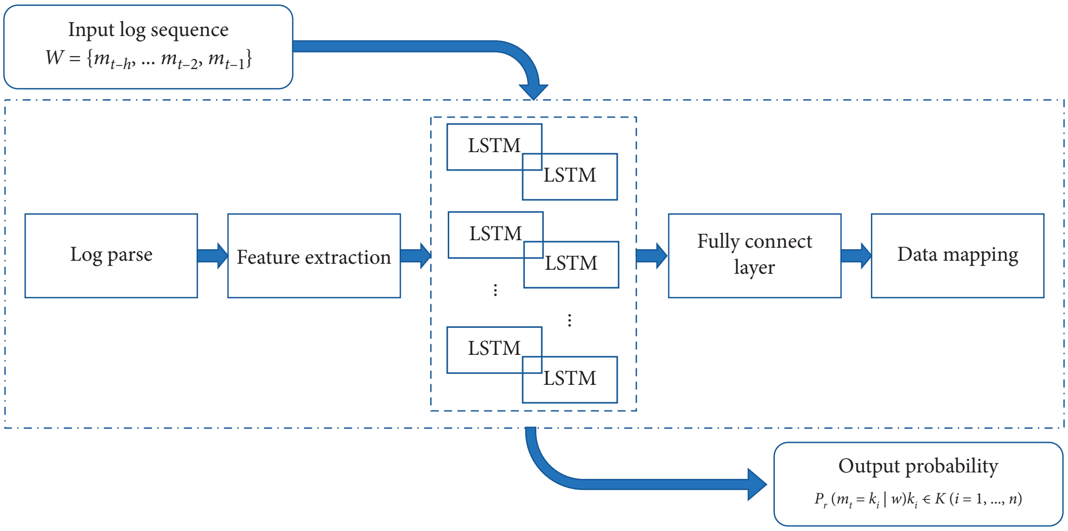


Figure 2: The flowchart of the IDS using LSTM.

```
      Input: log input
      Output: sequence output
 (1)  Initialization (log object, log template, line number list)
 (2)  Store all log objects to map
 (3)  Read log by STREAM
 (4)  Traverse the map to find the largest common subsequence
 (5)  if matching object is found then
 (6)      GOTO Line 11
 (7)  else
 (8)      GOTO Line 10
 (9)  end if
(10)  Initialize the line of log into the list map
(11)  Update the line log
(12)  Update the template
(13)  GOTO Line 3
```

Algorithm 1: Log parser.

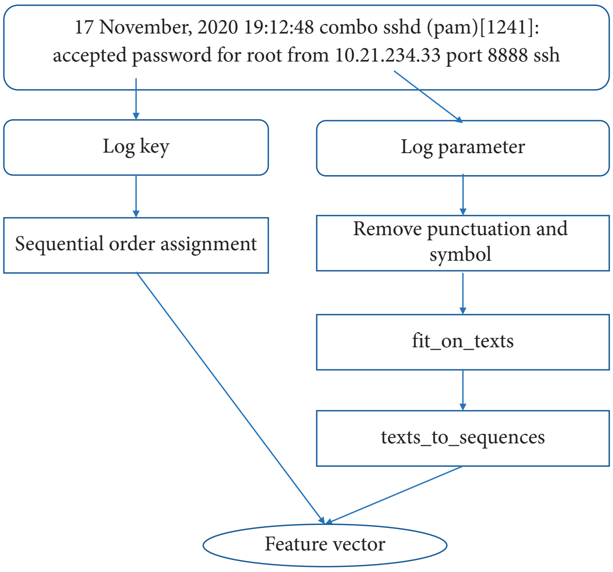

17 November, 2020 19:12:48 combo sshd (pam)[1241]:
accepted password for root from 10.21.234.33 port 8888 ssh
Log key
Log parameter
Sequential order assignment
Remove punctuation and symbol
fit_on_texts
texts_to_sequences
Feature vector

Figure 3: The flow chart illustrates the procedure of feature extraction.

frequency of parameter values. Then use the texts_to_sequences function of the text.Tokenizer module to convert the parameter text to a number, and use 0 to pad sequences of different lengths to the same length.

3.4. Anomaly Detection Scheme. Given a small number of regular log key sequences, and then, input the LSTM model for training. A detection model is obtained. When the system is running normally, the log sequence generated by the system is collected, a log key sequence is obtained after log analysis and feature extraction, and then the LSTM model is used to predict the possibility of a log key after the sequence. The next log in the situation has a greater probability of probability distribution; then, this log is considered a regular log. Otherwise, it is judged to be abnormal.

LSTM is a special type of RNN to avoid long-term dependence through deliberate design. Storing information for a long time is the default behavior of LSTM. All recurrent neural networks have chain repeating modules of neural networks. Different from the single neural network layer of RNN, there are four network layers in LSTM, and they interact in a very special way [34]. The LSTM core unit function process is shown in Figure 4, and the whole steps of the process are described as follows (Algorithm 2).

The first step of LSTM is to select what information to abandon from the cell state. This decision is made by the S-shaped network layer called the "forget gate layer." It receives h_{t-1} and x_t, and the output value is between 0 and 1 for each number in the cell state C_{t-1}. 1 means "accept this completely," and 0 means "ignore this completely:"

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right). \tag{1}$$

The next step is to determine which information needs to be stored in the cell state. This is divided into two parts. In the first part, an S-shaped network layer called the "input gate layer" determines which information needs to be updated. In the second part, a tanh network layer creates a new candidate value vector-$\widetilde{C}_t$, which can be used to add to the cell state. In the next step, we combine the above two parts to generate an update to the state:

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right),\\ \widetilde{C}_t &= \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right). \end{aligned} \tag{2}$$

Now, update the old cell state C_{t-1} to C_t. The previous steps have already decided what to do, and we just need to do it. We multiply the old state by f_t to forget what we decided to forget. Then, we add $i_t * \widetilde{C}_t$, which is the new candidate value, which is scaled proportionally according to the updated value we decide for each state:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t. \tag{3}$$

Finally, we need to calculate the output value. The output depends on our cell state, but will be a "filtered" version. Firstly, we run the S-shaped network layer to determine which parts of the cell state can be output. Then, we input the cell state into tanh (adjust the value between −1 and 1) and multiply it with the output value of the S-shaped network layer so that we can output the points we want to output:

$$\begin{aligned} o_t &= \sigma\left(W_o [h_{t-1}, x_t] + b_o\right),\\ h_t &= o_t * \tanh\left(C_t\right). \end{aligned} \tag{4}$$

In the training phase, the model needs to find an appropriate weight distribution so that the ultimate output data of the LSTM sequence generates the required labels and outputs it along with the input in the training dataset. In the training procedure phase, each input and output use the gradient descent method to find the minimum loss to update these parameters' weights. The input is a log sequence, and the output is the log key value immediately following this log sequence. In training, the loss function used by the log key is categorical cross-entropy loss, and the parameter uses the mean square loss to measure the error.

In the detection stage, we use a layer containing x LSTM blocks to predict the output of an input sequence w and add each log key of w to the LSTM block corresponding to this layer. We use multiple hidden layers, and the previous hidden state will be used as the input in the next layer's core blocks, turning the model into a deep neural network, and the input layer will come from all log key types K. The n logs are encoded as one-hot vectors. The output layer uses a standard polynomial logistic function (standard multinomial logistic function) to convert the output into a probability distribution function. When the model predicted log key and the actual log key have an enormous difference, which exceeds the threshold we defined, it is determined that the log has an abnormal execution path.

The model architecture of anomaly detection based on the LSTM is shown in Figure 5. The figure's top shows an LSTM module, and the repeated LSTM blocks make up the entire architecture. Each LSTM module will record a state as a fixed-dimensional vector. The LSTM module's state from the previous time step and its external input S_{t-i} will be used as the next LSTM module's input to calculate the new state and output. This method ensures that the log information in the log sequence can be passed to the next in an LSTM block [35].

Figure 4 shows a sequence of the core blocks that form an expanded form of the cyclic pattern. Each block retains a hidden state $Ht - i$ and a state vector C_{t-i}, both of which are transferred to the next block, thus initializing its state. For each log key data in the input sequence w, we use an LSTM core block. Therefore, a layer is composed of h-expanded LSTM core blocks. In an LSTM core block, the input m_{t-i} and the output H_{t-i-1} of the previous block can be determined:

The degree to which the previous unit state C_{t-i-1} is retained in the state C_{t-i}

How the current layer input and previous layer output affects the state

The establishment of the output H_{t-i}

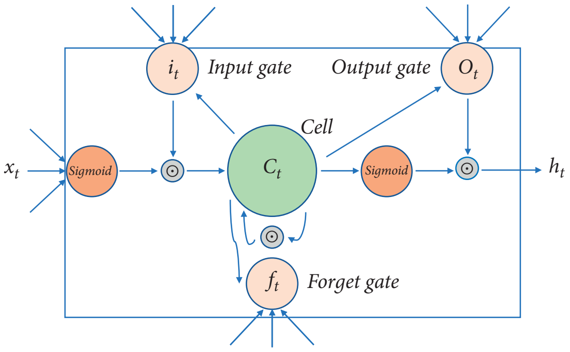


Figure 4: The function process of the LSTM core unit.

```
Input: input sequence
Output: prediction
while BatchNotFinished do
    InitializeParameters(normal_log, sequence_window, num_layers, hidden_size)
    Connect the previous hidden state with the current input − >combine
    Put the combineinto forget layer, DELETE irrelevant data
    Create a candidate layer using combine < −cell state
    combine − >input layer, decide candidate layer data
    Calculate the vector using forget, candidate and input layers
    Calculate the current output
    Update the new hidden state
end while
Output the prediction
```

Algorithm 2: Anomaly detection.

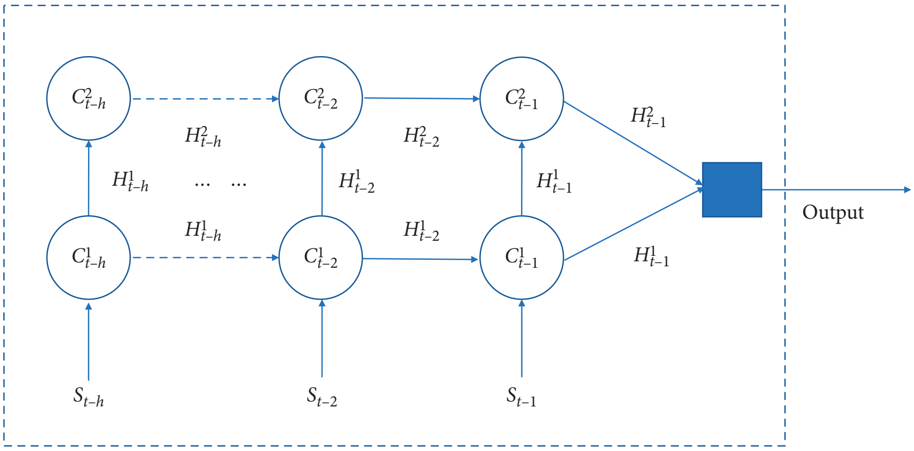


Figure 5: The detection model using LSTM.

LSTM is determined to use a combination of gate functions. These functions resolve the state by influencing the state reserved by the previous LSTM block, the previous block's output information, and the current block's input information flow. A set will learn the parameters of each gate of weights.

4. Result and Analysis

To solve the security problem of social network servers, we propose an IDS based on log analysis. The proposed method in this paper is composed of log analysis, feature extraction, and classification detection. The feature extraction part uses the LSTM neural network so that the IDS can better extract the feature information hidden in the log and achieve better detection results. This section will analyze the dataset, evaluation method, and experimental results.

The HDFS audit log (audit log) reflects the user operations on HDFS. The detailed information includes the operation's success or failure, user name, client address, operation command, and operation directory. For each user's operation, the NameNode will organize the information in the form of key-value pairs into a log in a fixed format and then record it in the audit.log file. Through the audit log, we can view various operating conditions of HDFS in real time, track various misoperations, and do some indicator monitoring.

The dataset uses the public HDFS log dataset. The HDFS data contains 11,175,629 log messages, which are Hadoop logs collected from the Amazon EC2 platform, and are marked as normal or abnormal by experts in the relevant field [36]. The parsed dataset is shown in Table 1.

4.1. Numerical Standardization. First, calculate the average value and average absolute error of each attribute:

$$\overline{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{ik}, \qquad S_A = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ik} - \overline{x}_k\right)^2}. \tag{5}$$

Thereinto, x_k stands for the average value of the attribute k, S_k stands for the average absolute error of the attribute k, and x_{ik} stands for the attribute k of the record i. Then, standardize each data record:

$$Z_{it} = \frac{x_{it} - \overline{x}_i}{S_t}. \tag{6}$$

Thereinto, Z_{ik} stands for the attribute k of the record i after standardization.

4.2. Numerical Normalization. Normalize each value after standardization to the section $[0, 1]$:

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{7}$$

Thereinto, $x_{\max}$ stands for the maximum value of the sample data, and $x_{\min}$ stands for the minimum value [37].

4.3. TF-IDF Term Weighting. In the large text corpus, some words appear very many, and they carry a small amount of information. We cannot directly use these words' frequency in the classifier, which will reduce the terms that we are interested in, but the frequency is minimal. We need to further re-weight the feature frequency of the feature into a floating-point number to facilitate the classifier's use. This step is completed by TF-IDF conversion [38].

If a word is more common, the denominator is larger, and the inverse document frequency is smaller and closer to 0. The reason for adding 1 to the denominator is to avoid the denominator being 0 (that is, all documents do not contain the word). The operator log means taking the logarithm of the obtained value:

$$\mathrm{tf}_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}}, \qquad \mathrm{idf}_i = \log\frac{|D|}{\left|\left\{j\colon t_i \in d_j\right\}\right|}. \tag{8}$$

A high word frequency in a particular document and a low word frequency in the entire document collection can produce a high-weight TF-IDF. Therefore, TF-IDF tends to sift through universal words and keep meaningful words. That is, TF-IDF is actually $\mathrm{tf} * \mathrm{idf}$.

4.4. Word2Vec. The training process of word2vec is to train an external neural network to map each word in the training set to a vector space of a specified dimension [39].

The basic unit of word2vec vectorization is words. Each word is mapped to a vector of a specified dimension, and all words form a word sequence (sentence) to become a vector matrix (the number of words x in the specified word2vec embedding dimension). However, the input required by the machine learning algorithm is a one-dimensional tensor. Therefore, we also need to perform feature processing using the word vector table to perform feature encoding on the original corpus via the TF-IDF method.

Precision and Recall rate are used in this paper as indicators:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{9}$$

TP. True Positive relates to the num of correctly classified as malicious

TN. True Negative relates to the num of correctly classified as benign

FP. False Positive relates to the num of mistakenly classified as malicious

FN. False Negative relates to the num of mistakenly classified as benign

$$F1 - \mathrm{Score} = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{10}$$

The F1 − score is defined as the weighted harmonic mean of the test's precision and recall [40]. The calculation of the F1 − score takes into account the precision and recall of the test. The precision, also known as a positive predictive value, is the proportion of positive results that really indicate a positive. The recall rate (also called sensitivity) is the ability to test to identify a positive result to correctly obtain an accurate positive rate. The F1 − score can reach the best value, that is, the precision and recall rate are 1. The worst

Table 1: Parsed log structure.

Type	Data
Date	17 November, 2020, 19 : 12 : 48
Module	Combo
Process	sshd (pam)[1241]
Log content	Accepted password for root from 10.21.234.33 port 8888 ssh
Log key	Accepted password for * from * port * ssh
Parameter list	(“root,” “10.21.234.33,” “8888”)

Table 2: Hardware and software environment.

No.	Type	Description
1	OS and version	Ubuntu 20.04
2	Experiment platform	Tensorflow 2.0
3	CPU	Intel Core-i7 9700 K
4	Units of CPU	8 cores 8 threads
5	Memory	DDR4 32 GB
6	Graphics card	GTX1070 8 GB
7	Storage	2 TB Nvme SSD

F1 – score means that the lowest precision and lowest recall rate should be 0.

The test environment is shown in Table 2, and the parameters' description is shown in Table 3.

The parameter normal_log and sequence_window represent the sensitivity of the IDS, and the adjustment of these two parameters is the core mission of the experiment. The parameter num_layers and hidden_size reflect the performance of the IDS, and the adjustment should both consider the effect and overhead. The following parameter values normal_log = 10, sequence_window = 10, num_layers = 2, and hidden_size = 64 are used by default to train the deep learning model. normal_log represents the range of normal conditions in the prediction output (the threshold of the normal logs), sequence_window is the sequence window size, and num_layers and hidden_size represent the number of hidden layers and the size of hidden layers in the LSTM model. Each of the four parameters will be adjusted in these experiments; with the target parameter's adjustment, the other three parameters will remain in the default value. To evaluate the model, the F1 – score and the all-around performance should be concerned. Increasing the value of sequence_window, num_layers, or hidden_size will increase the model's complexity, increasing the training cycle and system overhead. Variable adjustments are taken as the parameters, and the results are as follows.

In Figure 6, the compared parameter is the normal_log, which refers to the several top numbers of the prediction output sequence. The range of this parameter chooses from 6 to 12 because the performance will be deficient when the value is under 6, and the overhead will be high when the value is above 12. It can be concluded from the figure when the parameter is 11, the F1 – score reaches the highest point and slightly decreases when the parameter is over 11. In Figure 7, the compared parameter is the size of the sequence_window, which refers to the length of the input sequence, The range of this parameter chooses from 6 to 13 for the reason of the efficiency and cost. In the experiment, the F1 – score is both at a high point when the parameter is 9 and 13. However, the precision and recall are closer when the parameter is 13. Therefore, the sequence_window value should be 13.

In Figure 8, the compared parameter is the amount of the hidden layers (num_layers), which relates to the increase of the neural network's resolution and complexity. The range of this parameter chooses from 1 to 5 concerning the balancing of the profit and cost. It can be concluded from the figure that the F1 – score and the performance of the model become more balanced when the parameter is 2. In Figure 9, the compared parameter is the size of each hidden layer (hidden_size), which represents the amount of the node units in the hidden layer. The value of this parameter is according to the convergence, scale of the input and output layer, and the training samples. Here, we tested from 16 to 512 nodes of each hidden layer, and the figure indicates that the F1 – score has already reached a higher point when the value of this parameter is 64. And, when the parameter increases from 64, the indicator does not vary much, but the overhead of the system increases obviously. Thus, it comes to the conclusion that the parameter should be 64 in this experiment.

In conclusion, the model achieves the best performance when normal_log = 11, sequence_window = 13, num_layers = 2, and hidden_size = 64.The further comparative study showed that the proposed method has better performance than other deep learning algorithms and traditional machine learning algorithms. This paper selects MLP [41], RBM [42], SVM [43], and naive Bayes algorithms [44] for comparative analysis, uses comprehensive F1-score indicators for comparison, and uses training samples of different magnitudes. As shown in Figure 10, the algorithm proposed in this paper achieves a more accurate recognition rate than traditional machine learning algorithms and has better results than ordinary deep learning algorithms.

Table 3: Parameters’ description.

No.	Parameter	Description	Adjustment range
1	normal_log	The threshold of the normal logs	6–13
2	sequence_window	The detection window size	6–13
3	num_layers	The number of hidden layers	1–6
4	hidden_size	The number of memory units	32–512

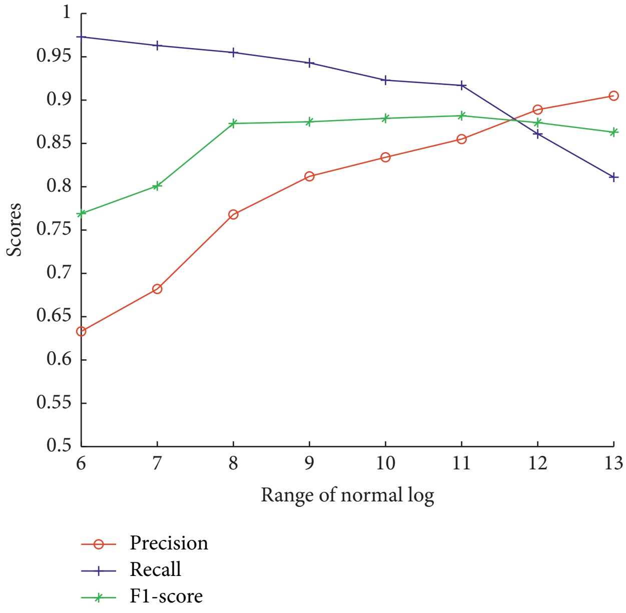


Figure 6: Scores with normal_log variation.

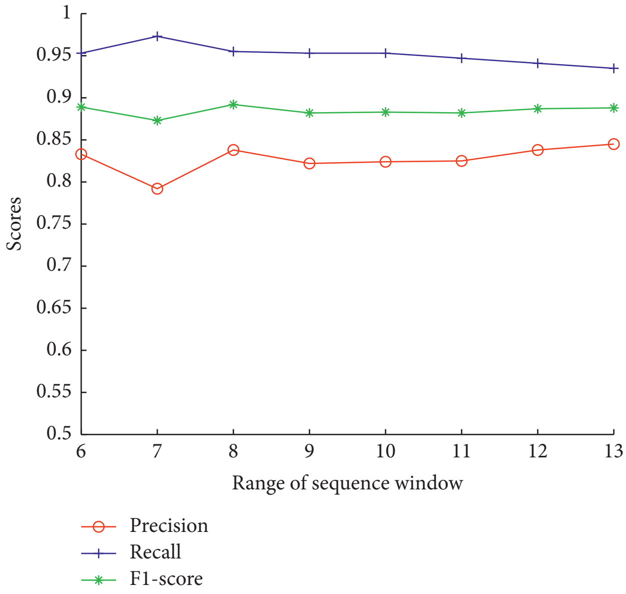


Figure 7: Scores with sequence_window variation.

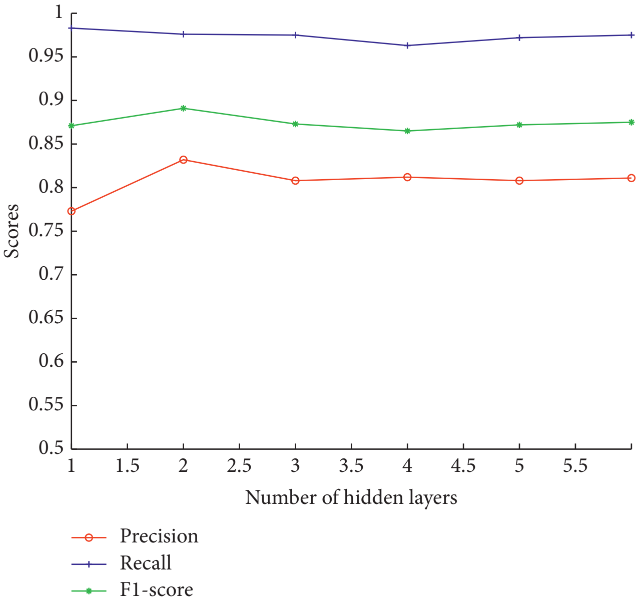


Figure 8: Scores with num_layers variation.

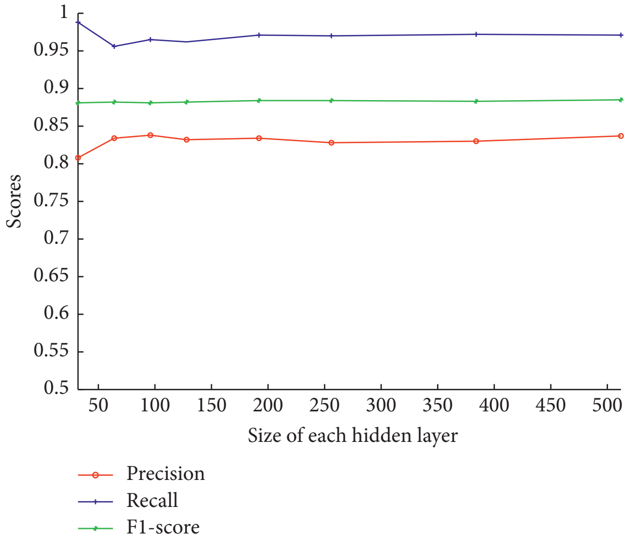


Figure 9: Scores with hidden_size variation.

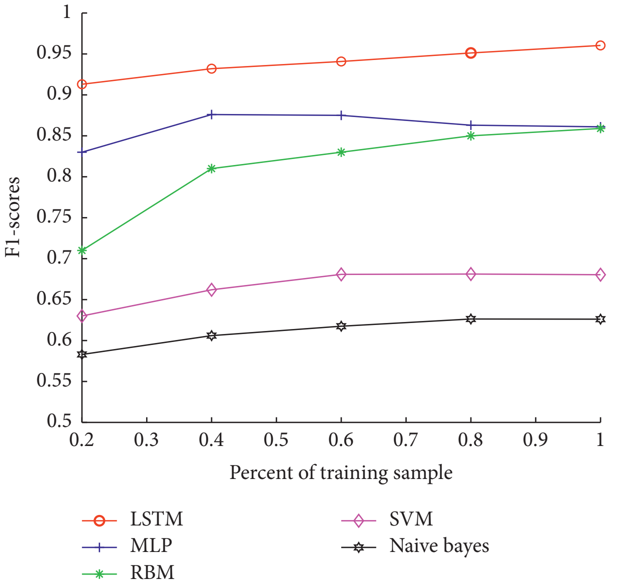


Figure 10: The comparison with other algorithms.

5. Conclusion and Future Works

Social networking sites are often faced with server security risks. The exposure of social network servers to the ubiquitous Internet environment leaves massive attacks everywhere. Applying deep learning-based log analysis methods, IDSs can detect potential cyber-attacks and prevent the social network server from being unsecured. This paper first analyzes the features of social network servers and summarizes the security issues existing in social network servers. Afterwards, the solution to the security issue was discussed, and the social network website server security protection was selected as the entry point, and the existing protection solutions were summarized. After comparing and analyzing the advantages and disadvantages, a log detection scheme based on deep learning is proposed to protect social network servers.

In this paper, a security log analysis method based on the IDS is proposed by using deep learning algorithms. The proposed IDS has three parts: log parser, feature extraction, and anomaly detection. The log analysis method combines deep learning algorithms with traditional black and white lists, rule matching, and statistical strategies to complement each other's advantages. Traditional analysis methods have good identification capabilities for known and common attack behaviors, while deep learning algorithms have an adaptive ability to detect unknown and new attack behaviors and distinguish different types of attacks. The performance of deep learning is better, and the overhead is lower. In the experiments, the proposed method is proved to have a significant improvement on the detection capabilities, which has a preferable $f1-$ score than other comparative methods. Nevertheless, this paper mainly focuses on classical datasets and algorithms but has not made further comparisons with new datasets and algorithms. More datasets and algorithms will be employed in the future work to conduct experiments, and the proposed scheme will be optimized.

Data Availability

The HDFS data used to support the findings of this study have been deposited in the Wei Xu repository (https://github.com/logpai/loghub/tree/master/HDFS).

Conflicts of Interest

There are no conflicts of interest to declare.

References

[1] R. A. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview," *Computer*, vol. 35, pp. supl27–supl30, 2002.

[2] W. Pieters, Z. Lukszo, D. Hadziosmanovic, and J. Van Den Berg, "Reconciling malicious and accidental risk in cyber security," *Journal of Internet Services and Information Security*, vol. 4, pp. 4–26, 2014.

[3] G. Vigna, W. Robertson, V. Kher, and R. A. Kemmerer, "A stateful intrusion detection system for world-wide web servers," in *Proceedings of the 19th Annual Computer Security Applications Conference*, pp. 34–43, Las Vegas, ND, USA, December 2003.

[4] D. Fincher, A. Sorkin, T. Reznor, S. Rudin, and B. Mezrich, "The social network," Sony Pictures Home Entertainment USA, Culver City, CL, USA, 2010.

[5] H. Peng, Z. Qian, Z. Kan, H. Ye, Z. Fang, and D. Zhao, "Security assessment for cascading failures of cyber-physical systems under target attack strategy," in *Proceedings of the International Conference on Frontiers in Cyber Security*, pp. 315–327, Tianjin, China, November 2020.

[6] S. Rathore, P. K. Sharma, V. Loia, Y.-S. Jeong, and J. H. Park, "Social network security: issues, challenges, threats, and solutions," *Information Sciences*, vol. 421, pp. 43–69, 2017.

[7] H. Peng, W. Peng, D. Zhao, and W. Wang, "Impact of the heterogeneity of adoption thresholds on behavior spreading in complex networks," *Applied Mathematics and Computation*, vol. 386, Article ID 125504, 2020.

[8] F. Amato, A. Castiglione, A. De Santo et al., "Recognizing human behaviours in online social networks," *Computers & Security*, vol. 74, pp. 355–370, 2018.

[9] Y. Hori, W. Claycomb, and K. Yim, "Guest editorial: frontiers in insider threats and data leakage prevention," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 3, 2012.

[10] J. Nagy and P. Pecho, "Social networks security," in *Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies*, pp. 321–325, Athens, Greece, June 2009.

[11] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, 2019.

[12] F. Ertam, "An efficient hybrid deep learning approach for internet security," *Physica A: Statistical Mechanics and Its Applications*, vol. 535, Article ID 122492, 2019.

[13] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys & Tutorials*, vol. 21, 2020.

[14] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in *Proceedings of the 2018 10th*

International Conference on Cyber Conflict (CyCon), pp. 371–390, Tallinn, Estonia, May 2018.

[15] W. Guo, D. Mu, J. Xu, P. Su, G. Wang, and X. L. Xing, “Explaining deep learning based security applications,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 364–379, Toronto, Canada, October 2018.

[16] X. Ling, S. Ji, J. Zou et al., “Deepsec: a uniform platform for security analysis of deep learning model,” in *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, pp. 673–690, San Francisco, CA, USA, May 2019.

[17] M. Alazab and M. Tang, *Deep Learning Applications for Cyber Security*, Springer, Berlin, Germany, 2019.

[18] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, “Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study,” *Journal of Information Security and Applications*, vol. 50, Article ID 102419, 2020.

[19] A. Thakkar and R. Lohiya, “A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges,” *Archives of Computational Methods in Engineering*, vol. 33, pp. 1–33, 2020.

[20] H. Peng, Z. Kan, D. Zhao, and J. Han, “Security assessment for interdependent heterogeneous cyber physical systems,” *Mobile Networks and Applications*, vol. 42, pp. 1–11, 2019.

[21] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, “A deep learning approach for network intrusion detection system,” in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pp. 21–26, New York , NY, USA, May 2016.

[22] H. J. Liao, C. H. R. Lin, Y. C. Lin, and K. Y. Tung, “Intrusion detection system: a comprehensive review,” *Journal of Network and Computer Applications*, vol. 36, pp. 16–24, 2013.

[23] T. Booth and K. Andersson, “Network security of internet services: eliminate DDoS reflection amplification attacks,” *Journal of Internet Services and Information Security (JISIS)*, vol. 5, pp. 58–79, 2015.

[24] C. Atapour, I. Agrafiotis, and S. Creese, “Modeling Advanced Persistent Threats to enhance anomaly detection techniques,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 9, 2018.

[25] S. R. Chhetri, A. B. Lopez, J. Wan, and M. A. Al Faruque, “Gan-sec: generative adversarial network modeling for the security analysis of cyber-physical production systems,” in *Proceedings of the 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 770–775, Florence, Italy, March 2019.

[26] E. Aminanto and K. Kim, “Deep learning in intrusion detection system: an overview,” in *Proceedings of the 2016 International Research Conference on Engineering and Technology (2016 IRCET). Higher Education Forum*, Seoul, South Korea, January 2016.

[27] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, “Intrusion detection by machine learning: a review,” *Expert Systems with Applications*, vol. 36, pp. 11994–12000, 2009.

[28] S. S. Roy, A. Mallik, R. Gulati, M. S. Obaidat, and P. V. Krishna, “A deep learning based artificial neural network approach for intrusion detection,” in *Proceedings of the International Conference on Mathematics and Computing*, pp. 44–53, Haldia, India, January 2017.

[29] G. Meena and R. R. Choudhary, “A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA,” in *Proceedings of the 2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pp. 553–558, Manipal, India, April 2017.

[30] K. Alrawashdeh and C. Purdy, “Toward an online anomaly intrusion detection system based on deep learning,” in *Proceedings of the 2016 15th IEEE international conference on machine learning and applications (ICMLA)*, pp. 195–200, Anaheim, CL, USA, December 2016.

[31] A. F. M. Agarap, “A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data,” in *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pp. 26–30, Macau, China, March 2018.

[32] L. Stepanov, A. Parinov, L. Korotkikh, and A. Koltsov, “Approach to estimation of level of information security at enterprise based on genetic algorithm,” *Journal of Physics: Conference Series*, vol. 1015, Article ID 032141, 2018.

[33] Q. Fu, J. G. Lou, Y. Wang, and J. Li, “Execution anomaly detection in distributed systems through unstructured log analysis,” in *Proceedings of the 2009 ninth IEEE international conference on data mining*, pp. 149–158, Miami, FL,USA,, December 2009.

[34] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: continual prediction with lstm,” 1999.

[35] M. Sundermeyer, R. Schlüter, and H. Ney, “LSTM neural networks for language modeling,” in *Proceedings of the Thirteenth annual conference of the international speech communication association*, Portland, OR, USA, September 2012.

[36] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, “Detecting large-scale system problems by mining console logs,” in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pp. 117–132, Big Sky, MT, USA, October 2009.

[37] M. Zhong, Y. Zhou, and G. Chen, “Sequential model based intrusion detection system for IoT servers using deep learning methods,” *Sensors*, vol. 21, 2021.

[38] J. Ramos, “Using tf-idf to determine word relevance in document queries,” in *Proceedings of the first instructional conference on machine learning*, pp. 133–142, Washington D.C., USA, August 2003.

[39] Y. Goldberg and O. Levy, “Word2vec explained: deriving Mikolov et al.’s negative-sampling word-embedding method,” 2014, https://arxiv.org/abs/1402.3722.

[40] R. Joshi, “Accuracy, precision, recall & f1 score: interpretation of performance measures,” 2016.

[41] P. Barapatre, N. Tarapore, S. Pukale, and M. Dhore, “Training MLP neural network to reduce false alerts in IDS,” in *Proceedings of the 2008 International Conference on Computing, Communication and Networking*, pp. 1–7, Dalian, China, July 2008.

[42] M. Mayuranathan, M. Murugan, and V. Dhanakoti, “Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 23, pp. 1–11, 2019.

[43] J. Wang, X. Hong, R. R. Ren, and T. H. Li, “A real-time intrusion detection system based on PSO-SVM,” in *Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009)*, p. 319, Busan, Korea, August 2009.

[44] S. Mukherjee and N. Sharma, “Intrusion detection using naive bayes classifier with feature reduction,” *Procedia Technology*, vol. 4, pp. 119–128, 2012.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6631247, 13 pages
https://doi.org/10.1155/2021/6631247

Research Article

Towards Revealing Parallel Adversarial Attack on Politician Socialnet of Graph Structure

Yunzhe Tian,[1] **Jiqiang Liu**,[1] **Endong Tong**,[1] **Wenjia Niu**,[1] **Liang Chang**,[2] **Qi Alfred Chen**,[3] **Gang Li**,[4] **and Wei Wang**[1]

[1]*Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing, China*
[2]*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China*
[3]*University of California, Irvine, CA, USA*
[4]*Australia Centre for Cyber Security Research and Innovation, Deakin University, Geelong, Australia*

Correspondence should be addressed to Endong Tong; edtong@bjtu.edu.cn and Wenjia Niu; niuwj@bjtu.edu.cn

Received 16 October 2020; Revised 25 January 2021; Accepted 26 February 2021; Published 9 March 2021

Academic Editor: Zhe-Li Liu

Socialnet becomes an important component in real life, drawing a lot of study issues of security and safety. Recently, for the features of graph structure in socialnet, adversarial attacks on node classification are exposed, and automatic attack methods such as fast gradient attack (FGA) and NETTACK are developed for per-node attacks, which can be utilized for multinode attacks in a sequential way. However, due to the overlook of perturbation influence between different per-node attacks, the above sequential method does not guarantee a global attack success rate for all target nodes, under a fixed budget of perturbation. In this paper, we propose a parallel adversarial attack framework on node classification. We redesign new loss function and objective function for nonconstraint and constraint perturbations, respectively. Through constructing intersection and supplement mechanisms of perturbations, we then integrate node filtering-based P-FGA and P-NETTACK in a unified framework, finally realizing parallel adversarial attacks. Experiments on politician socialnet dataset Polblogs with detailed analysis are conducted to show the effectiveness of our approach.

1. Introduction

With the development of Internet and IT technology, an emerging cyber space [1], which refers to the global network of interdependent information technology infrastructures, telecommunications networks, and computer processing systems, is covering most aspects of our daily life nowadays. In such space, as a highly important and detailed representation, various emerging social networks (e.g., Facebook, Twitter, WeChat, and TikTok) are greatly pushing the new revolution of network interconnection and interdependence, as well as the social relations and information propagation [2, 3].

Social network is called socialnet in short. Due to the popularity, billions of socialnet users share their personal data and connect with friends and family through various devices and applications. Since the socialnet can be abstracted to a simple kind of graph with features of nodes and edges, many researchers have contributed their efforts to study socialnet and corresponding services based on graph- and workflow-related approaches [4–8]. One of the most frequently applied tasks on graph data is node classification, the goal of which is to predict the labels of the remaining nodes when given a single large graph and the class labels of a few nodes [9]. For example, we can utilize node classification to predict the political labels of politician such as Liberals and Conservatives, according to their socialnet interactions.

For node classification in recent years, the graph convolutional network (GCN) [10, 11], a kind of graph neural network (GNN) [12, 13] based on deep learning, has shown a great potential. Unfortunately, such GCN also opens a new door for cyber attacks. Adversarial attacks against GCN are discovered, through few edge perturbations of addition or deletion, and they are uneasy to notice [14]. Furthermore, automatic attack methods are developed to explore effective perturbations including constraint and nonconstraint perturbations. Constraint perturbation refers to the edge perturbation satisfying specific requirements such as node degree distribution of graph [15]. Nonconstraint perturbation means a free perturbation. Accordingly, fast gradient attack (FGA) [16] and NETTACK [17] are typical nonconstraint and constraint methods, respectively. The above methods enable per-node attacks, as well as multinode attacks in a sequential way. However, due to the overlook of perturbation influence between different per-node attacks, the above sequential method does not guarantee a global attack success rate for all target nodes, under a fixed budget of perturbation. Figure 1 shows the differences between sequential and parallel attacks in a motivating example. For the No. 1, 2, and 3 nodes, the attack goal is to change their class labels through changing graph structure with perturbations, including edge addition and edge deletion. We can see that, due to removing the edges from efforts of edge addition in attack of No. 1 node, No. 2 and No. 3 node attacks of sequential attack waste edges perturbations and cause No. 1 node attack to fail with a global attack success rate of 2/3, while the parallel attack considers perturbation influence and has a higher attack success rate with a lower budget.

In this work, we are the first to perform multinode attack in a parallel way by integrating two methods P-FGA and P-NETTACK in a unified attack framework. Based on nonconstraint FGA, we redesign a new loss function in P-FGA, which employs CW-loss [18] to replace CE-loss. For P-NETTACK, we utilize the maximum sum of surrogate loss as new objective function to support parallel attack. Moreover, we apply a node filtering mechanism to P-NETTACK and P-FGA, which filters out those nodes that are successfully attacked from target node set. After extracting common perturbations, we also provide a random supplement of perturbations to fill the budget.

We experiment on politician socialnet dataset Polblogs [19] of 1222 nodes and 16714 edges, showing the effectiveness of our approach. We find that our approach can achieve a high attack success rate (*ASR*) of 71.5% at the lowest perturbation budget of $1/5\, d_{\text{sum}}$ (d_{sum} is the sum of the degrees of all target nodes), that is over 15% higher than that of NETTACK or FGA, still keeping a satisfied test statistic of 0.005. The filtering mechanism can greatly improve *ASR*, with nearly 20% average increment. We summarize our contributions as follows:

(1) We give the very first attempt to propose a multinode parallel adversarial attack framework on node classification in socialnet of graph structure, based on considering perturbation influence between per-node attacks.

(2) Node filtering-based nonconstraint P-FGA and node filtering-based constraint P-NETTACK are proposed, and we integrate them into a unified multinode parallel attack framework, through constructing intersection and supplement mechanisms of perturbation.

(3) We evaluate our approach empirically on real dataset of politician socialnet Polblogs. Based on parallel attacking on the graph of 1222 nodes and 16714 edges, we reveal and verify the effectiveness of our approach compared to sequential attacks in terms of attack strength and attack stealthiness.

The rest of the paper is structured as follows: Section 2 introduces the preliminaries and problem definition. Section 3 proposes a multinode parallel attack framework. Section 4 reports our experiments and evaluations on the politician socialnet dataset Polblogs. In Section 5, we discuss the related works. Finally, Section 6 concludes the work of this paper.

2. Preliminaries and Problem Definition

2.1. Graph Structure of Socialnet. In real socialnet, one person can have an interaction with others by operations like the following: commenting, reposting, etc. Such interaction can be quantified and qualified, varying from different measurements. For simplicity, we assume that we just use an undirected unweighted edge to denote an interaction existence, constructing graph structure of socialnet (See Figure 2). Moreover, we simply assume that one node only has one classification label, and we do not focus on multiple free-label user profile or granularity-based hierarchical user profile [20] in attack scenarios of this paper. Thus, for a socialnet graph, we have a triple $G = (V, C, \mathbf{A})$ including node set V, label set C, and adjacent matrix $\mathbf{A}$, in which $V = \{v_1, v_2, \ldots, v_n\}$, $C = \{c_1, c_2, \ldots, c_n\}$ ($|V| = |C| = n$), and $\mathbf{A}$ is shown as follows:

$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}, \quad a_{ij} \in \{0, 1\}. \tag{1}$$

2.2. Graph Convolutional Network. As a kind of GNN, GCN is an extremely powerful neural network architecture for deep learning on graphs to produce useful feature representations of nodes in networks. Given a $G = (V, C, \mathbf{A})$, we can partially delete some node's label ($c_i =$ null) to obtain a new $G' = (V, C', \mathbf{A})$. The goal of node classification is to learn a function $\mathbf{Z}$, which maps each node $v \in V$ to one class ($|\{c_i = \text{null}\}|$=0).

We use a two-layer GCN to approximate the function $\mathbf{Z}$:

$$\mathbf{Z} = f_{\boldsymbol{\theta}}(\mathbf{A}, \mathbf{X}) = \text{softmax}\left(\widehat{\mathbf{A}}\sigma\left(\widehat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(1)}\right)\mathbf{W}^{(2)}\right), \tag{2}$$

where $\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-(1/2)}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-(1/2)}$, $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I_N}$ is the adjacent matrix $\mathbf{A}$ of the input graph G' with self-loops added via the identity matrix $\mathbf{I_N}$, $\widetilde{\mathbf{D}}_{\mathbf{ii}} = \sum_j \widetilde{a}_{ij}$ is the degree matrix of $\widetilde{\mathbf{A}}$, and $\mathbf{X}$ is a matrix of node feature vectors. For the graph whose nodes do not have feature attributes, $\mathbf{X}$ can be set to an identity matrix $\mathbf{I_N}$. $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ are the trainable weight matrices of the first and second layers, respectively, and $\sigma(\cdot)$ is a ReLU activation function. For the semisupervised node classification, the optimal parameters $\boldsymbol{\theta} = \left\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}\right\}$ are

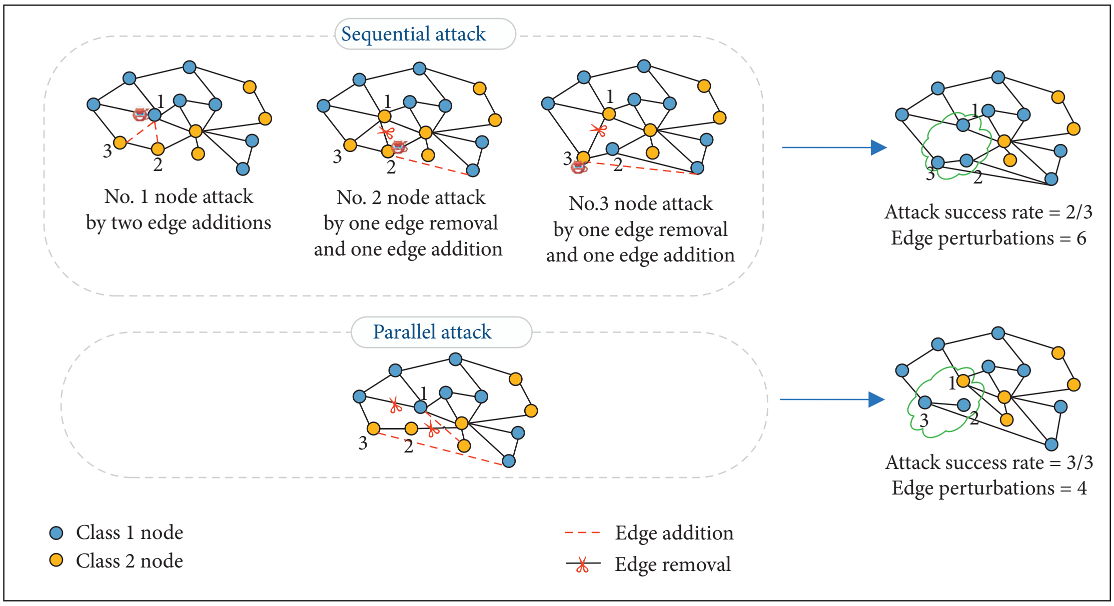

Sequential attack
1
3
2
No. 1 node attack by two edge additions
No. 2 node attack by one edge removal and one edge addition
No.3 node attack by one edge removal and one edge addition
Attack success rate = 2/3
Edge perturbations = 6
Parallel attack
Attack success rate = 3/3
Edge perturbations = 4
Class 1 node
Class 2 node
Edge addition
Edge removal

Figure 1: Differences between sequential and parallel attacks in a motivating example.

learnt by minimizing the cross-entropy loss over all labeled examples:

$$L(\boldsymbol{\theta}; \mathbf{A}, \mathbf{X}) = -\sum_{v \in V_l} \ln \mathbf{Z}_{v,c_v}, \tag{3}$$

where $V_l \subseteq V$ is the set of nodes with labels, namely, training set, c_v is the given true label of node v, and $\mathbf{Z}_{v,c_v}$ is the probability of assigning class c_v to node v.

2.3. Problem Definition. Given the attack target set $V_t \subseteq V$ in G' and perturbation budget Δ, multinode attack on GCN can be regarded as the following optimization problem:

$$\mathbf{A}^* = \arg\max_{\mathbf{A}^*} \sum_{v \in V_t} \left(\operatorname{sign}\left(\max_{c \neq c_v} \mathbf{Z}_{v,c} - \mathbf{Z}_{v,c_v} \right) \right), \tag{4}$$

s.t.

$$\sum_{u<v} \left| a_{uv} - a^*_{uv} \right| \leq \Delta, \tag{5}$$

$$u \in V_t \vee v \in V_t, \quad \text{where } a^*_{uv} \neq a_{uv}, \tag{6}$$

where $\max_{c \neq c_v} \mathbf{Z}_{v,c} - \mathbf{Z}_{v,c_v} > 0$, $\operatorname{sign}(\cdot) = 1$, else $\operatorname{sign}(\cdot) = 0$.

Formula (4) shows the objective function, aiming to find the optimal adjacent matrix. When the sum of misclassification for all target nodes is maximum, it means a most successful multinode attack. Formulas (5) and (6) show the constraints that should be satisfied. Formula (5) requires that the number of edge perturbations be no more than Δ (a predefined constant). Formula (6) has the constraint that any edge perturbation must be linked to a target attack node.

3. Multinode Parallel Attack Framework

Our multinode parallel attack framework is shown in Figure 3. Firstly, given an original graph $G' = (V, C', \mathbf{A})$ as defined in Section 2.1, we train a GCN for node classification task, and we obtain C, in which all nodes are labeled with prediction, and record G into testing result as ground truth. Then, given a target node set $V_t \subseteq V$, we utilize P-FGA and P-NETTACK to perturb the original graph, attacking target nodes in V_t. In each iteration of nonconstraint P-FGA method, based on GCN-gradient information of the adjacent matrix $\mathbf{A}$, we select the pair of nodes (v_i, v_j) of maximum absolute value of gradient to perform perturbation (edge deletion or edge addition), generating a new adversarial graph $G^{\text{adv}}_{P\text{-FGA}}$ by the generator. In each iteration of constraint P-NETTACK, to ensure keeping the perturbations unnoticeable and preserving the important structural characteristics, we firstly compute the candidate perturbation set C_{pert} to ensure the similar node degree distribution after perturbation execution. Then, according to our redesigned objective function, from candidate perturbation set, we greedily select the optimal perturbation (v_m, v_n), which obtains the highest objective score, generating a new adversarial graph $G^{\text{adv}}_{P\text{-NETTACK}}$ by the generator.

In the filtering mechanism, after each perturbation of P-FGA or P-NETTACK, the new predicted labels should be compared with the testing result to determine the attack effect. For those nodes that are successfully attacked, such mechanism filters them out from target node set V_t to form a new target node set V'_t, ignoring those nodes in the next gradient/objective function computation and perturbation selection. Such process is repeated until the perturbation budget Δ is reached. $\mathbf{D}_{P\text{-FGA}}$ and $\mathbf{D}_{P\text{-NETTACK}}$ are perturbation sets based on P-FGA and P-NETTACK, respectively. To integrate $\mathbf{D}_{P\text{-FGA}}$ and $\mathbf{D}_{P\text{-NETTACK}}$ and generate unified perturbations, we provide an intersection mechanism to extract common perturbations and a perturbation supplement mechanism to fill the perturbation budget Δ. Finally, the integrated perturbation set $\mathbf{D}_{\text{comb}}$ is used to realize effective multinode parallel adversarial attacks.

3.1. P-FGA Method. In our P-FGA, to adapt to the multinode attack, we redesign a new loss function L_{multi} for attack target set V_t, which employs CW-loss [18] to replace CE-loss and takes all target nodes into consideration (see equation (7)).

$$L_{\text{multi}} = \sum_{v \in V_t} \left\{ \max_{c \neq c_v} \ln Z_{v,c} - \ln Z_{v,c_v} \right\}. \tag{7}$$

Following the gradient-based idea of original FGA, based on the new loss function L_{multi} for multinode attack, we firstly calculate the partial derivatives with respect to the element a_{ij} of adjacency matrix $\mathbf{A}$ and further obtain gradient matrix $\mathbf{GM}$, and its element g_{ij} can be calculated by

$$g_{ij} = \frac{\partial L_{\text{multi}}}{\partial a_{ij}}. \tag{8}$$

Considering that the adjacency matrix is symmetric and its gradient matrix should also be symmetric, thus, we have

$$\begin{gathered} \hat{g}_{ij} = \hat{g}_{ji} = \begin{cases} \dfrac{g_{ij} + g_{ji}}{2}, & i \neq j, \\ 0, & i = j, \end{cases} \\ \tilde{g}_{ij} = \hat{g}_{ij} \times \left(-2 \cdot a_{ij} + 1\right), \end{gathered} \tag{9}$$

where $\tilde{g}_{ij}$ forms $\widetilde{\mathbf{GM}}$. A bigger value of multinode loss function L_{multi} corresponds to worse prediction results for the target nodes in V_t. And edge perturbations along the direction of the gradient can make the loss increase more faster locally. That is, for a positive gradient $\hat{g}_{ij}$, adding the edge between the pair of nodes (v_i, v_j) can increase the loss. Similarly, for a negative gradient $\hat{g}_{ij}$, deleting the edge also increases the loss.

However, since the adjacent matrix $\mathbf{A}$ is binary discrete and $a_{ij} \in \{0, 1\}$, not all edges can be perturbed along the direction of the gradient. For example, for a pair of nodes (v_i, v_j) who have positive/negative gradient (i.e., $(\hat{g}_{ij} > 0/\hat{g}_{ij} < 0)$) and meanwhile are connected/disconnected (i.e., $(a_{ij} = 1/a_{ij} = 0)$), we cannot further add/delete the edge along the direction of the gradient. Thus, we design equation (9); for a positive gradient $\hat{g}_{ij}$, when $a_{ij} = 0$, $\tilde{g}_{ij}$ is positive; when $a_{ij} = 1$, $\tilde{g}_{ij}$ is negative. Similarly, for a negative gradient $\hat{g}_{ij}$, when $a_{ij} = 1$, $\tilde{g}_{ij}$ is positive; when $a_{ij} = 0$, $\tilde{g}_{ij}$ is negative. Only the positive $\tilde{g}_{ij}$ enables the addition/deletion of the edge along the direction of the gradient. Then, for edge addition or deletion, we pick the optimal edge (v_m, v_n), $v_m \in V_t \vee v_n \in V_t$, with the maximum $\tilde{g}_{mn}$, and the adjacent matrix $\mathbf{A}$ is updated to $\mathbf{A}'$ by changing the corresponding value (a_{mn} and a_{nm}) to a different binary value (see equation (10)).

$$a'_{mn} = a'_{nm} = 1 - a_{mn}. \tag{10}$$

The pseudocode for P-FGA is given in Algorithm 1.

3.2. P-NETTACK Method. In constraint P-NETTACK, we use test statistic $\Lambda(G', G^{\text{adv}})$ to determine whether our generated adversarial graph $G^{\text{adv}} = (V, C', \mathbf{A}')$ and original graph $G' = (V, C', \mathbf{A})$ have similar node degree distribution of pow-law distribution $p(x) \propto x^{-\alpha}$, in which $p(x)$ denotes the probability of certain degree x, and α refers to scaling parameter. The test statistic Λ can be calculated based on the following formulas.

$$\alpha_G = 1 + \left|\mathbf{D}_{G'}\right| \cdot \left[\sum_{d_i \in \mathbf{D}_{G'}} \log \frac{d_i}{d_{\min} - (1/2)} \right]^{-1}, \tag{11}$$

$$\begin{aligned} l(\mathbf{D}_{G'}) = {} & \left|\mathbf{D}_{G'}\right| \cdot \log \alpha_{G'} + \left|\mathbf{D}_G\right| \cdot \alpha_{G'} \cdot \log d_{\min} \\ & + (\alpha_{G'} + 1) \sum_{d_i \in \mathbf{D}_{G_\prime}} \log d_i, \end{aligned} \tag{12}$$

$$\Lambda\left(G', G^{\text{adv}}\right) = -2 \cdot l\left(\mathbf{D}_{G_\prime} \bigcup \mathbf{D}_{G^{\text{adv}}}\right) + 2 \cdot \left[l(\mathbf{D}_{G'}) + l(\mathbf{D}_{G^{\text{adv}}})\right]. \tag{13}$$

In equation (11), $d_{\min}$ is the minimum degree that a node has to be considered in the power-law test, and $\mathbf{D}_{G'} = \left\{d_v^{G'} | v \in V, d_v^{G'} \geq d_{\min}\right\}$ is the multiset containing the list of node degrees, where $d_v^{G'}$ is the degree of node v in G' [21]. Equation (12) is used to evaluate the log-likelihood $l(\mathbf{D}_{G'})$ for the sample $\mathbf{D}_{G'}$ [22]. Then, we can get final test statistic by equation (13). Similar to NETTACK, we only accept adversarial graph G^{adv} whose degree distribution fulfils $\Lambda(G', G^{\text{adv}}) < 0.004$ and thus obtain the candidate perturbation set C_{pert}. In our P-NETTACK, the edge perturbations in C_{pert} must be linked to an attack target node.

To efficiently select the optimal perturbation from C_{pert}, NETTACK utilizes a linear surrogate model $\mathbf{Z}'$ to approximate the nonlinear GCN model $\mathbf{Z}$ by removing the activation function $\sigma(\cdot)$. $\mathbf{Z}'$ is calculated as follows:

$$\mathbf{Z}' = \operatorname{softmax}\left(\hat{\mathbf{A}}\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(1)}\mathbf{W}^{(2)}\right) = \operatorname{softmax}\left(\hat{\mathbf{A}}^2\mathbf{X}\mathbf{W}\right). \tag{14}$$

In our P-NETTACK, given an attack target set V_t, we utilize the sum of single surrogate losses for each $v \in V_t$ as the new surrogate loss to support multinode attack:

$$L_{\text{multi}}(\mathbf{A}; \mathbf{X}, \mathbf{W}, V_t) = \sum_{v \in V_t} \left\{ \max_{c \neq c_v} \left[\hat{\mathbf{A}}^2\mathbf{X}\mathbf{W}\right]_{v,c} - \left[\hat{\mathbf{A}}^2\mathbf{X}\mathbf{W}\right]_{v,c_v} \right\}, \tag{15}$$

where $[\hat{\mathbf{A}}^2\mathbf{X}\mathbf{W}]_{v_t,c}$ is the value of class c given to the node v_t by the surrogate model. The multinode scoring function that evaluates the multinode surrogate loss obtained after adding/deleting an edge $e = (i, j) \in C_{\text{pert}}$ is defined as

$$s_{\text{multi}}(e; \mathbf{A}, V_t) := L_{\text{multi}}(\mathbf{A}'; \mathbf{X}, \mathbf{W}, V_t), \tag{16}$$

where $\mathbf{A}$ is updated to $\mathbf{A}'$ by $a'_{ij} = a'_{ji} = 1 - a_{ij}$. Following the greedy approximate scheme in NETTACK, during each iteration, we select the optimal perturbation that has the highest value of multinode scoring function from the candidate perturbation set C_{pert} to execute. The above processes including candidate perturbation computation, determining

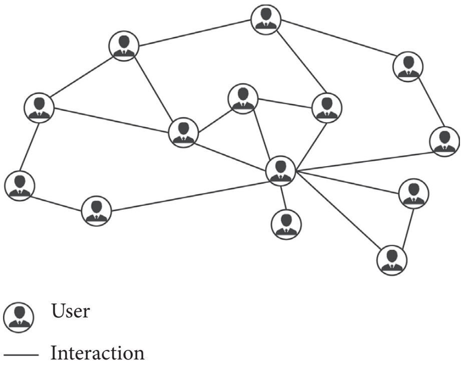


Figure 2: Illustration of socialnet graph.

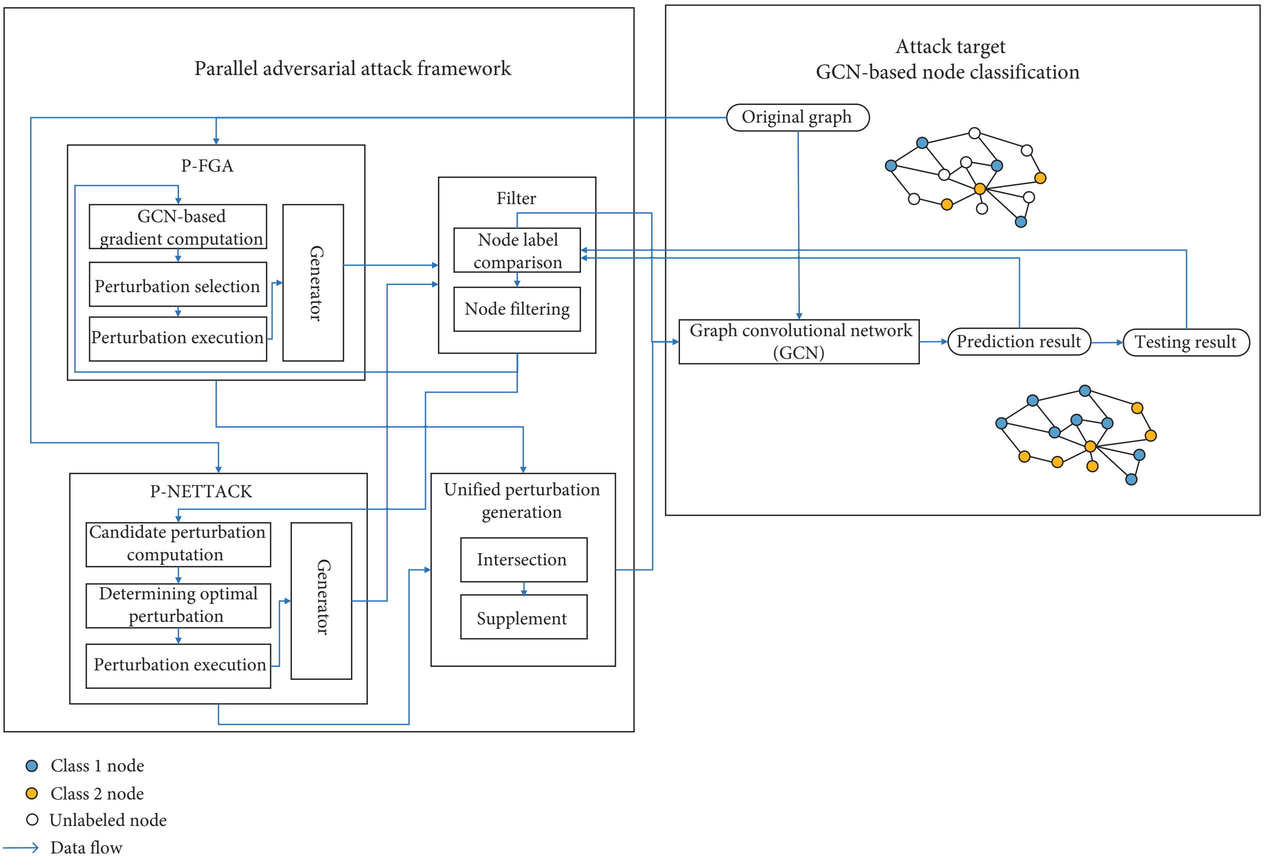


Figure 3: The parallel adversarial attack framework against GCN-based node classification.

optimal perturbation, and perturbation execution are repeated until the perturbation budget Δ is reached. The pseudocode for P-NETTACK is given in Algorithm 2.

3.3. Filtering Mechanism.

3.3. Filtering Mechanism. In this part, we propose a filtering mechanism that filters out target nodes that are successfully attacked from the target node set. After each perturbation, by the filtering mechanism, we obtain a filtered attack target set V_t', which is used in the next iteration. If there are no nodes in V_t', which means that all target nodes have been attacked successfully, and we reset the attack target set V_t' to the original attack target set V_t. The pseudocode for filtering mechanism is given in Algorithm 3.

3.4. Intersection and Supplement Mechanism. In this section, we construct intersection and supplement mechanism of perturbations. Given the perturbation sets $\mathbf{D}_{P\text{-FGA}}$ and $\mathbf{D}_{P\text{-NETTACK}}$ under a fixed perturbation budget Δ, we first utilize intersection mechanism to extract their common perturbations $\mathbf{D}_{\text{comb}}$. In general, the number of common perturbations is less than perturbation budget Δ. Thus, we should provide a perturbation supplement mechanism to fill the budget.

```
Input: $G' = (V, C', \mathbf{A})$, attack target set $V_t$, perturbation budget $\Delta$
Output: perturbation set $D_{P-\text{FGA}}$
Train the GCN model $\mathbf{Z}$ on original graph $G'$
Initialize $\mathbf{A}^{(0)} = \mathbf{A}$
Initialize perturbation set $D_{P-\text{FGA}}$
for $h = 1$ to $\Delta$ do
    //GCN-based Gradient Computation
    Calculate multi-node target loss function $L_{\text{multi}} = \sum_{v\in V_t}\left\{\max_{c\neq c_v} \ln \mathbf{Z}_{v,c} - \ln \mathbf{Z}_{v,c_v}\right\}$
    Construct $\widetilde{\mathbf{GM}}^{(h-1)}$ based on the $\mathbf{A}^{(h-1)}$:
    $g_{ij}^{(h-1)} = (\partial L_{\text{multi}}/\partial a_{ij}^{(h-1)})$, $\widehat{g}_{ij}^{(h-1)} = \widehat{g}_{ji}^{(h-1)} = \begin{cases} g_{ij}^{(h-1)} + g_{ji}^{(h-1)}/2 & i \neq j \\ 0 & i = j \end{cases}$,
    $\widetilde{g}_{ij}^{(h-1)} = \widehat{g}_{ij}^{(h-1)} \times (-2 \cdot a_{ij}^{(h-1)} + 1)$
    //Perturbation Selection
    Select $e^* = (v_m, v_n)$ where $v_m \in V_t \vee v_n \in V_t$, having the maximum $\widetilde{g}_{mn}^{(h-1)}$
    //Perturbation Execution
    Obtain the adjacency matrix $A^{(h)}$ by $a_{mn}^{(h)} = a_{nm}^{(h)} = 1 - a_{mn}^{(h-1)}$
    Generate a new adversarial graph $G^{(h)} = (V, C', \mathbf{A}^{(h)})$
    Add $e^*$ to $D_{P-\text{FGA}}$
end
return $D_{P-\text{FGA}}$
```

Algorithm 1: Parallel fast gradient attack (P-FGA).

```
Input: $G' = (V, C', \mathbf{A})$, attack target set $V_t$, perturbation budget $\Delta$
Output: perturbation set$D_{P-\text{NETTACK}}$
Train the surrogate model $\mathbf{Z}$’ on original graph $G'$ to obtain $\mathbf{W}$
Initialize $\mathbf{A}^{(0)} = \mathbf{A}$
Initialize perturbation set $D_{P-\text{NETTACK}}$
for $h = 1$ to $\Delta$ do
    Construct the valid candidate perturbations set $C_{\text{pert}}$
        $\Lambda(G', G^{(h)}) < 0.004$ and $v_i \in V_t \vee v_j \in V_t$
        where $(v_i, v_j) \in C_{\text{pert}}, G^{(h)} = (V, C\prime, \mathbf{A}^{(\mathbf{h})})$ and $a_{ij}^{(h)} = a_{ji}^{(h)} = 1 - a_{ij}^{(h-1)}$
    Select $e^* = (v_m, v_n)$ of the maximum multi-node scoring function value in $C_{\text{pert}}$
        $e^* = (v_m, v_n) = \arg\max_{e\in C_{\text{pert}}} s_{\text{multi}}(e; \mathbf{A}^{(h-1)}, V_t)$
    Obtain the adjacency matrix $\mathbf{A}^{(\mathbf{h})}$ by $a_{mn}^{(h)} = a_{nm}^{(h)} = 1 - a_{mn}^{(h-1)}$
    Generate a new adversarial graph $G^{(h)} = (V, C', \mathbf{A}^{(h)})$
    Add $e^*$ to $D_{P-\text{NETTACK}}$
end
return $D_{P-\text{NETTACK}}$
```

Algorithm 2: Parallel NETTACK (P-NETTACK).

We denote $\mathbf{D}'_{P-\text{NETTACK}}$ as the set consisting of the perturbations in $\mathbf{D}_{P-\text{NETTACK}}$ but not in $\mathbf{D}_{\text{comb}}$. Similarly, $\mathbf{D}'_{P-\text{FGA}}$ contains the perturbations in $\mathbf{D}_{P-\text{FGA}}$ but not in $\mathbf{D}_{\text{comb}}$. Δ' is the difference between Δ and the number of $\mathbf{D}_{\text{comb}}$. Besides, we use a supplementary factor k to control the proportion of supplementary perturbations from $\mathbf{D}'_{P-\text{NETTACK}}$. Specially, we randomly select $[k \cdot \Delta']$ and $\Delta' - [k \cdot \Delta']$ perturbations from $\mathbf{D}'_{P-\text{NETTACK}}$ and $\mathbf{D}'_{P-\text{FGA}}$, respectively, and add them to the $\mathbf{D}_{\text{comb}}$, forming the final unified perturbation set. The pseudocode for intersection and supplement mechanism of perturbations is given in Algorithm 4.

4. Experiments

4.1. Dataset and Environment. We use the well-known politician socialnet Polblogs [19] as our experimental dataset to evaluate our methods. The basic statistics are summarized in Table 1, and only the largest connected component is considered. We randomly choose 20% nodes in the dataset as the labeled nodes for training. The testing set consists of the rest of the unlabeled nodes.

We also give our experimental environment configuration in Table 2.

4.2. Target Parameters and Baselines. Our GCN as an attack target is constructed based on the program on the Github (https://github.com/tkipf/gcn). We train all models for a maximum of 200 epochs using Adam [23] with a learning rate of 0.01. We initialize weights using the initialization described in Glorot and Bengio [24] and accordingly (row-) normalize input feature vectors.

We compare our proposed attack method with comprehensive state-of-the-art adversarial attack methods

```
Input: perturbed graph $G^{adv} = (V, C, \mathbf{A}')$, attack target set $V_t$, node classification model $\mathbf{Z}$
Output: filtered attack target set $V_t'$
Initialize $V_t' = V_t$
for each $v \in V_t$ do
    Predict the label of $v$ in $G^{adv}$ by $\mathbf{Z}$
    if $c_v$ of ground truth is not equal to prediction result then
    Remove $v$ from $V_t'$ //filtering
end
if $|V_t'| == 0$ then
    $V_t' = V_t$ //reset
return $V_t'$
```

Algorithm 3: Filtering mechanism.

```
Input: $D_{P-\mathrm{FGA}}$, $D_{P-\mathrm{NETTACK}}$, supplementary factor $k$, perturbation budget $\Delta$
Output: $D_{\mathrm{comb}}$
Execute the intersection of $D_{P-\mathrm{FGA}}$ and $D_{P-\mathrm{NETTACK}}$ to obtain $D_{\mathrm{comb}}$
$D_{\mathrm{comb}} = D_{P-\mathrm{FGA}} \cap D_{P-\mathrm{NETTACK}}$
if $|D_{\mathrm{comb}}| < \Delta$ then
    Obtain $D'_{P-\mathrm{NETTACK}} = D_{P-\mathrm{NETTACK}} - D_{\mathrm{comb}}$
    Obtain $D'_{P-\mathrm{FGA}} = D_{P-\mathrm{FGA}} - D_{\mathrm{comb}}$
    Calculate $\Delta' = \Delta - |D_{\mathrm{comb}}|$
    Randomly add $[k \cdot \Delta']$ perturbations from $D'_{P-\mathrm{Nettack}}$ to $D_{\mathrm{comb}}$
    Randomly add $[\Delta' - k \cdot \Delta']$ perturbations from $D'_{D-\mathrm{FGA}}$ to $D_{\mathrm{comb}}$
return $D_{\mathrm{comb}}$
```

Algorithm 4: Intersection and supplement mechanism.

Table 1: Dataset statistics of Polblogs.

Nodes	Edges	Classes	Maximum degree	Minimum degree	Average degree
1222	16714	2	351	1	27.4

including FGA and NETTACK. We use codes of the baselines provided by their authors.

(i) **FGA** [16] extracts the gradient of pairwise nodes based on the adversarial network and then selects the pair of nodes with maximum absolute link gradient to realize the attack and update the adversarial network.

(ii) **NETTACK** [17] designs adversarial attacks based on a static surrogate model and greedily selects the optimal perturbation through preserving the key structural features of a graph.

(iii) **Random attack** randomly perturbs the edges related to target nodes.

5. Evaluations

5.1. Evaluation Metric

5.1.1. Attack Success Rate (ASR). ASR is the ratio of the number of successfully attacked nodes to the total number of target nodes, which can be calculated as follows:

$$\mathrm{ASR} = \frac{n_{\mathrm{succ}}}{|V_t|}, \tag{17}$$

where n_{succ} denotes the number of successfully attacked nodes and V_t is the attack target set.

5.1.2. Average Attack Speed (AAS). AAS refers to average running time of each attack, and it can be calculated as follows:

$$\mathrm{AAS} = \frac{t_{\mathrm{total}}}{\Delta}, \tag{18}$$

where t_{total} denotes the total attack time on target set V_t, and Δ is the perturbation budget.

5.1.3. Test Statistic Λ. Test statistic Λ is used to evaluate attack stealthiness (see equation (13)), which measures the structural difference between original graph and adversarial graph. A smaller Λ means that the degree distribution of the adversarial graph is more similar to the original graph's, and thus, the perturbations are more unnoticeable.

Table 2: Experimental environment configuration.

Experimental environment	Environmental configuration
Operating system	Windows 10
CPU	2.4 GHz intel core i5
Memory	16 GB
Hardware	500G
Software	Python 3.6

5.2. ASR Analysis. In our experiments, each attack target set consists of five target nodes, and all of them are from the test set that has been classified correctly in original graph. We divided the perturbation budgets into five levels according to the sum of degrees of all target nodes in the attack target set V_t, i.e., $\Delta \in \{(1/5)d_{\text{sum}}, (2/5)d_{\text{sum}}, (3/5)d_{\text{sum}}, (4/5)d_{\text{sum}}, (5/5)d_{\text{sum}}\}$.

As we can see from Table 3, for each Δ, we compare ASR among P^*, P-NETTACK ($k = 1$), P-FGA ($k = 0$), NETTACK, FGA, and Random Attack, in which P^* is the best value of our unified approach. From Algorithm 4, we know that if $k = 1$, our unified method can be simplified as P-NETTACK; and if $k = 0$, our unified method can be simplified as P-FGA. $P*$ has the highest ASR values of 0.715, 0.880, 1, and 1 at Δ_1, $\Delta_2, \Delta_4, \Delta_5$, respectively. When there is a quite low budget Δ_1, the ASR of P^* is over 15% higher than that of NETTACK or FGA. P-NETTACK ($k = 1$) and P-FGA ($k = 0$) have extremely close values for all budget Δ. Figure 4 shows the visual comparison in Table 3.

In Table 4, we can see that, for Δ_1, Δ_2, Δ_3, our approach achieves highest *ASR* values of 0.715 ($k = 0.5$), 0.880 ($k = 0.7$), and 0.987 ($k = 0.8$), respectively. At Δ_4, Δ_5, for many k settings, ASR values can reach 1. For example, at Δ_4, ASR = 1 when $k = 0.1, 0.2, 0.3, 0.4, 0.5, 0.7$. Figure 5 shows the detailed *ASR* variation along with k increment.

5.3. Test Statistics Λ Analysis. As we can see from Table 5, P-NETTACK ($k = 1$) has the lowest Λ values of 0.003, 0.005, 0.005, and 0004 at Δ_1, $\Delta_2, \Delta_4, \Delta_5$, respectively. Although P-NETTACK ($k = 1$) and NETTACK have the same constraint mechanism, theΛ values of P-NETTACK ($k = 1$) are always lower than those of NETTACK. For P-FGA ($k = 0$) and FGA, which have not enforced the constraint, the Λ values are extremely higher and continue increasing with the increment of Δ. Figure 6 shows the visual comparison in Table 5.

In Table 6, we can see that, for all $\Delta \in \{\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5\}$, with the increment of k, the test statistics Λ keep decreasing, towards better results. Figure 7 clearly shows the Λ variation along with k increment.

5.4. AAS Analysis. As we can see from Table 7, P-NETTACK is the most time-consuming adversarial attack method, with an average of 11.17s of each attack. Since the candidate perturbation set of P-NETTACK is larger than that of NETTACK, the *AAS* of P-NETTACK is much higher than that of NETTACK. Instead, P-FGA and FGA have extremely close AAS values, 0.17s and 0.14s, respectively.

5.5. Filtering Mechanism Analysis. In Table 8, we can see that, for all $\Delta \in \{\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5\}$, the filtering mechanism can greatly improve ASR, with nearly 20% average increment. And for P-FGA, the ASR values at Δ_2 are higher than those of P-FGA (without filtering) at Δ_5. Thus, we can see that the filtering mechanism plays a quite important role for P-NETTACK and P-FGA.

6. Related Work

6.1. Politician Socialnet Analysis. In the last few years, social media has become an important political communication channel, attracting a lot of studies. Adamic and Glance [19] analyzed the political blogosphere over the period of two months preceding the US Presidential Election of 2004, measuring the degree of interaction between liberal and conservative blogs and revealing many interesting differences between the two communities such as linking patterns and discussion topics. Caton et al. [25] presented a Social Observatory, which focused on public Facebook profiles of 187 German politicians from five federal parties, observing how they interacted with constituents, measuring sentiment difference between the politicians and their followers, and analyzing online speech patterns of different parties. Stieglitz and Dang-Xuan [26] proposed a social media analytics framework in political context, aiming at continuously collecting, storing, monitoring, analyzing, and summarizing politically relevant user-generated content from different social media to gain a deeper insight into political discourse in social media.

However, few studies focus on the security analysis of politician socialnet including politician label classification from the perspective of adversarial graph attack. In comparison, we focus on studying security issues of politician socialnet based on graph structure, targeting a GCN model for politician label classification. Interestingly, politician socialnet is highly vulnerable, and the attack cost is quite cheap only by deleting few existing interactions or adding few new interactions. As an important communication bridge between politicians and citizens, the security analysis of politician socialnet should be highly valued.

6.2. Adversarial Attack on Graphs. Recently, some studies have investigated the adversarial attack on neural networks for graph structure. Zügner et al. [17] first revealed the existence of adversarial attack against GCN in node classification task, by slightly modifying graph structure or node attributions to lead to misclassification of a target node. Dai et al. [27] studied test-time nontargeted adversarial attacks on both node classification task and graph classification [28] task based on reinforcement learning. In addition to white-box attack scenario, they also extended their attack method into practical black-box and restricted black-box attack scenarios. Zhang et al. [29] systematically investigated the vulnerability of knowledge graph embedding for the first time. By adding or deleting facts in the knowledge graph, they destroyed the relation prediction model based on representative knowledge graph embedding methods

Table 3: ASR comparison between P^*, P-NETTACK, P-FGA, NETTACK, FGA, and Random Attack.

Perturbation budget	P^*	P-NETTACK ($k=1$)	P-FGA ($k=0$)	NETTACK	FGA	Random attack
$\Delta_1 = 1/5d_{\text{sum}}$	**0.715 ($k=0.5$)**	0.710	**0.715**	0.453	0.480	0.0
$\Delta_2 = 2/5d_{\text{sum}}$	**0.880 ($k=0.7$)**	0.867	0.874	0.857	0.877	0.024
$\Delta_3 = 3/5d_{\text{sum}}$	0.987 ($k=0.8$)	0.963	0.953	**0.995**	0.985	0.025
$\Delta_4 = 4/5d_{\text{sum}}$	**1**	0.993	**1**	0.992	0.992	0.012
$\Delta_5 = 5/5d_{\text{sum}}$	**1**	0.992	**1**	**1**	**1**	0.01

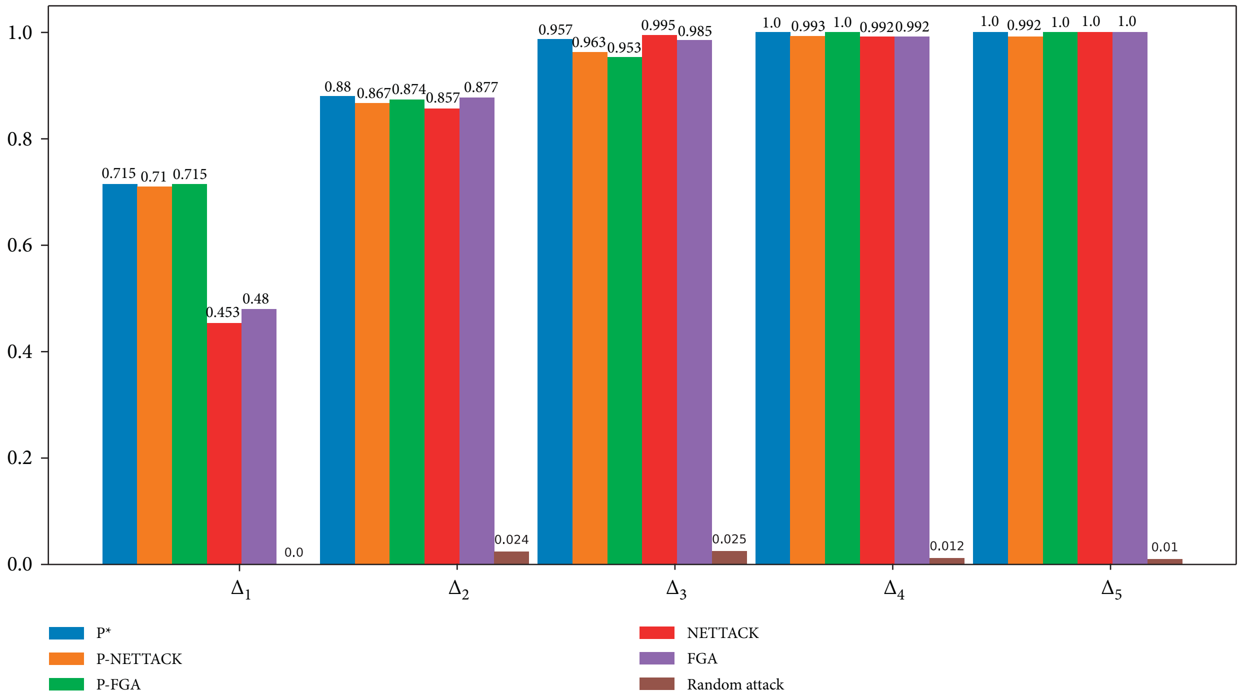


Figure 4: ASR comparison under different perturbation budget Δ.

Table 4: ASR variation under different supplementary factor k.

Perturbation budget Δ	$k=0.1$	$k=0.2$	$k=0.3$	$k=0.4$	$k=0.5$	$k=0.6$	$k=0.7$	$k=0.8$	$k=0.9$
$\Delta_1 = 1/5\ d_{\text{sum}}$	0.705	0.713	0.706	0.711	**0.715**	0.703	0.695	0.698	0.697
$\Delta_2 = 2/5\ d_{\text{sum}}$	0.864	0.874	0.878	0.865	0.86	0.879	**0.880**	0.867	0.878
$\Delta_3 = 3/5\ d_{\text{sum}}$	0.958	0.967	0.962	0.978	0.982	0.983	0.973	**0.987**	0.978
$\Delta_4 = 4/5\ d_{\text{sum}}$	**1**	**1**	**1**	**1**	**1**	0.997	**1**	0.997	0.997
$\Delta_5 = 5/5\ d_{\text{sum}}$	**1**	**1**	**1**	**1**	**1**	**1**	0.997	0.998	0.998

including TransE [30] and RESCAL [31], which is also the first investigation on adversarial attack for heterogeneous graph. Chen et al. [16] explored the adversarial attack on both node classification task and community detection task [32] based on GCN-based gradient information.

However, most works about adversarial attack on node classification only focus on the per-node attack, aiming to achieve misclassification for a target node. Although, for those per-node attack methods, the multinode attack can be performed in a sequential way, the perturbation influence of different per-node attacks is overlooked. In comparison, our parallel attack method, which considers all target nodes and perturbation influence at the same time, is better for multinode attack. In addition, as the first to propose the parallel attack on graph structure, our work can provide an inspiration for adversarial attack on other tasks in a parallel way, such as parallel adversarial attack on prediction of multiple links.

In addition to the benefits mentioned above, the main drawback of our method is that it is time-consuming, especially the P-NETTACK (see Table 7), due to the reason that, at each iteration, more candidate perturbations are taken into computation compared with sequential per-node attack. One of the solutions is developing more computationally efficient test statistic function and scoring function. On the other hand, proposing a perturbation filtering mechanism to reduce the size of multinode candidate perturbations set is also an effective way. In addition, our method does not consider the constraints of attributed graphs [33], such as attribution-based node similarity

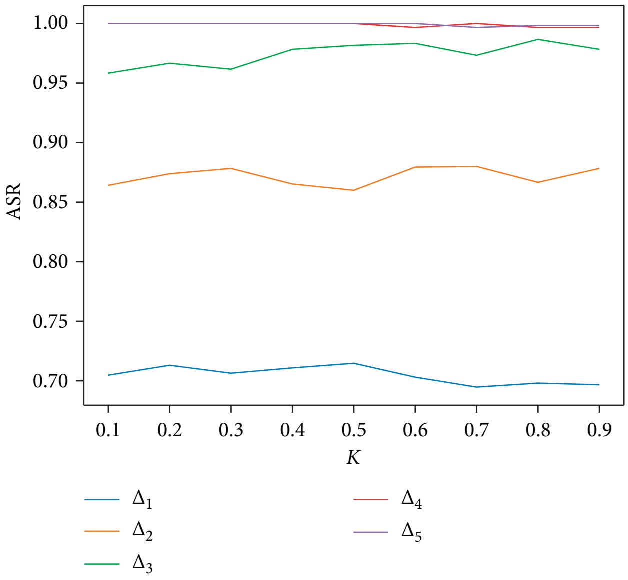


Figure 5: k influence on ASR value among different perturbation budgets Δ.

Table 5: Test statistics Λ comparison between P^*, P-NETTACK, P-FGA, NETTACK, and FGA.

Perturbation budget Δ	P^*	P-NETTACK $(k=1)$	P-FGA $(k=0)$	NETTACK	FGA
$\Delta_1 = 1/5\ d_{\text{sum}}$	0.005	**0.003**	0.035	0.008	0.022
$\Delta_2 = 2/5\ d_{\text{sum}}$	0.006	**0.005**	0.100	0.013	0.091
$\Delta_3 = 3/5\ d_{\text{sum}}$	**0.005**	0.007	0.147	0.017	0.126
$\Delta_4 = 4/5\ d_{\text{sum}}$	**0.005**	**0.005**	0.166	0.016	0.134
$\Delta_5 = 5/5\ d_{\text{sum}}$	0.006	**0.004**	0.204	0.014	0.156

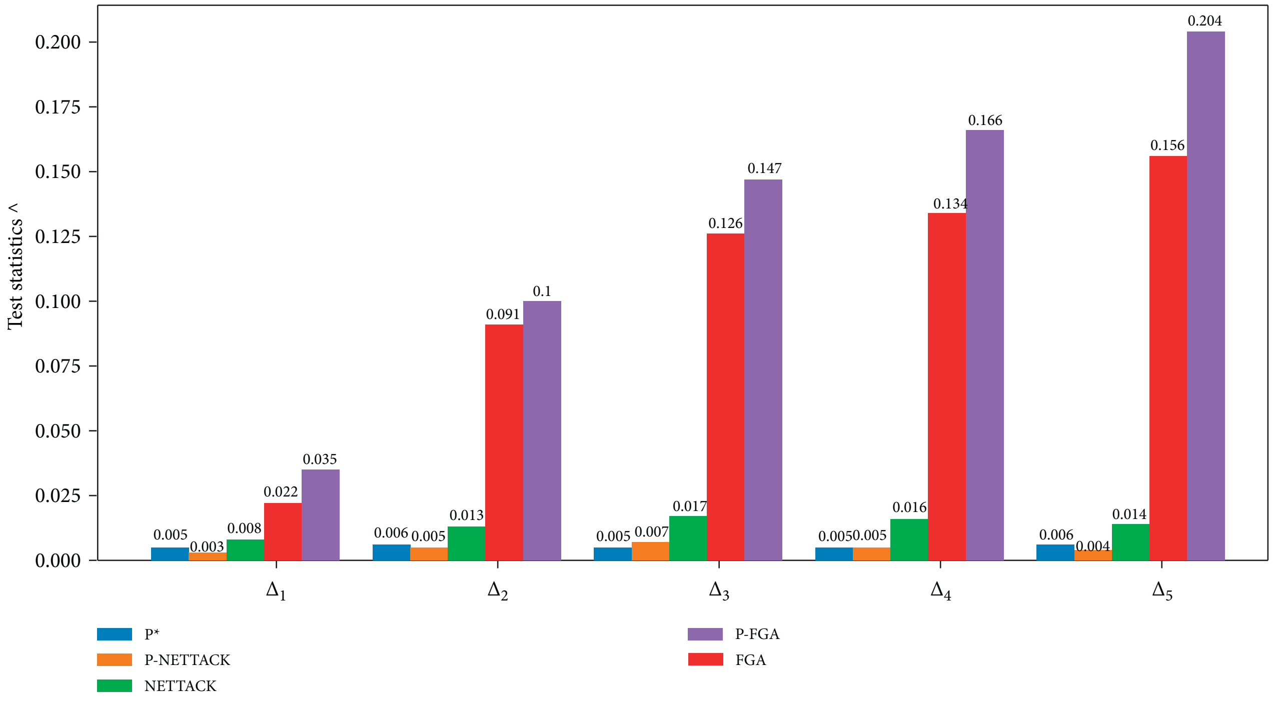


Figure 6: Test statistics Λ comparison under different perturbation budgets Δ.

Table 6: Test statistics Λ variation under different supplementary factors k.

Perturbation budget Δ	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$	$k = 0.6$	$k = 0.7$	$k = 0.8$	$k = 0.9$
$\Delta_1 = 1/5\ d_{\text{sum}}$	0.033	0.026	0.018	0.017	0.014	0.011	0.008	0.006	0.005
$\Delta_2 = 2/5\ d_{\text{sum}}$	0.092	0.059	0.050	0.035	0.026	0.020	0.010	0.008	0.006
$\Delta_3 = 3/5\ d_{\text{sum}}$	0.124	0.098	0.081	0.045	0.033	0.021	0.019	0.006	0.005
$\Delta_4 = 4/5\ d_{\text{sum}}$	0.138	0.110	0.077	0.066	0.035	0.027	0.017	0.011	0.005
$\Delta_5 = 5/5\ d_{\text{sum}}$	0.160	0.120	0.101	0.080	0.051	0.028	0.019	0.013	0.006

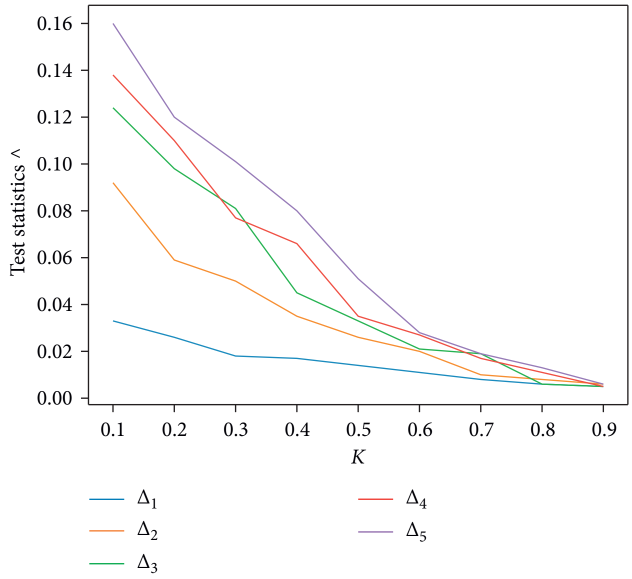


Figure 7: k influence on test statistics Λ value among different perturbation budgets Δ.

Table 7: AAS comparison between P-NETTACK, P-FGA, NETTACK, and FGA.

Methods	AAS (second)
P-NETTACK	11.17
P-FGA	0.17
NETTACK	1.30
FGA	0.14

Table 8: Filtering influence on *ASR* of P-NETTACK and P-FGA.

Perturbation budget Δ	P-NETTACK	P-NETTACK (without filtering)	P-FGA	P-FGA (without filtering)
$\Delta_1 = 1/5\ d_{\text{sum}}$	0.710	0.45	0.715	0.551
$\Delta_2 = 2/5\ d_{\text{sum}}$	0.867	0.662	0.874	0.653
$\Delta_3 = 3/5\ d_{\text{sum}}$	0.963	0.736	0.953	0.721
$\Delta_4 = 4/5\ d_{\text{sum}}$	0.993	0.818	1	0.759
$\Delta_5 = 5/5\ d_{\text{sum}}$	0.992	0.887	1	0.813

constraint [34] and attribution cooccurrence constraint [17]. Parallel multinode adversarial attack on attributed graph and Heterogeneous Information Network (HIN) [35] still needs further exploration.

7. Conclusions

In this paper, we propose a multinode parallel adversarial attack framework on node classification in socialnet of graph structure, based on considering perturbation influence between per-node attacks. Through redesigning new loss function and objective function for nonconstraint and constraint perturbations, respectively, and constructing intersection and supplement mechanisms of perturbation, we integrate nonconstraint P-FGA and constraint P-NETTACK into a unified attack framework. Based on politician socialnet Polblogs of 1222 nodes and 16714 edges, we evaluate attack success rate, test statistics, and average attack speed for our approach. Our approach shows a high attack success rate of 71.5% at the lowest perturbation budget of $1/5\ d_{\text{sum}}$, keeping a satisfied test statistic of 0.005.

This work severs as a first step to take security analysis on multinode parallel adversarial attack in politician socialnet. It is expected to inspire a series of follow-up studies, including but not limited to (1) adversarial attack on prediction of multiple links; (2) more concrete defense design and implementation.

Data Availability

The dataset of Polblogs can be obtained from http://networkrepository.com/polblogs.php.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (61972025, 61802389, 61672092, U1811264, and 61966009), the National Key R&D Program of China (2020YFB1005604 and 2020YFB2103800), Fundamental Research Funds for the Central Universities of China (2018JBZ103 and 2019RC008), and Guangxi Key Laboratory of Trusted Software (KX201902).

References

[1] L. Strate, "The varieties of cyberspace: problems in definition and delimitation," *Western Journal of Communication*, vol. 63, no. 3, pp. 382–412, 1999.

[2] J. Li, J. Li, and X. Chen, "MobiShare+: security improved system for location sharing in mobile online social networks," *Journal of Internet Services and Information Security*, vol. 4, no. 1, pp. 25–36, 2014.

[3] A. Branitskiy, D. Levshun, N. Krasilnikova et al., "Determination of young generation's sensitivity to the destructive stimuli based on the information in social networks," *Journal of Internet Services and Information Security*, vol. 9, no. 3, pp. 1–20, 2019.

[4] D. F. Nettleton, "Data mining of social networks represented as graphs," *Computer Science Review*, vol. 7, pp. 1–34, 2013.

[5] M. Kolomeets, A. Benachour, D. El Baz et al., "Reference architecture for social networks graph analysis tool," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 10, no. 4, pp. 109–125, 2019.

[6] M. Kolomeets, A. Chechulin, and I. V. Kotenko, "Social networks analysis by graph algorithms on the example of the VKontakte social network," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 10, no. 2, pp. 55–75, 2019.

[7] W. Niu, G. Li, H. Tang, X. Zhou, and Z. Shi, "CARSA: a context-aware reasoning-based service agent model for AI planning of web service composition," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1757–1770, 2011.

[8] W. Niu, G. Li, Z. Zhao, H. Tang, and Z. Shi, "Multi-granularity context model for dynamic Web service composition," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 312–326, 2011.

[9] S. Dabhi and M. Parmar, "Nodenet: a graph regularised neural network for node classification," 2020, https://arxiv.org/abs/2006.09022.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, https://arxiv.org/abs/1609.02907.

[11] S. Zhang, H. Tong, and J. Xu, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, p. 11, 2019.

[12] F. Scarselli, M. Gori, and A. C. Tsoi, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[13] Z. Wu, S. Pan, and F. Chen, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[14] D. Zügner and S. Günnemann, "Certifiable robustness of graph convolutional networks under structure perturbations," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery\& Data Mining*, pp. 1656–1665, Virtual Event, CA, USA, July 2020.

[15] L. Sun, Y. Dou, and C. Yang, "Adversarial attack and defense on graph data: a survey," 2018, https://arxiv.org/abs/1812.10528.

[16] J. Chen, Y. Wu, and X. Xu, "Fast gradient attack on network embedding," 2018, https://arxiv.org/abs/1809.02797.

[17] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2847–2856, London, UK, August 2018.

[18] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, pp. 39–57, IEEE, San Jose, CA, USA, May 2017.

[19] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 US election: divided they blog," in *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43, New York, NY, USA, August 2005.

[20] Y. Zhang and J. Koren, "Efficient bayesian hierarchical user modeling for recommendation system," in *Proceedings of the 30th Annual International Acm Sigir Conference on Research and Development in Information Retrieval*, pp. 47–54, Amsterdam, The Netherlands, July 2007.

[21] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.

[22] A. Bessi, "Two samples test for discrete power-law distributions," 2015, https://arxiv.org/abs/1503.00643.

[23] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, Sardinia, Italy, May 2010.

[25] S. Caton, M. Hall, and C. Weinhardt, "How do politicians use facebook? an applied social observatory," *Big Data & Society*, vol. 2, no. 2, 2015.

[26] S. Stieglitz and L. Dang-Xuan, "Social media and political communication: a social media analytics framework," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1277–1291, 2013.

[27] H. Dai, H. Li, and T. Tian, "Adversarial attack on graph structured data," 2018, https://arxiv.org/abs/1806.02371.

[28] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1024–1034, Long Beach, CA, USA, December 2017.

[29] H. Zhang, T. Zheng, and J. Gao, “Towards data poisoning attack against knowledge graph embedding,” 2019.

[30] A. Bordes, N. Usunier, and A. Garcia-Duran, “Translating embeddings for modeling multi-relational data,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2787–2795, Lake Tahoe, CA, USA, December 2013.

[31] M. Nickel, V. Tresp, and H. P. Kriegel, “A three-way model for collective learning on multi-relational data,” *ICML*, vol. 11, pp. 809–816, 2011.

[32] K. Allab, L. Labiod, and M. Nadif, “A semi-NMF-PCA unified framework for data clustering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 2–16, 2016.

[33] G. Cui, J. Zhou, and C. Yang, “Adaptive graph encoder for attributed graph embedding,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 976–985, San Diego, CA, USA, August 2020.

[34] H. Wu, C. Wang, and Y. Tyshetskiy, “Adversarial examples on graph data: deep insights into attack and defense,” 2019, https://arxiv.org/abs/1903.01610.

[35] Y. Lu, Y. Fang, and C. Shi, “Meta-learning on heterogeneous information networks for cold-start recommendation,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1563–1573, San Diego, CA, USA, August 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6661124, 11 pages
https://doi.org/10.1155/2021/6661124

Research Article

An Improved Feature Extraction Approach for Web Anomaly Detection Based on Semantic Structure

Zishuai Cheng,[1,2] **Baojiang Cui**,[1,2] **Tao Qi**,[3] **Wenchuan Yang**,[1,2] **and Junsong Fu**[1,2]

[1] *School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *The National Engineering Laboratory for Mobile Network, Beijing 100876, China*
[3] *School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Baojiang Cui; cuibj@bupt.edu.cn

Received 1 December 2020; Revised 4 January 2021; Accepted 27 January 2021; Published 11 February 2021

Academic Editor: Zhe-Li Liu

Anomaly-based Web application firewalls (WAFs) are vital for providing early reactions to novel Web attacks. In recent years, various machine learning, deep learning, and transfer learning-based anomaly detection approaches have been developed to protect against Web attacks. Most of them directly treat the request URL as a general string that consists of letters and roughly use natural language processing (NLP) methods (i.e., Word2Vec and Doc2Vec) or domain knowledge to extract features. In this paper, we proposed an improved feature extraction approach which leveraged the advantage of the semantic structure of URLs. Semantic structure is an inherent interpretative property of the URL that identifies the function and vulnerability of each part in the URL. The evaluations on CSIC-2020 show that our feature extraction method has better performance than conventional feature extraction routine by more than average dramatic 5% improvement in accuracy, recall, and F1-score.

1. Introduction

Web attack still is one of the largest IT security threats with many types of Web attacks (e.g., SQL injection, cross-site scripting, and Web-shell) in the rapid development of 5G, IoT, and cloud computing. Web-based applications provide various services, such as e-commerce, e-government, e-mail, and social networking, for individuals and organizations [1, 2]. Users usually store their sensitive data on these applications. The importance and sensitivity of the Web-based application make it into an attractive target for attackers. Defending Web-based applications from attacks is a challenging task because cyber-defence is asymmetric warfare as the attackers have great advantage than defenders [3]. The intrusion detection system continuously identifies attacks relying on the up-to-date signature or model, while attacker only needs a single vulnerability for victory. Unknown attacks, specifically *Zero-day*, are difficult to identify by the signature-based intrusion detection system and can cause great damage to individuals and organizations.

To detect unknown attacks, a great number of anomaly-based intrusion detection methods have been proposed by researchers in recent years. The anomaly detection method can detect unknown attacks by identifying their abnormal behaviours that obviously deviate from the normal behaviours which have been modelled in the training phase [4, 5]. No matter which specific algorithm (i.e., support vector machine, hidden Markov model, and random forests) was used to profile the normal behaviours, feature extraction is essential to the anomaly-based detection model. The widely used feature extraction methods can be classified into two types: expert knowledge-based models and NLP-based models as follows:

(i) In expert knowledge-based approaches, researchers design a set of handcrafted rules to describe the normal or malicious behaviour of HTTP request, such as whether exits sensitive keyword, the length of each value, and whether contains special character [6, 7]

(ii) In NLP-based approaches, researchers extract contiguous sequences of n characters from the URL [8–11]

Although these methods have achieved a good performance, they roughly treat HTTP request URL as a general string that consists of letters and pay average attention to each character.

Semantic structure is a knowledge that is comprised of a set of information entities, such as the deserving Web resource, number and sequence of logical parts, and the property of each logical part (trivial or salient) [12]. A resource is a function that provides a type of interaction for users by Web application. Consider an e-commerce application, the function can be register, login, view products, or order products. In general, URLs requesting same resource (or function) have the identical semantic structure although the values of logical parts are variable. In a request URL, each logical part plays different roles. Salient logical parts are mostly be used to indicate requesting resource. Values of these parts are stationary or only have a few numbers of values. On the contrary, trivial logical parts are always used to deliver users' input payloads to the server-side program, such as username, page number, delivery address, or product ID.

To the best of our knowledge, the utilization of semantic structure for feature extraction has not been investigated. We see a good reason to believe that the insights gained in the semantic structure carry over to feature extraction. In general, the attacker always manipulates the values of trivial logical parts to attack the Web-based application. On the contrary, the values of salient logical parts are rarely be used to launch attacks. Thus, we should pay more attention to the values of trivial logical parts rather than pay average attention to every logical parts in intrusion detection.

In our preliminary work [13], we introduced an anomaly detection method based on the semantic structure. However, it has some limitations in HTTP request imbalance. Hence, in this paper, we proposed an improved feature extraction approach that efficiently uses the semantic structure. This approach helps the anomaly-based detection model pay more attention to sensitive trivial parts which are more likely used by the attacker to launch attacks. A method that can automatically learn semantic structure by observing training dataset is proposed in this paper. We further eliminate the request imbalance by using skeleton structure to improve the accuracy of the semantic structure. Request imbalance is a serious problem which is caused by the fact that some functions are requested more frequently than others, such as viewing product function is more likely to be requested than ordering product function. The evaluation results show the anomaly-based detection models with the semantic structure outperform other models that were built with conventional feature extraction procedure.

To learn the semantic structure and use it to help build a detection model, we first define a notion of skeleton structure for the URL and classify URLs into several subgroups based on their skeleton structure. Then, we propose a statistical-based algorithm to learn the semantic structure from each group, respectively, and then combine these independent semantic structures into an entire semantic structure. Pattern-tree which is proposed by Lei et al. is used to encode the semantic structure [12]. After that, we build the anomaly-based detection model for each trivial logical part by observing their values. Finally, we introduce how to detect anomaly attacks based on the semantic structure and the built detection model.

Based on the semantic structure, the anomaly detection model can pay more attention to detect the values of trivial logical parts. Thus, the detection model using semantic structure is more sensitive and precise to detect attacks.

The contributions of this paper can be summarized as follows:

(i) An enhanced feature extraction approach is proposed for Web anomaly detection. This approach takes the advantage of semantic structure to pay more attention to trivial logical parts which are more vulnerable than salient parts. Compared with conventional feature extraction methods, the significant innovation is that we treat the URL as a combination of meaningful logical parts rather than meaningless string that consists of letters.

(ii) We proposed a notion of skeleton structure which is used to eliminate the request-imbalance problem. This method can improve the accuracy of the learned semantic structure.

(iii) We evaluate our approach on CSIC-2010 dataset [14]. Experimental results show that the semantic structure is vital to improving the performance of the anomaly-based intrusion model.

The rest of this paper is organized as follows. In Section 2, we introduce the related work focusing on anomaly-based detection and semantic structure. The framework of our approach and the details of how to learn semantic structure are separately introduced in Sections 3 and 4. The method that to build the anomaly-based detection model for each trivial logical part is described in Section 5. In Section 6, we illustrate how to use semantic structure and the built detection model to detect attacks. In Section 7, we report the simulation environment and experiment results. Finally, we draw conclusions and future points in Section 8.

2. Related Work

Since anomaly-based intrusion detection was firstly introduced in 1987 by Denning [15], research in this area has been rapidly developed and attracted lots of attention. A great number of methods have been proposed by researchers in recent years. According to the types of algorithms that used to build the detection model, the anomaly-based WAF can be categorized into statistics, data mining, machine learning, and deep learning-based. No matter which specific algorithm is used, feature extraction always is an important part of building the anomaly-based detection model. The feature extraction methods can be widely divided into expert knowledge-based and NLP-based.

In the field using expert knowledge to extract features from URLs, Cui et al. proposed a feature extraction approach which extracts 21 features from HTTP request based on domain knowledge to describe the behaviour of HTTP request [7]. Then, they train a random forest (RF) classification model based on these features to classify HTTP request as normal and anomaly. Niu and Li extracted eight features with good classification effect to augment the original data [16]. Tang et al. proposed an approach that extracts behaviour characteristics of SQL injection based on the handcrafted rules and uses the long short-term memory (LSTM) network to train a detection model [17]. And, authors combined expert knowledge with N-gram feature for reliable and efficient Web attack detection and used the generic-feature-selection (GFS) measurement to eliminate redundant and irrelevant features in [18, 19]. Zhou and Wang proposed an ensemble learning approach to detect XSS attack [20]. The ensemble learning approach uses a set of Bayesian networks which is built with both domain knowledge and threat intelligence. More recently, Tama et al. proposed a stack of the classifier ensemble method which relies on the handcrafted features [21]. All these authors extract features mostly based on their expert knowledge. These handcrafted features have achieved a good performance in these datasets. However, there exists a strong difference between the network environments or the behaviours of Web applications. These selected features that perform well in one training dataset may not perform well in other Web applications.

To address the problem of expert knowledge-based feature extractions, lots of researchers use natural language processing (NLP) and neural network (NN) to automatically learn the significant features and build a powerful anomaly detection model. Kruegel et al. proposed an anomaly detection system for Web attacks, which takes advantage of the particular structure of HTTP query that contains parameter-value pairs [22, 23]. In this paper, authors built six models to detect attacks in different aspects such as attribute's length, character distribution, structural inference, token finder attribute presence or absence, and attribute order and separately output the anomaly probability value. The request is marked as malicious if one or more features' probability exceeds the defined threshold. Cho and Cha proposed a model which uses Bayesian parameter estimation to detect anomalous behaviours [24]. PAYL is proposed by Wang and Stolfo which uses the frequency of N-grams in the payload as features [25]. Tian et al. used continuous bag of words (CBOW) and TF-IDF to transform the HTTP request into vector [26, 27]. Both are the popular algorithms for text analysis in the field of NLP. Wu et al. exploited word embedding techniques in NLP to learn the vector representations of characters in Web requests [28]. Tekerek used bag of words (BOW) to produce a dictionary and convert HTTP request as a $200 \times 170 \times 1$ matrix [29]. If the payload matches an entry in the dictionary, the label is set to 1 that is represented with white pixel in image; if it does not, it is set to 0 that is represented with black pixel in image. Then, Tekerek used the conventional neural network (CNN) to learn the normal pattern of HTTP request and detects attacks. All these authors focus their efforts on solving the problem of how to build the behaviour models that can significantly distinguish abnormal behaviour from normal behaviour without a lot of human involvement. They ignore the semantic structure of HTTP request and treat the URLs as a general string that is comprised of letters and extract features directly from these URLs.

No matter using expert knowledge-based feature extraction methods or N-gram-based methods, the anomaly detection model will pay average attention to every letter or logical part. Thus, these models are taking the negative effects of some redundant letters or useless logical parts. Thus, it is necessary to use semantic structure to help the model pay more attention to those vulnerable logical parts.

To the best of our knowledge, there are few Web instruction detection methods that use the semantic structure of URLs. However, in other research areas, some researchers had taken advantage of it. Lei et al. proposed a concept of pattern-tree to learn the semantic structure of URLs [30]. They proposed a top-down strategy to build a pattern-tree and used statistic information of the values of logical parts to make the learning process more robust and reliable. Yang et al. further proposed an unsupervised incremental pattern-tree algorithm to construct a pattern-tree [31]. Our approach that is used to learn semantic structure is inspired by these works. However, in our approach, we take account of the negative effect of request imbalance that wildly exists in Web applications and we introduce a concept of skeleton to eliminate request imbalance.

3. Framework and Definition

Without loss of generality, we mainly analyse the HTTP requests using the GET method in this paper. Although we focus on GET requests here, our method can be extended to other methods easily by converting users' data to the parameter-value pairs format that is similar to GET.

As shown in Figure 1, our approach is composed of three steps. In the learning step, we eliminate the request-imbalance problem, learn separate semantic structure from each subgroup, and merge these independent subsemantic structures into an entire semantic structure. Then, in the building anomaly-based detection model step, we build models for each trivial logical part by observing its values. In the detection step, the method that classifies new HTTP request as normal or abnormal based on the semantic structure and learned model is proposed.

Before introducing our model, we first define the training dataset of URLs as $U = \{u_1, u_2, \ldots, u_m\}$, in which u_i is the i^{th} request URL. According to HTTP protocol [32],

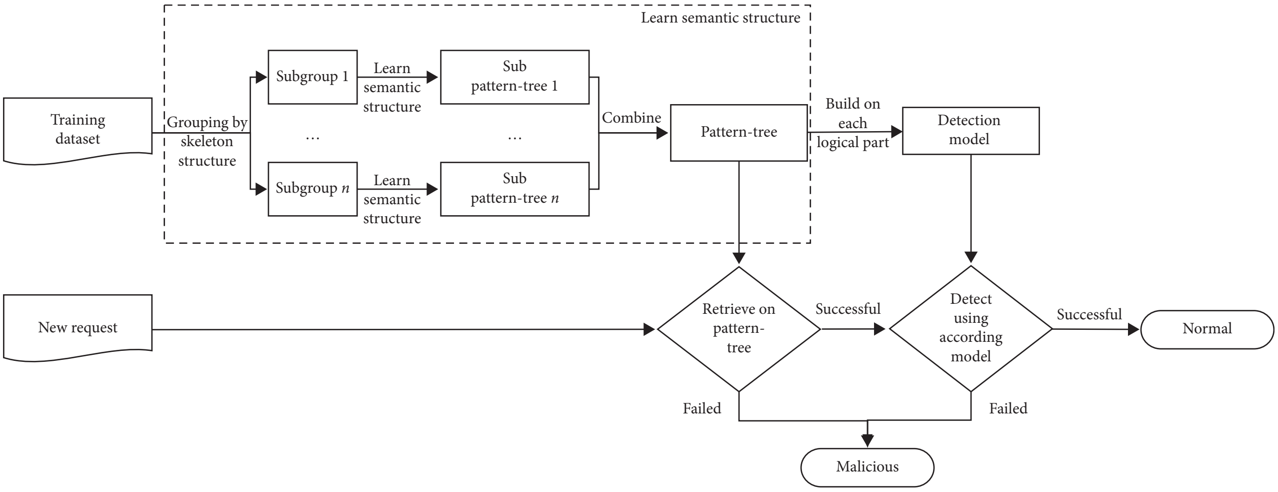


Figure 1: The framework of the improved feature extraction approach.

each request URL u can be decomposed into several components (e.g., scheme sch, authority auth, path path, optional path information component opinfo, and optional query string query) by delimiters like ":," "/," and "?." Components before "?" are called static parts (e.g., scheme, authority, path, and opinfo), and the rest of components (e.g., query) are called dynamic part. path can be further decomposed into a collection of parameter-value parts, also called logical parts, according to its hierarchical structure like $pv_{\text{path}} = \{(p_1, v_1), (p_2, v_2), \ldots, (p_n, v_n)\}$, where v_i is the i^{th} segment value in path split by "/" and p_i is the index of v_i represented as "path-i.". The dynamic part query is usually used to transmit the values submitted by end-users to the server-side program. query can further be decomposed into a collection of parameter-value parts or logical parts like $pv_{\text{query}} = \{(p_1, v_1), (p_2, v_2), \ldots, (p_n, v_n)\}$, in which p_i is the name of the i^{th} parameter in query split by "?" and v_i is the corresponding value of the i^{th} parameter. Finally, we combine pv_{path} and pv_{query} into a parameter-value collection pv.

However, the confusion between the function of logical parts in path and query proposes a challenge to determine a logical part is trivial or salient. Path path not only identifies the requesting resource but also sometimes contains the values submitted by end-users. query also can contain the identifier that indicates the requesting resource. Especially, the rapid development of search engine optimization (SEO) aggravates the confusion problem [33, 34]. Thus, we propose a top-down method to infer the function and semantic of each logical part in path and query and learn the semantic structure. This method will be introduced in detail in the next section.

4. Learn Semantic Structure Information

Our method can automatically learn semantic structure in three major steps: eliminating request-imbalance problem, learning semantic structure form each subgroup, and merging all independent part semantic structures into an entire semantic structure.

4.1. Eliminating Request-Imbalance Problem. As noticed before, request imbalance presents a major challenge of learning semantic structure accurately. For example, in an e-commerce website, users are more likely to choose and order products compared with register or login. Thus, logical parts contained in choose and order functions are requesting more frequently than others and have more appearance frequency than others. Thus, these logical parts are more likely determined as salient even if it is trivial.

As we all know, each URL has its basic structure (e.g., scheme, authority, depth of path, number, and sequence of logical parts in query). URLs which request same resource have same basic structure. Thus, we can split URLs into several subgroups based on their basic structures. For a Web application, the scheme and authority are mostly invariant. And thus, in this paper, we mainly use the priorities of path and query to divide URLs into subgroups.

To spilt URLs into subgroups, we firstly extract pv_{path} and pv_{query} for each URL u. Then, we construct a hash key using the size of pv_{path} and the parameter sequence of pv_{query} to split URLs into subgroups. The URLs with the same size and parameter sequence are classified into one subgroup. As shown in Figure 2, we split URLs showed in Table 1 into four subgroups according to their basic structure. After that, we can separately learn semantic structure from each group.

Splitting URLs in subgroups cannot change the fact that there has a request-imbalance problem. However, we can limit the imbalance between URLs to the imbalance between subgroups and ensure the URLs in each subgroup are request-balance. Thus, this method eliminates the impact of the request imbalance on the accuracy of the learned semantic structure.

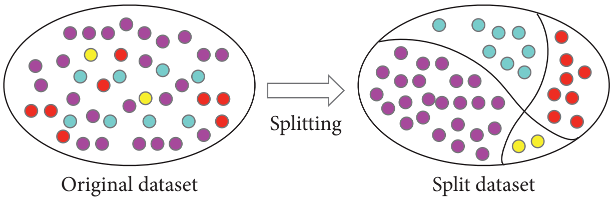


Figure 2: An example that illustrates the processing of splitting. By using this method, we divide those four types of URLs (as shown in Table 1) into four subgroups. Each URL in subgroups has the same basic structure and requests the same resource. Thus, the URLs in each subgroup are balanced.

Table 1: Four examples of URL and each URL represents a type of HTTP requests.

No.	URL	Parameter sequence	Semantic structure
1	/question/search?q = docker	Path 1, Path 2, q	question/search?q = *
2	/question/top?q = windows server &page = 10	Path 1, Path 2, q, page	question/top?q = *&page = *
3	/user/news?page = 1	Path 1, Path 2, page	/user/news?page = *
4	/teams/create/Linux	Path 1, Path 2, Path 3	/teams/create/*

Parameter sequence is the parameter-sequence extracting from pv_{path} and pv_{query}. Semantic structure is the semantic structure information of each type of HTTP requests. The "*" in semantic structure denotes the corresponding part is trivial, and other symbols mean the corresponding segments are salient.

4.2. Learn Semantic Structure and Construct Pattern-Tree. The crucial thing in learning semantic structure is to determine the logical part whether is trivial or salient. In this section, we will introduce the method about learning semantic structure in detail.

According to our observation, different logical parts (or components) play different roles and have distinct different appearance frequencies. In general, salient parts denoting directories, functions, and document types only have a few numbers of values, and these values have high appearance frequencies. In contrast, trivial parts denoting parameters such as usernames and product IDs have quite diverse values, and these values have low appearance frequencies.

Thus, we proposed an approach to determine the property of the logical part based on its entropy and the number of distinct values. The entropy for a logical part is defined as $H(K) = \sum_{i=1}^{V} -(v_i/N)\log(v_i/N)$, where V is the number of distinct values of this part, v_i is the frequency of the i^{th}value, and N is the number of total values. We determine the logical part whether is trivial according to the following equation:

$$\text{logical part } i = \begin{cases} \text{trivial}, & \text{others}, \\ \text{salient}, & \text{if } H(k) < \lambda \log V \text{ or } V, \end{cases} \tag{1}$$

where $\lambda \in [0,1]$ and $\gamma \in \mathbb{N}$ are two hyperparameters to control the sensitivity of the learned semantic structure.

As shown in Algorithm 1, we proposed a top-down algorithm to recursively split the URLs into subgroups and build a pattern-tree in the meantime. We determine the logical part whether is salient or trivial according equation (1) in each splitting process. Values are reserved in V^* if the logical part is salient. Otherwise, values will be generalized as "*" and V^* is set as {'*'}. * is a wildcard character that represents any characters. According to the values in V^*, we can split URLs into subgroups. Then, we further determine the next logical part as salient or trivial on each subgroup recursively. This determining and splitting process is repeated until the subgroup is empty. Finally, we learn a pattern-tree $\mathcal{N}_i$ form subgroup U_i. Each path from the root to leaf in this tree is a piece of semantic structure information. Each node in the pattern-tree represents a logical part of the URL. And, the type of node is identified by to its value.

After applying the construct pattern – tree algorithm to each subgroup, we finally get several independent pattern-trees. Then, we can merge these independent pattern-trees into an entire patten-tree that describes the whole semantic structure of Web application. Figure 3 shows the processing of learning semantic structure and constructing pattern-tree. There are four pattern-trees separately learned on U_1, U_2, U_3, and U_4 using the construct pattern – tree algorithm. Then, we merge these four independent trees into an entire pattern-tree $\mathcal{T}$ as shown on the right. The entire pattern-tree describes the whole semantic structure and is used to build the anomaly detection model and detect attacks.

The entire pattern-tree can be retrieved using parameter-value collection KV. For example, for a request URL "/question/search?q = docker," we retrieve a path on pattern-tree according to parameter-value collection $pv = \{(\text{path}_1, \text{question}), (\text{path}_2, \text{search}), (q, \text{docker})\}$. Firstly, we examine the first parameter-value pair $(\text{path}_1, \text{question})$ on pattern-tree. If the parameter-value pair exists, it indicates that this parameter-value pair is valid and we further examine the next parameter-value pair on the corresponding child-tree. Otherwise, we replace the value of this parameter-value pair with "*" and re-examine it. This process is repeated until all parameter-value pairs in pv are examined or subtree is null or parameter-value pair not exists. For this request URL, the successful retrieval path is shown in Figure 3 and is marked with the red dash arrow. This path shows that the semantic structure is "/question/

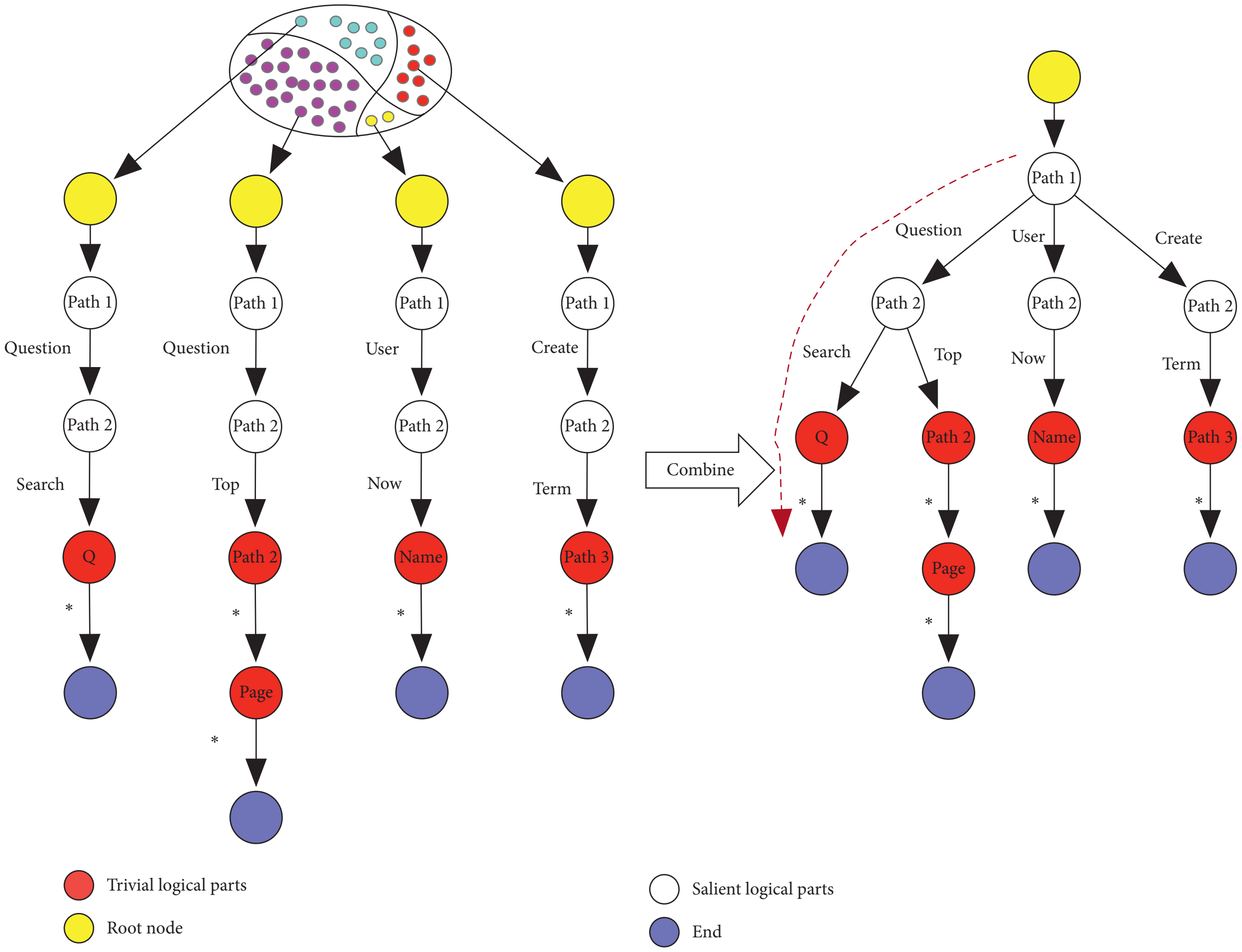


Figure 3: The processing of learning semantic structure and constructing pattern-tree. We first separately learn semantic structure and construct pattern-tree on each subgroup. Then, we merge these independent semantic structures into an entire pattern-tree which describes the whole semantic structure of Web application. The entire pattern-tree is used in building the anomaly detection model and detecting malicious.

search?q = ∗ ," where the parameter q is trivial and the value of parameter q is more vulnerable than others.

5. Build Anomaly Detection Model

As mentioned earlier, the values of trivial logical parts change frequently and depend on users' input. Values of these trivial logical parts are mostly crafted by attackers to attack Web application. Thus, in anomaly detection, we can pay more attention to trivial logical parts to improve the accuracy and efficiency of the detection model.

We firstly split HTTP request URLs U into several subsets $U_1, \cdots$, and U_n according to pattern-tree $\mathcal{T}$, where n is the number of semantic structure pieces in pattern-tree (also is the number of paths from the root to leaves). The subset U_i has the following characters:

(i) $\forall u \in U_i$, u has the same semantic structure

(ii) $\forall i \neq j, U_i \cap U_j = \varnothing$

(iii) $U_1 \cup U_2 \cup \cdots \cup U_n = U$

We further extract the value of each trivial part from a URL u and combine them as a vector $v_{\text{trivial}} = \{v_1, \ldots, v_q\}$, where v_i is the value of i^{th} trivial logical part. Furthermore, we combine v_{trivial} of each u in U_i as a $m \times q$ matrix $\mathrm{P}_{\text{trivial}}$, as shown in equation (2), where m is the numbers of URLs in U_i. The j^{th} column $[v_{1j}, v_{2j}, \ldots, v_{mj}]$ is the value of j^{th} trivial part for all URLs in U_i:

$$P_{\text{trivial}_i} = \begin{bmatrix} v_{11} & \cdots & v_{1q} \\ \vdots & \ddots & \vdots \\ v_{m1} & \cdots & v_{mq} \end{bmatrix}. \qquad (2)$$

We build anomaly-based intrusion detection models for each logical part by observing the corresponding column values in P_{trivial_i}. Finally, each node of pattern-tree that represents a logical part maps a detection model. The entire anomaly detection model M of this Web application is composed of several submodels $\{m_{11}, \ldots, m_{1q_1}, \ldots, m_{21}, \ldots, m_{2q_2}, \ldots, m_{nq_n}\}$, where m_{ij} is built by observing the values of j^{th} column in P_{trivial_i}.

The specific algorithm used to build the anomaly-based detection model is beyond the scope of this paper. Our method can integrate with any anomaly-based detection algorithm to build more precious model for detection attacks.

6. Detect Malicious Attacks

In this section, we will introduce the approach to detect malicious attack according to the pattern-tree $\mathcal{T}$ and anomaly-based detection model M. The URL is detected in the following two levels. (a) Semantic structure level: we retrieve the URL on pattern-tree $\mathcal{T}$ to determine whether the new request matching the exiting semantic structure; (b) Value level: we then detect the values of each trivial logical part whether is anomaly using the corresponding learned anomaly-based detection model M. As long as the new request does not follow the existing semantic structure or any value of trivial logical parts, it will be classified as an anomaly. Otherwise, we determine it as benign.

More specifically, we first convert the URL u into parameter-value collection pv before detecting the HTTP request. Then, we retrieve pattern-tree $\mathcal{T}$ using pv. We simultaneously detect the value of trivial logical part whether is abnormal in the retrieve process. If a value is determined as an anomaly, we stop further retrieving and directly report this HTTP request as abnormal. If the URL of new request does not fit the expectation of $\mathcal{T}$ (e.g., there exists any parameter-value pair that has not examined when a null subtree is reached, and the subtree is not null after all parameter-values pairs are examined), we report the HTTP request as abnormal. Only if the new request satisfies both the semantic structure and anomaly-detection model, we classify it as normal.

7. Experiments

To evaluate the effectiveness of our approach, we implemented a prototype of our proposed method sketched in Figure 1. The components are implemented in Python and Scikit-learn 0.23. And, the dataset used in evaluation experiments is CSIC-2010 [14].

7.1. Experimental Settings

7.1.1. Dataset Description. CSIC-2010 is a modern Web intrusion detection dataset introduced by the Spanish National Research Council which includes two classes: normal and anomalous. It contains thousands of Web requests automatically generated by creating traffic to an e-commerce Web application using Paros proxy and W3AF. The dataset consists of three subsets: 36,000 normal requests for training, 36,000 normal requests, and 25,000 anomalous requests for testing. There are three types of anomalies in this dataset: static attack that requests for the hidden (nonexistent) resource, dynamic attack that craftily modifies the value of logical part to attack Web application, and unintentional illegal request that does not follow the normal semantic structure; however, it has no malicious payload. The dataset consists of HTTP requesting for several resources with two request methods: GET and POST.

7.1.2. Metrics. There are numbers of performance metrics that can be used to evaluate the performance of the anomaly detection system. The most commonly used metrics in this field are precision, recall, F1-score, and accuracy (ACC). In this paper, we also use these metrics to evaluate our approach:

(i) Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives given as follows:

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}. \quad (3)$$

(ii) Recall is defined as the percentage of positive cases you caught given as follows:

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}. \quad (4)$$

(iii) F1-score is the harmonic mean of precision and recall taking both metrics into account given as follows:

$$F_1 = 2^* \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \quad (5)$$

(iv) Accuracy measures in percentage form where instances are correctly predicted given as follows:

$$\text{accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{true positives} + \text{false positives} + \text{true negatives} + \text{false negatives}}. \quad (6)$$

8. Results and Discussion

The hyperparameters λ and γ play significant role to control the accuracy of pattern-tree. With the best λ and γ, the learned pattern-tree achieves an appropriate tradeoff between the size and integrity. With the increase in λ or γ, the policy to determine logical part whether is trivial or salient is getting more tolerant and more parts are determined as salient.

To choose the best λ and γ, we trained several pattern-trees with different parameters λ from 1 to 9 with step 1 and γ from 0.1 to 0.9 with step 0.1 on all GET method URLs in training dataset. As shown in Figure 4, it is obvious that with

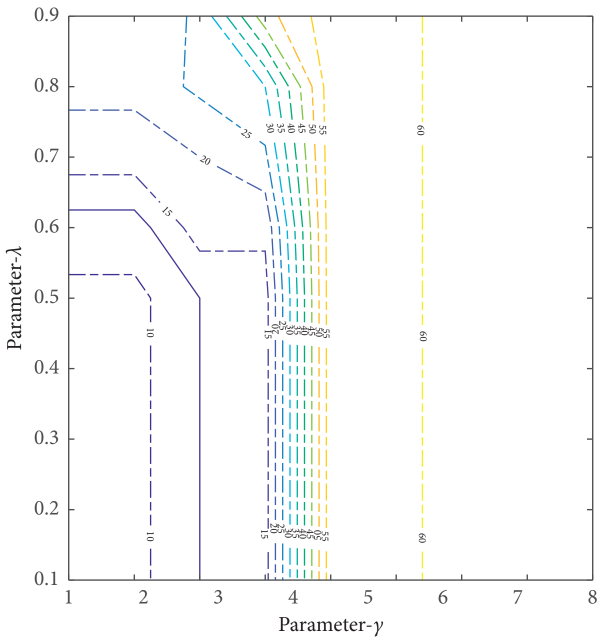
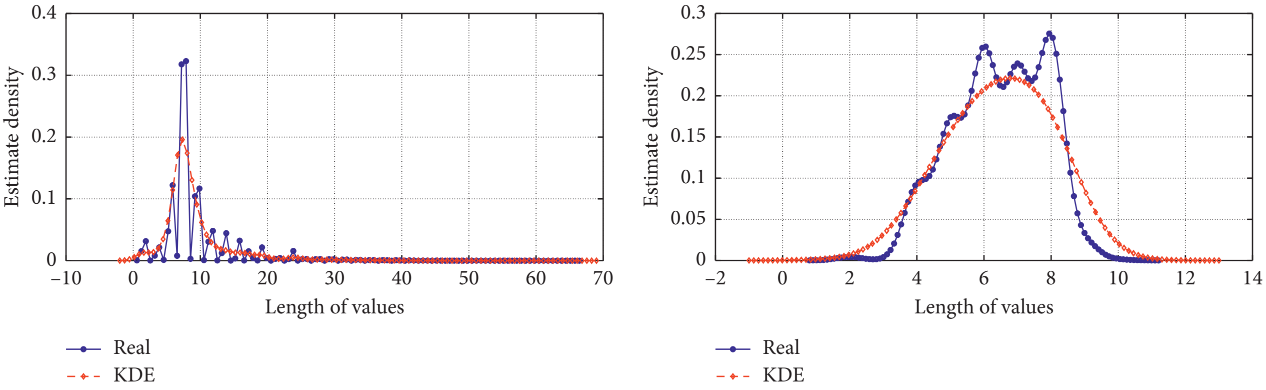


FIGURE 4: The number of resources recognized in training dataset on different parameters λ and γ.



(a)

(b)

FIGURE 5: The difference of kernel density estimate (KDE) of length between all values and values of an example trivial logical part. (a) The KDE of length for all request values. (b) The KDE of length for request values for an example trivial logical part. It is obvious that the length distribution using semantic structure is more regular and learnt.

the increasing of γ, the number of semantic structure pieces encoded in $\mathscr{T}$ is increasing rapidly. The cause of this phenomenon is that γ pays more significance than λ in controlling the tolerance of determining a logical part whether is trivial. The solid blue line in Figure 4 is the ground true number of resources in this Web application. In this paper, we chose hyperparameter λ as 0.3 and γ as 3.

To demonstrate how semantic structure helps to build a more precise anomaly-detection model, we compared the distribution of length feature which is separately extracted with and without using semantic structure. Length feature is a common feature to measure the length of value and is widely used in many anomaly detection types of research.

Figure 5 shows the comparison of these two distributions. The probability distribution and kernel density estimation (KDE) of the original length feature observed from all URLs are shown in Figure 5(a). In contrast, Figure 5(b) shows the probability distribution and KDE which are observed from an example logical part. It is obvious that the distribution shown in Figure 5(a) is more regular than Figure 5(b) and is further easy to be profiled by the anomaly-based detection model. This experiment shows that the semantic structure has significant point to improve the learning ability and accuracy of the detection model.

Finally, we further implemented an experiment to demonstrate that the semantic structure can extremely improve the performance of the detection model. We construct two types of models. One is using the conventional routine that directly extracts the features on the dataset using the feature proposed in [7] and trains the anomaly detection

```
Input: given a subgroup $U$ obtained by Section 5.1 and initialize $j$ as 1
Output: a tree node $\mathcal{N}$ for URLs in $U$
Create a new node $\mathcal{N}$ and extract parameter-value collection $kv$ for a random URL
if $j >$ the size of $kv$, then
    return the node $\mathcal{N}$
end if
extract $pv$ for each URL in $U$, and combine the value of $j^{\text{th}}$ parameter into collection $K$
calculate $H(K)$ of $K$
if $H(K) < \lambda \log V$ or $V < \gamma$, then
    $V^{*}$ = the set of distinct values in $K$
else
    $V^{*} = \{`*'\}$
end if
further split $U$ into several subgroups $\{U_1, \ldots, U_t\}$ according to $V^{*}$
for all subgroup $U_i$ do
    child = construct pattern − tree $(U_i, j+1)$
    add child as a child of node $\mathcal{N}$
end for
return the node $\mathcal{N}$
```

Algorithm 1: Construct pattern − tree (U, j).

Table 2: The performance comparison of difference models.

Algorithm	Precision	Recall	F1-score	Accuracy
Random forest	0.8899	0.7860	0.8348	0.8169
Random forest*	0.9154	0.8508	0.8819	0.8376
Decision tree	0.8646	0.8075	0.8351	0.8124
Decision tree*	0.9169	0.8454	0.8797	0.8352
Support vector machine	0.6746	0.9090	0.7745	0.6912
Support vector machine*	0.9636	0.8541	0.9056	0.8731
K-neighbours	0.8879	0.7764	0.8292	0.8109
K-neighbours*	0.9335	0.8491	0.8893	0.8493

The algorithms with * are trained with semantic structure, and others are trained with conventional routine.*

model. Other is trained within the semantic structure. The specific machine learning algorithms used in this experiment are random forest, decision tree, support vector machine, and K-neighbours. The hyperparameters of these models are not tuned but only used the default parameter value initialized in Scikit-learn.

Table 2 shows the comparison results. It is obvious that the performance of the detection model is briefly enhanced by using semantic structure. In random forest, decision tree, and K-neighbour-based detection model, the F1-score has considerable average 5% improvement. Especially in the support vector machine-based model, F1-score has dramatic 13% improvement. The significant improvements in precision, recall, F1-score, and accuracy in different machine learning algorithms strongly suppose the importance of semantic structure.

As highlight earlier, there exist three types of anomalies in CSIC-2010. Our anomaly detection model can efficiently perform detection than traditional models. In conventional scenarios, no matter static attacks, dynamic attacks, or unintentional attacks, the anomaly detection model has to inspect each value or character in the requesting URL. However, in our method, most of static and unintentional attacks can be detected by semantic structure because these URLs seriously violate the learned semantic structure (e.g., the value of salient logical part that has not observed in training dataset presents and there still exits pair that has not been inspected in *kv* when semantic structure tree has reached the bottom). Moreover, our method pays more attention to the values of vulnerable logical parts and builds a more precise detection model. Because our method detects little volume of URLs and has more precise model than conventional models, we achieve a significant lower false positive and higher accuracy.

9. Conclusion and Future Work

We introduced an enhanced feature extraction method for Web anomaly detection that uses semantic structure of request URLs. We propose to use skeleton structure to eliminate the request-imbalance problem. By using semantic structure, the detection model is able to pay more attention to the vulnerable logical parts and produces a precise model.

The feature distribution comparison demonstrates the reason why the semantic structure can help to improve the performance of the detection model. And, the improvement of performance shown in Table 2 also indicates the value of semantic structure.

We plan to study how to learn the nonstationary semantic structure with an increment learning mechanism. To provide a better service for users, Web application is constantly evolved, such as adding new or removing old resources and changing the parameters of some resources.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The previous version of this research work was published at the 6th International Symposium, SocialSec 2020. The content in this new version has been extended by more than 30%.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by a grant from the National Natural Science Foundation of China (nos. 61941114 and 62001055) and Beijing Natural Science Foundation (no. 4204107).

References

[1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, "Web services," in *Web Services*, pp. 123–149, Springer, Berlin, Germany, 2004.

[2] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: a review," *Computers & Security*, vol. 30, no. 6-7, pp. 353–375, 2011.

[3] W. Yurcik, J. Barlow, and J. Rosendale, "Maintaining perspective on who is the enemy in the security systems administration of computer networks," in *ACM CHI Workshop on System Administrators Are Users*, Fort Lauderdale, FL, USA, April 2003.

[4] D. M. Hawkins, *Identification of Outliers*, Vol. 11, Chapman and Hall, London, UK, 1980.

[5] V. Jyothsna, V. Rama Prasad, and K. Munivara Prasad, "A review of anomaly based intrusion detection systems," *International Journal of Computer Applications*, vol. 28, no. 7, pp. 26–35, 2011.

[6] R. Funk and N. Epp, "Anomaly-based web application firewall using http-specific features and one-class svm," *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação*, vol. 2, p. 1, 2018.

[7] B. Cui, S. He, X. Yao, and P. Shi, "Malicious URL detection with feature extraction based on machine learning," *International Journal of High Performance Computing and Networking*, vol. 12, no. 2, pp. 166–178, 2018.

[8] R. Pal and N. Chowdary, "Statistical profiling of n-grams for payload based anomaly detection for HTTP web traffic," in *Proceedings of the 2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Indore, India, December 2018.

[9] A. M. Vartouni, S. Kashi, and M. Teshnehlab, "An anomaly detection method to detect web attacks using stacked autoencoder," in *Proceedings 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, Kerman, Iran, March 2018.

[10] W. Khreich, B. Khosravifar, A. Hamou-Lhadj, and C. Talhi, "An anomaly detection system based on variable N-gram features and one-class SVM," *Information and Software Technology*, vol. 91, pp. 186–197, 2017.

[11] M. Zolotukhin, T. Hämäläinen, T. Kokkonen, and J. Siltanen, "Analysis of HTTP requests for anomaly detection of web attacks," in *Proceedings 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*, Dalian, China, August 2014.

[12] T. Lei, "A pattern tree-based approach to learning URL normalization rules," in *Proceedings of the 19th International Conference on World Wide Web*, 2010.

[13] Z. Cheng, B. Cui, and J. Fu, "A novel web anomaly detection approach based on semantic structure," in *International Symposium on Security and Privacy in Social Networks and Big Data*, pp. 20–33, Tianjin, China, August 2020.

[14] Giménez, C. Torrano, A. P. Villegas, and G. Álvarez Marañón, "HTTP data set CSIC 2010," Information Security Institute of CSIC, Spanish Research National Council, Madrid, Spain, 2010, https://www.tic.itefi.csic.es/dataset/.

[15] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.

[16] Q. Niu and X. Li, "A high-performance web attack detection method based on CNN-GRU model," in *Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 1, pp. 804–808, Chongqing, China, June 2020.

[17] P. Tang, W. Qiu, Z. Huang, H. Lian, and G. Liu, "Sql injection behavior mining based deep learning," in *Proceedings of the International Conference on Advanced Data Mining and Applications*, Nanjing, China, November 2018.

[18] H. T. Nguyen, C. Torrano-Gimenez, G. Alvarez, S. Petrović, and K. Franke, "Application of the generic feature selection measure in detection of web attacks," *Computational Intelligence in Security for Information Systems*, pp. 25–32, Springer, Berlin, Germany, 2011.

[19] C. Torrano-Gimenez, H. T. Nguyen, G. Alvarez, and K. Franke, "Combining expert knowledge with automatic feature extraction for reliable web attack detection," *Security and Communication Networks*, vol. 8, no. 16, pp. 2750–2767, 2015.

[20] Y. Zhou and P. Wang, "An ensemble learning approach for XSS attack detection with domain knowledge and threat intelligence," *Computers & Security*, vol. 82, pp. 261–269, 2019.

[21] B. A. Tama, L. Nkenyereye, S. M. R. Islam, and K.-S. Kwak, "An enhanced anomaly detection in web traffic using a stack of classifier ensemble," *IEEE Access*, vol. 8, pp. 24120–24134, 2020.

[22] C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, Washington D.C. USA, October 2003.

[23] C. Kruegel, G. Vigna, and W. Robertson, "A multi-model approach to the detection of web-based attacks," *Computer Networks*, vol. 48, no. 5, pp. 717–738, 2005.

[24] S. Cho and S. Cha, "SAD: web session anomaly detection based on parameter estimation," *Computers & Security*, vol. 23, no. 4, pp. 312–319, 2004.

[25] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Proceedings of the International Workshop on Recent Advances in Intrusion Detection*, Sophia Antipolis, France, September 2004.

[26] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A novel web attack detection system for internet of things via ensemble classification," *IEEE Transactions on Industrial Informatics*, 2020.

[27] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020.

[28] J. Wu, Z. Yang, L. Guo, Y. Li, and W. Liu, "Convolutional neural network with character embeddings for malicious web request detection," in *Proceedings of the 2019 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking*, pp. 622–627, 2019.

[29] A. Tekerek, "A novel architecture for web-based attack detection using convolutional neural network," *Computers & Security*, vol. 100, p. 102096, 2021.

[30] T. Lei, R. Cai, J.-M. Yang, Y. Ke, X. Fan, and L. Zhang, "A pattern tree-based approach to learning URL normalization rules," in *Proceedings of the 19th International Conference on World Wide Web*, Raleigh, North Carolina, USA, April 2010.

[31] Y. Yang, "UPCA: an efficient URL-pattern based algorithm for accurate web page classification," in *Proceedings of the 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Zhangjiajie, China, August 2015.

[32] R. Fielding, J. Gettys, J. Mogul et al., *Hypertext transfer protocol–HTTP/1.1*, http://www.hjp.at/doc/rfc/rfc2616.html, 1999.

[33] J. Shi, Y. Cao, and X.-J. Zhao, "Research on SEO strategies of university journal websites," in *Proceedings of the 2nd International Conference on Information Science and Engineering*, Hangzhou, China, December 2010.

[34] M. Cui and S. Hu, "Search engine optimization research for website promotion," in *Proceedings of the 2011 International Conference of Information Technology, Computer Engineering and Management Sciences*, vol. 4, Nanjing, China, 2011.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6656470, 12 pages
https://doi.org/10.1155/2021/6656470

Research Article

Zombie Follower Recognition Based on Industrial Chain Feature Analysis

Juan Tang,[1] **Hualu Xu**,[2] **Pengsen Cheng**,[2] **Jiayong Liu**,[2] **Cheng Huang**,[2] **and Xun Tang**[2]

[1]*College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China*
[2]*School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China*

Correspondence should be addressed to Jiayong Liu; ljy@scu.edu.cn

Received 20 November 2020; Revised 3 January 2021; Accepted 25 January 2021; Published 8 February 2021

Academic Editor: Zhe-Li Liu

Zombie followers, a type of bot, are longstanding entities in Sina Weibo. Although the features and detection of zombie followers have been extensively studied, zombie followers are continuously increasing in social networks and gradually developing into a large-scale industry. In this study, we analyze the features of eight groups of zombie followers from different companies. The findings indicate that although zombie followers controlled by different companies vary greatly, some industries may be controlled by the same organization. Based on the feature analysis, we use multiple machine learning methods to detect zombie followers, and the results show that zombie follower groups with short registration time are more easily detected. The detection accuracy of zombie followers that have been cultivated for a long duration is low. Moreover, the richer the feature sets, the higher the recall, precision, and F_1 of their detection results will be. Under a given rich feature set, the accuracy of the combined-group detection is not as high as that of the single-group detection. The random forest achieves the highest accuracy in both single- and combined-group detections, yielding 99.14% accuracy in the latter case.

1. Introduction

Sina Weibo is an online social network service, such as Twitter and Facebook, and has nearly 516 million monthly active users till December 2019 according to the fourth quarter financial report of Sina Weibo [1]. Similar to the case of other social media services, many misbehaving accounts [2] exist in Sina Weibo, such as bots [3–5], trolls [6–8], sockpuppets [9, 10], and compromised accounts [11, 12]. The ultimate aim of such accounts that participate in social networks is to cause disruption of the normal order.

Zombie followers [13–15], a type of bot [3, 4, 16], are longstanding entities in Sina Weibo. They are often used to spread malicious information, manipulate public opinion, steal personal information, and so on [4, 17–19]. They not only undermine users' social credibility but also adversely affect users' network security and social environment [15, 20, 21]. Researchers in the past have often focused on the detection of zombie followers [15, 22]. They analyzed the feature differences between zombie followers and normal users, such as text features [23, 24], behavior features [25, 26], or network structure features [27–29], and then combined machine learning methods for zombie follower detection [30, 31].

Although the features and detection of zombie followers have been extensively studied, zombie followers are continuously increasing in online social networks and gradually developing into an industry [26, 32]. We observed that zombie followers on Sina Weibo are gradually moving toward this trend and forming a large-scale ecosystem, wherein the user can get many zombie followers for a small cost. Previous studies have not analyzed the characteristics of different zombie follower industries. Questions such as will there be differences in the characteristics of zombie followers from different sources and in the detection results if the same detection method is used for zombie followers with different characteristics have not been explored. Therefore, the study of zombie followers' ecosystem and industry features will help us better understand and automatically detect them.

Present work. In this paper, we focus on the feature analysis and detection of eight zombie follower groups. Herein, zombie followers [15] are defined as malicious users that are manipulated and maintained by programs. They imitate human behaviors and influence normal social behaviors on social networks. The zombie follower industry [33] is defined as a new black market formed by merchants engaged in the production and sale of zombie followers. To analyze the characteristics of the zombie follower industry, we investigate various organizations engaged in the trading of zombie followers on the Internet. Here, an organization or company that provides sales of zombie followers is termed a zombie follower company (hereafter, ZF company).

We collected eight zombie follower groups (each group having more than 5,000 accounts) from different ZF companies. Based on the collected data, our paper provides the following three main contributions:

(1) We analyzed the basic features and content features of zombie follower groups and found that zombie follower companies always mass produce zombie followers. Due to varying registration time and service scope, those zombie followers usually have different features.

(2) We study the interactive relationship between the eight zombie follower groups. The findings indicate that some of the merchants selling zombie followers are actually operated by the same organization.

(3) Finally, based on the study of the aforementioned features of the zombie follower industry, we use machine learning methods to detect the zombie followers in a single group and in combination. In the single-group detection, zombie followers with short registration time are more easily detected. The detection accuracy of zombie followers that have been cultivated for a long time declines. Moreover, the richness of the feature set plays an important role in the detection. The richer the feature sets, the higher the recall, precision, and F_1 values will be. Although the accuracy of the combined-group detection is not as high as that of the single-group detection, the random forest is the highest in both detections, with 99.14% accuracy in the combined-group detection.

2. Data Specification

This section details the source of our dataset as well as the settings of honeypot accounts and the data crawling process.

2.1. Data Source. On Weibo, the number of followers of users often depends on the users' influence. Driven by the benefits of pan-entertainment and commercialization, users' demand for zombie followers has grown, leading to a large-scale purchase of zombie followers in the market. Based on the investigation of the various advertisements on the Internet to sell zombie follower services, the following main sales channels can be found:

Weibo profile: some zombie followers leave sales advertisements with contact information on normal users' microblogs, while others mark such information on their avatars and spread it by following normal users.

Taobao shop: Taobao, a popular C2C platform in China, has taken all efforts to stop illegal sales, but a search using specific keywords can still lead one to the sales of zombie follower services. The stores offer various packages and for each package, the basic information, quantity, and price of zombie followers are explained on the product details page. Buyers can place orders directly according to the instructions.

Search engine: when searching for keywords related to the sales of zombie follower services on major search engines, a series of related websites will appear, which contain information such as the categories and number of zombie followers. Buyers can purchase directly from the website or contact the customer service staff according to the information provided on the website.

2.2. Honeypot Account Description. Honeypot is a common means to collect zombie followers [32, 34]. On Facebook, MySpace, and Twitter, it is often used to detect spammers [35–37]. Aiming at studying the current ecosystem of zombie followers, we created eight honeypot accounts on Weibo, corresponding to the eight ZF company groups.

All honeypots remain in the initial state and empty (i.e., no basic information or microblog is present). We collected more than 5,000 zombie followers each from eight different companies and injected them into the corresponding honeypot account. Table 1 lists honeypot account details and the sources of zombie followers. All zombie followers were collected at the same time. Overall, we collected a total of 43,352 zombie followers in eight groups.

2.3. Data Collection. What are the characteristics of the zombie follower industry? What are the characteristic differences between zombie followers and normal users? How to detect zombie followers? To answer these questions, in this study, we mainly collected two datasets: (1) the zombie followers' data and related data collected through the honeypot account and (2) the normal users' data and related data collected through the Python crawler. All the above data were exclusively open data obtained through Sina API. In addition, we encrypted the data to ensure data security. The collection of the datasets and the detailed analysis of the basic characteristics are described below.

We used the Python crawler to monitor the corresponding honeypot account and detect the injected zombie followers. After all zombie followers were collected, we performed a second crawl on the collected data. The public information, followers (the latest 1000), and microblogs (the latest 100) of each zombie follower were crawled, and the results are presented in Table 2. Meanwhile, we collected the data, including public information and 3,394,129 microblogs

Table 1: Summary statistics of honeypot accounts.

User ID	Campaign name	Provider/source	#Zombie followers
01_72****37	A000 Douyin and Weibo flagship shop	Weibo profile	5069
02_72****04	Sihui Network	Taobao shop	5270
03_72****91	Xingchen Network Technology	Search engine	6172
04_72****62	Aijia Network	Search engine	5258
05_72****21	Self-help business platforms for Douyin, Kuaishou, and Weibo	Search engine	6313
06_72****43	A Weibo-Douyin-WeChat platform	Weibo profile	5091
07_72****47	Niuweifen marketing	Taobao shop	5073
08_72****28	Yunyidingdian platform	Weibo profile	5106

Table 2: Summary statistics of the eight honeypot accounts.

User ID	#Followers	#Followers obtained	#F_followers[1]	#Microblogs
01_72****37	5,069	5,063	661	20,528
02_72****04	5,270	943	21	3
03_72****91	6,172	6,145	227,101	285,756
04_72****62	5,258	5,257	79	—
05_72****21	6,313	5,000	80	—
06_72****43	5,091	5,022	1,081,774	642,679
07_72****47	5,073	5,072	2,550	0
08_72****28	5,106	5,105	1,186	18,584

[1] F_followers are the followers of the zombie followers in the eight zombie follower groups.

(the latest 100) of 45,559 normal users as a comparison dataset.

Each ZF company assures us that their products are authentic and reliable. Their zombie followers have avatars, personal information, and irregular updates of microblogs. More importantly, they cannot be blocked. However, the zombie followers of some ZF companies were blocked by Sina Weibo within a short time. For example, Groups 02, 04, and 05 were blocked shortly after the infusion was completed. Among them, 943 zombie followers belonging to Group 02 were unblocked after some time, but they were soon blocked again. Groups 04 and 05 were banned in the early period, so some of their data were missing (only Weibo ID, user name, and the number of followers and friends could be collected). In addition, we found that a small number of zombie followers were blocked in other groups. Finally, we obtained a total of 37,607 zombie followers, having 1,313,452 followers and 967,550 microblogs.

3. Analysis of Characteristics

In previous studies, zombie followers and normal users were usually distinguished from various perspectives [15], such as users' personal information [13, 14], relationship features [28], behavioral features [26], and emotional features. In this section, we attempt to answer the following two questions on the basis of basic features and content features: what is the difference between zombie follower groups and what is the difference between zombie follower groups and the normal user group?

3.1. Basic Characteristics. We randomly selected 5000 users from the normal user group as Group 09. For comparison, we also calculated the average value of Groups 01–08 and considered it as Group 00. In this section, we combine existing fields in the dataset and compare the groups in terms of five aspects: registration time, the number of followers and friends of users, username complexity, and user hierarchy.

3.1.1. More Centralized Registration Time. Figure 1 shows the cumulative distribution function (CDF) graphs of registration time of users in all the groups. It indicates that the registration time of the normal user group (Group 09) is evenly distributed. However, the CDF graphs of the zombie follower groups (data in Groups 02, 04, and 05 were missing) are significantly different from that of Group 09. The distribution graphs of Groups 01 and 08 are similar, with a stepped increase and high consistency in value. Groups 03 and 06 are also similar, and their graphs are closer to that of Group 09. However, compared with the graph of Group 09, graphs of Groups 03 and 06 are not smooth and show a stepped increase. Their distribution is similar to that of Groups 01 and 08. All the zombie followers in Group 07 were registered three days before purchase, so its CDF is more concentrated. As Figure 1 shows, most of the zombie followers were produced recently. Therefore, we can infer that most ZF companies continue to mass produce zombie followers.

In summary, zombie follower groups manipulated by different ZF companies have significant differences in registration time. Some ZF companies hold and mass produce zombie followers close to the purchase time, while others mass produce them in advance. As a comprehensive CDF graph of zombie followers, the curve of Group 00 indicates that the zombie follower industry is developing on a large scale.

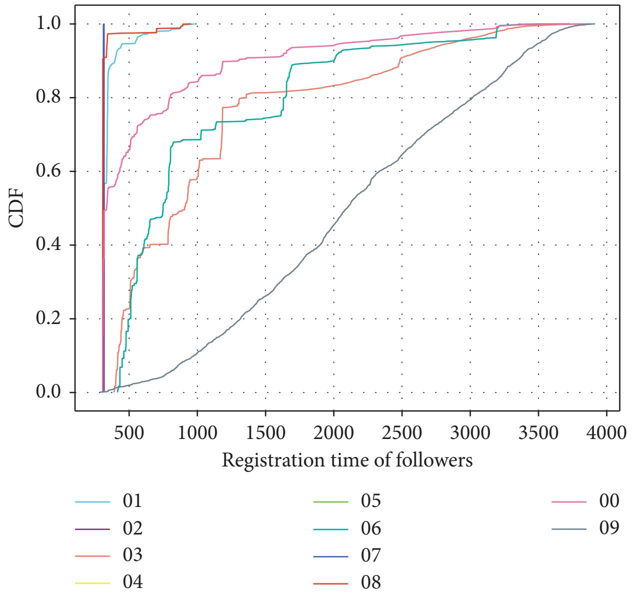


Figure 1: Cumulative distribution of registration time of users in different groups (00 is the average registration time of zombie follower groups; 09 is the registration time of the normal user group).

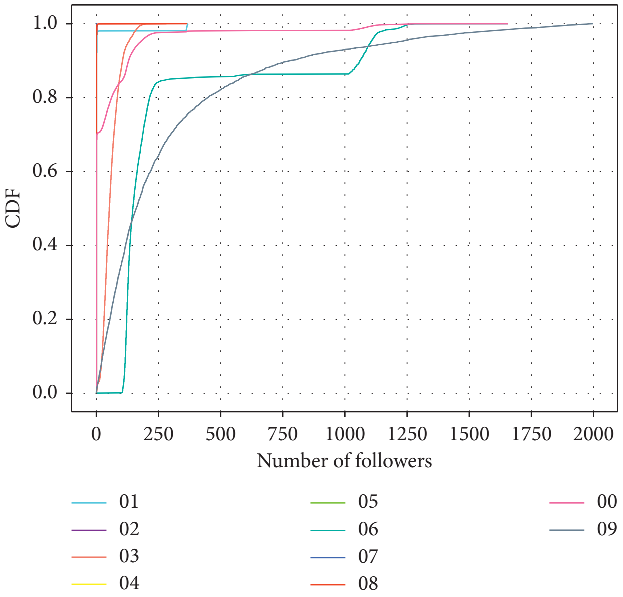


Figure 2: Cumulative distribution of the number of user followers in different groups.

3.1.2. Fewer Followers and Mutually Following. In Figure 2, the median number of user followers in Group 09 is 186, whereas that of Groups 01–08 is 29, indicating a large difference between them. Most zombie followers of Groups 04, 05, and 07 have only one to three followers. The distribution of Groups 01 and 08 is similar, and most of their users have no followers. The followers in Group 03 are relatively dispersed, evenly distributed between 0 and 250, while those in Group 06 are almost all over 100 (only two users have less than 100).

Based on the above results, compared with Group 09, the number of user followers in Groups 01–08 is more consistent. A cursory investigation reveals that to make zombie followers resemble normal users, zombie followers in Groups 01–08 generally follow each other, thus forming a network of zombie followers.

3.1.3. Prefer to Be a Follower Based on the Service. As shown in Figure 3, the distribution of the number of friends of users in Group 09 is even, with about 80% being less than 500. By contrast, the distribution of Groups 01–08 is irregular and most of the zombie followers have more friends than normal users have, such as Groups 03 and 06. The registration time distribution for Groups 03 and 06 suggests that they have been engaged in selling follower services for a long duration, so their users have more friends. Furthermore, most users in Groups 01 and 08 have less than 200 friends. In Groups 04, 05, and 07, all have fewer than 200 friends, and their CDF graphs show an irregular stepped increase (Figure 4). Based on the registration time distribution, we believe that most zombie followers in the five groups have not been engaged in the business for a long duration, so they have not accumulated a large number of friends.

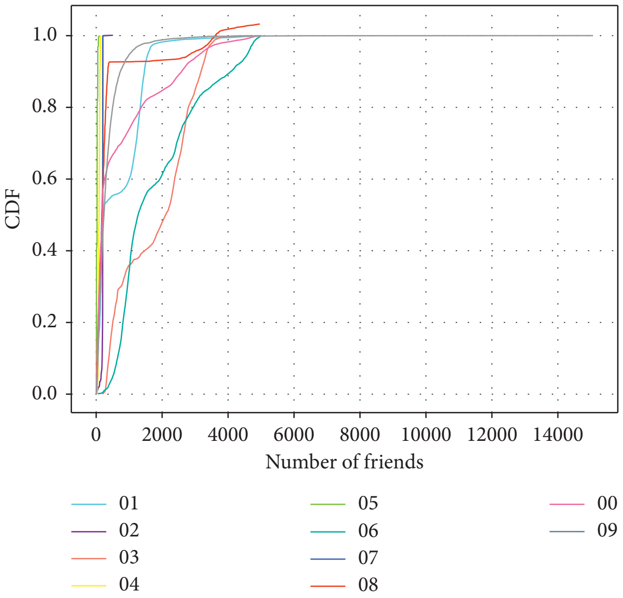


Figure 3: Cumulative distribution of the number of friends in different groups.

To better reflect the composition of users' social relations, we use the interaction index function [38], defined as

$$\text{interaction index} = \frac{\text{followers count}}{\text{friends count}}. \tag{1}$$

Figure 5 shows the CDF graphs of the interaction index for all groups. In Group 09, the interaction index of 96.76% of the users is less than 10, and the maximum index is 81. In Groups 01–08, the interaction index of only 16.09% is less than 10, while that of 26.12% is greater than 81.

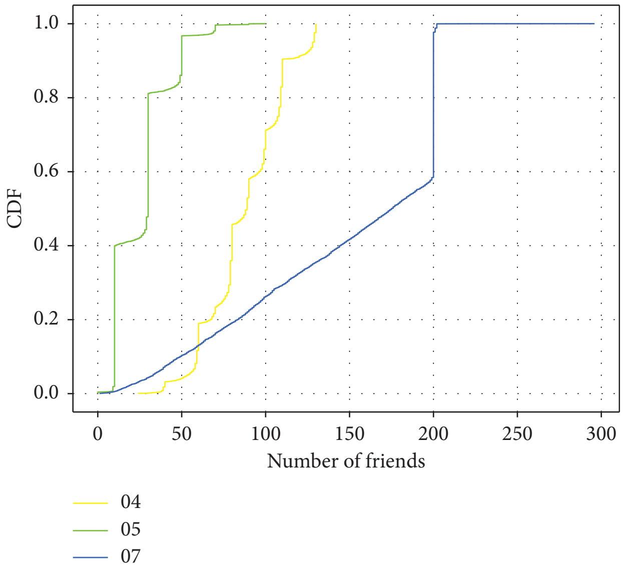


Figure 4: Cumulative distribution of the number of friends of users in Groups 04, 05, and 07; an irregular ladder increase in the number is observed.

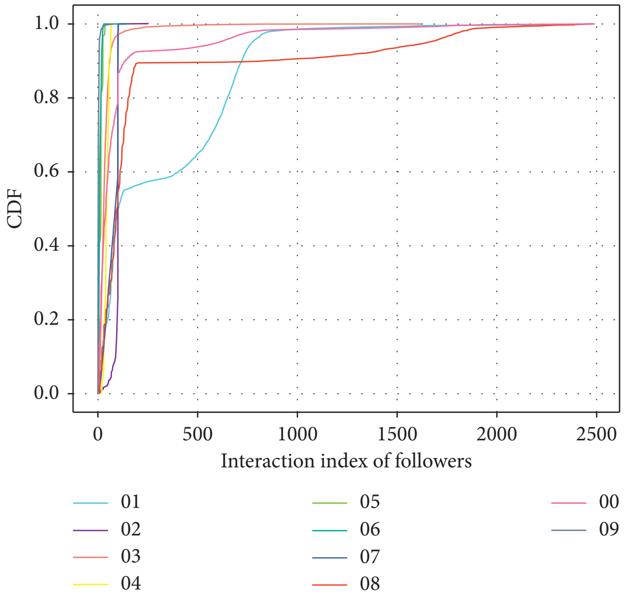


Figure 5: Cumulative distribution of the interaction index for different groups.

3.1.4. Simpler and Meaningless Usernames. We next analyze the username complexity of Groups 01–09 by using the Jieba [39] algorithm to segment the usernames. Accordingly, if n is the number of words in the username, K is the number of numerals, and len_i is the length of the i-th word, then the complexity of the username [40] is given as

$$\text{complex} = n + \sum_{i=1}^{k} \frac{\text{len}_i}{3}. \quad (2)$$

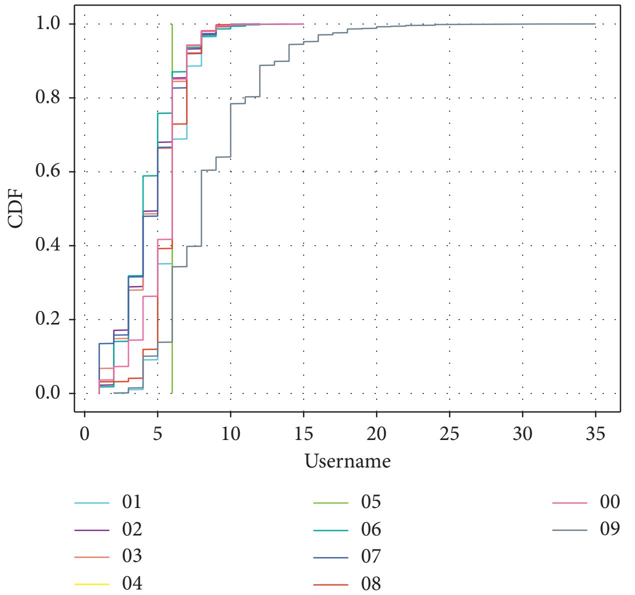


Figure 6: Cumulative distribution of the username complexity in different groups.

The username complexity of Groups 01–09 is shown in Figure 6. The figure indicates that the username complexity of Group 09 is greater than that of Groups 01–08. The analysis of the composition of usernames in Groups 01–08 reveals that the usernames of Groups 04 and 05 are automatically generated by the system, with their structure being "user" + random number. Compared with other groups, Groups 01 and 08 usernames are more readable and have specific rules for their generation. Most usernames in Groups 02, 03, 06, and 07 are random combinations of Chinese characters and letters, bearing no specific meaning.

3.1.5. Lower Level in Weibo's Hierarchy. Our calculation of the hierarchies of users in Groups 01–09 shows that out of the 43,352 zombie followers, only 1,278 have a user hierarchy greater than 0, accounting for 4.84%. However, in Group 09, 2945 users are greater than 0, accounting for 58.9%. Due to the mass production of zombie followers, ZF companies cannot improve the hierarchies for most zombie followers. However, it is worth noting that ZF companies not only sell zombie follower services but also control mass social robots with advanced authentication and higher hierarchies than normal users. In the future, we will conduct research considering this aspect.

3.2. Content Characteristics. This section presents a comparison of the relevant features of users' microblogs in Groups 01–09. We could not obtain the microblog data of Groups 04 and 05 because the zombie followers in these groups were blocked. Moreover, 5072 zombie followers in Group 07 did not post any microblog as their registration time was shorter, and 943 zombie followers in Group 02 only posted three microblogs. Therefore, we were focusing on the content of only Groups 01, 03, 06, 08, and 09.

3.2.1. Replication of Original Content. To determine what content zombie followers often post, we analyzed the microblog content of Groups 01–09. As Table 3 shows, every microblog in Groups 01 and 08 is a repost, and mostly the same posts. Conversely, Groups 03 and 06 are more balanced, including original microblogs and reposts. Among them, most of the original microblogs repeat celebrity quotes or common senses. The repetition rate of reposts of zombie followers is generally higher than that of normal users. Moreover, in Groups 03 and 06, the rate is 50% or more. Thus, we can infer that the zombie followers manipulated by ZF companies hardly post original microblogs and their reposts are related to their business. Therefore, ZF companies are suspected of manipulating public opinion.

3.2.2. Low Interaction of Microblogs. By analyzing the content of microblogs of Groups 01–09, we investigated if zombie followers write differently from normal users. As Table 4 shows, compared to Group 09, zombie follower groups contain fewer URLs, mentions (@), and hashtags, and they are less interactive with other users. In addition, although the length of microblogs of different groups is different, groups with similar basic characteristics have similar length of microblogs.

3.2.3. Poor Microblog Sources. Regarding the microblog sources, Table 5 shows that Group 09 has 3,650 sources, accounting for 94.32% of the total (3,870), whereas the sources of zombie follower groups are considerably less. Among them, Groups 01 and 08 have only three consistent sources, whereas Group 06 has the most abundant sources (only 429).

3.2.4. Poor Spreading of Microblogs. The communication features [26] of the microblogs of Groups 01–09 are analyzed in Table 6. As the table indicates, the zombie follower groups are significantly different from Group 09 as more than 98% of the zombie followers have zero reposts, attitudes, and comments, and almost no group has a count above 10. We conclude that although zombie followers do post microblogs, they usually get little attention from other users; thus, the posts have poor ability to spread.

3.3. Discussion. Zombie follower groups have different features because of varying registration time. Due to longer survival time, Groups 03 and 06 have greater similarity to normal users. However, the registration time of Group 07 is only three days, so all its features are notably different. It can be expected that some ZF companies have been engaging in cultivating zombie followers for a long duration, and the longer they manipulate, the more similar the zombie followers will be to normal users.

Comparison of the features of zombie follower groups reveals an interesting phenomenon. The features of Groups 04 and 05, Groups 01 and 08, and Groups 03 and 06 are very similar. In the next section, we will analyze whether these zombie follower groups are correlated.

4. Ecosystem Characteristics

We focus here on the following three questions: why is it difficult for normal users to identify zombie followers? Are different ZF companies correlated? Are there social relations among zombie follower groups?

4.1. Why Is It Difficult for Normal Users to Identify Zombie Followers?. From the features of zombie follower groups described in Section 3, we can conclude that zombie followers are considerably different from normal users. However, it is often difficult for normal users to judge whether a user is a zombie follower. Note that when normal users visit others' profiles, they usually judge the profile authenticity by observing its basic information and microblogs.

We analyzed the differences in the basic information between Groups 01–08 and Group 09 (Table 7). Compared with normal users, zombie followers (95.83%) have more complete basic information. To avoid blocking of the zombie followers, ZF companies make them behave more like humans. For example, Groups 03 and 06 not only have extremely complete basic information (99.86%) but also have rich original microblogs (Table 4). Clearly, zombie followers also have real avatars, complete basic information, and simulated original microblogs and reposts. Therefore, the boundary between the zombie followers and normal users becomes increasingly blurred, making distinction of zombie followers difficult.

4.2. Are Different ZF Companies Correlated?. We analyzed the relationship among zombie follower groups to determine if ZF companies are correlated, if these companies manipulate different zombie follower groups with different sales methods and if these companies belong to the same organization.

By observing the interaction of each user from Groups 01–08, we determine if any zombie follower is present in two or more groups simultaneously. After matching, we found 8 identical zombie followers in Groups 01 and 08 and 104 in Groups 04 and 05 (Figure 7). Considering the characteristics of these groups, it is reasonable to conclude that they are controlled by the same organization.

4.3. Are There Social Relations among Zombie Follower Groups? After determining that there may be some correlation among zombie follower groups, we attempt to determine whether there also have some correlating social relations.

The analysis of the followers of zombie followers indicates that most of them are zombie followers. Therefore, we compared the followers of the zombie followers in Groups 01–08. The results demonstrate a small amount of overlap between some groups (Table 8). However, the number of overlaps in Groups 03 and 06 is as high as 5,696. After removing the repeated followers, we obtained 218 followers of the zombie followers in Groups 03 and 06 multiple times.

Table 3: Summary statistics of microblog content.

Collection	Original count	Repost count	Repetition	Repetition rate (%)
01	0	20,528 (100%)[1]	2,552	31.41
03	252,690 (88.43%)	33,066 (11.57%)	5,493	65.41
06	312,630 (48.64%)	330,049 (51.36%)	44,652	55.39
08	0	18,584 (100%)	2,400	31.46
09	145,768 (37.05%)	247,720 (62.95%)	23,506	23.73

[1] Parenthetical information is the percentage of the total number of microblogs in each group.

Table 4: Summary statistics of the microblog content features.

Collection	Status count	URLs	Mentions	Hashtags	Ave-length
01	20,528	76 (0.37%)[1]	996 (4.85%)	0	12.42
03	285,756	10,620 (3.72%)	6,215 (2.17%)	19,225 (6.73%)	141.99
06	642,679	9,821 (1.53%)	66,379 (10.33%)	66,534 (10.35%)	65.94
08	18,584	60 (0.32%)	751 (4.04%)	0	11.70
09	393,488	85949 (21.84%)	76,682 (19.49%)	202,010 (51.34%)	55.56

[1] Parenthetical information is the percentage of the total number of microblogs in each group.

Table 5: Summary statistics of the microblog sources.

Collection	01	03	06	08	09
Count	3 (0.08%)	429 (11.09%)	217 (5.61%)	3 (0.08%)	3,650 (94.32%)

Table 6: Summary statistics of the microblog communication features.

Collection	Range	Reposts (%)	Attitudes (%)	Comments
01	0	99.58	98.82	99.92%
	0+	0.42	0.18	0.08%
03	0	99.79	97.07	99.47%
	1–10	0.19	2.81	0.51%
	10+	0.02	0.12	0.02%
06	0	99.61	98.12	99.61%
	1–10	0.38	1.87	0.39%
	10+	0.01	0.01	0
08	0	99.73	98.99	99.88%
	1–10	0.27	1.01	0.12%
09	0	92.27	71.72	82.11
	1–10	5.60	23.31	14.03%
	10+	2.13	4.97	3.86%

Although there are no identical zombie followers in Groups 03 and 06, their potential social relations suggest that they may belong to the same organization.

5. Detection Model

5.1. Experimental Features. As described in the above sections, we studied the behavior of zombie followers from the following three aspects and list the relevant features in Table 9: (1) basic features, which include the complexity of the user name, the number of followers and friends of users, the interaction index, hierarchy, and registration time of users. (2) Content features, including the rate of original microblogs and reposts, the total number of microblogs of users, URLs, hashtags, and mentions, and the average length of all microblogs. (3) Ecosystem features, it refers to the user's gender and age and whether the basic information of the user is provided.

5.2. Experimental Design. In this study, Python was used to realize the whole process of feature extraction and model construction, as shown in Figure 8. This model mainly consists of two parts: the data feature analysis module and zombie follower detection module. In the data feature analysis module, we obtain the original data set through the crawler, remove the invalid data after preprocessing, and format them. We then analyze the data features and transform them into the corresponding feature set. In the zombie follower detection module, we use five-fold cross-validation. The detailed data distribution is shown in Table 10. Finally, three machine learning methods (KNN [41], SVM [42], and random forest [43]) are used to detect the

Table 7: Summary statistics of the basic info of zombie followers.

Collection	Avatar	Location (%)	Gender (%)	Age (%)	Simple info (%)
01	No	77.07	77.64	40.61	77.68
02	Yes[1]	99.26	100	93.96	100
03	Yes	80.47	98.34	51.91	98.34
06	No	84.73	99.88	9.66	99.88
07	Yes	98.68	99.82	93.24	99.82
08	No	88.18	99.24	51.23	99.24
09	Yes	54.36	77.56	38.72	77.56

[1] "Yes" indicates that most of the zombie followers have avatars.

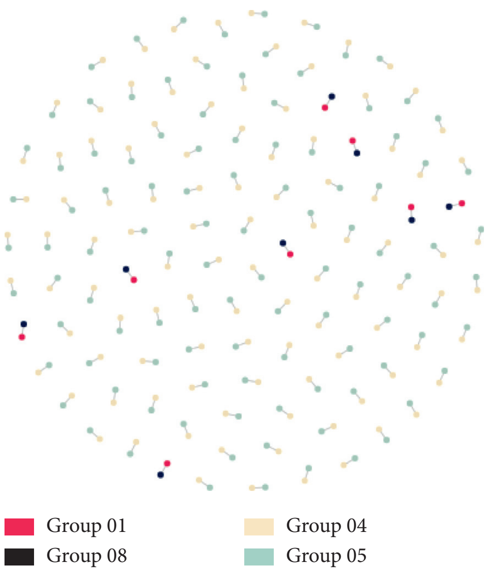


Figure 7: Matching of identical zombie followers in Groups 01–08.

Table 8: Summary statistics of the overlap of the followers of zombie followers.

Group	Num
01 and 03	10
01 and 06	20
01 and 08	18
03 and 04	2
03 and 05	2
03 and 06	5696
03 and 07	6
03 and 08	7
04 and 05	8
06 and 07	90
06 and 08	6
01, 03, and 06[1]	4
01, 03, and 08	2
03, 04, and 05	2
01, 03, 06, and 08	1

[1] Only the overlapped groups were output.

experimental data. Each of the three algorithms uses default parameters, and the calculation is executed on a computer with Intel(R) Xeon(R) CPU and 8 GB memory.

We evaluated the performance of the proposed model by using the indicators accuracy, recall, precision, and F_1. They are defined as follows:

$$\begin{aligned}\text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FT} + \text{FN}},\\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}},\\ \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}},\\ F_1 &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},\end{aligned} \tag{3}$$

where TP (true positive) is the number of normal users predicted as normal users; FP (false positive) is the number of zombie followers predicted as normal users; TN (true negative) is the number of zombie followers predicted as zombie followers; and FN (false negative) is the number of normal users predicted as zombie followers.

5.3. Experimental Results. In our experiments, we attempted to maximize the detection of zombie follower groups. We extracted the features of each user described in the previous sections and used three types of classifiers (i.e., KNN, SVM, and random forest) to detect eight zombie follower groups. Here, when the detection method uses any set of data from 01 to 08 zombie follower data and normal user data, it is called the single-group detection, and when it uses eight sets of zombie user data and normal user data, it is called the combined-group detection. Meanwhile, we also detected basic features, content features, and ecosystem features in single-group and combined-group detections, and the results are shown in Tables 11–13.

In the single-group detection, when part of data is missing (Groups 04 and 05 lack in content features and ecosystem features; Groups 02 and 07 lack in content

Table 9: Description of feature classifications.

Feature set	Features
Basic features	Username complexity, number of followers and friends, interaction index, user hierarchy, user registration time
Content features	Number of original post, repost, and status; number of URLs, hashtags, mentions, average length of microblogs (Avg-length)
Ecosystem features	Gender, age, simple info

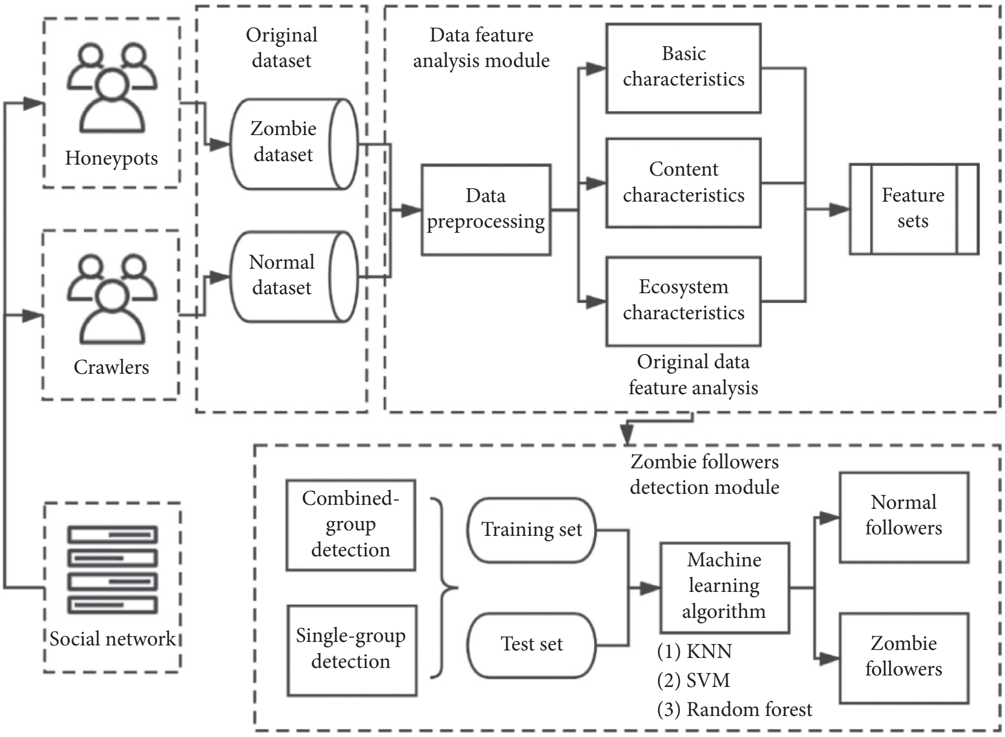


Figure 8: Structure of the detection model.

Table 10: Statistics of comparative experiment datasets.

Collection	Total
01_72****37	5063
02_72****04	943
03_72****91	6145
04_72****62	5257
05_72****21	5000
06_72****43	5022
07_72****47	5072
08_72****28	5105
Normal users	45559

features), although its detection accuracy was high, its recall, precision, and F_1 values were still generally low. In random forest detection, as an example, the accuracy of Groups 04 and 08 was 99.86% and 99.90%, respectively, but the recall, precision, and F_1 were 85.71%, 78.95%, and 82.19% for Group 04 and 99.58%, 99.37%, and 99.47% for Group 08. The large difference indicates that the richer the feature set is, the greater the promotion of recall, precision, and F_1 will be.

However, due to the feature differences among groups, the accuracy of the combined-group detection is generally lower than that of the single-group detection. It is worth noting that ecosystem features achieved approximately 90% accuracy in all single-group detections, but it decreased significantly in the combined-group detection. The random forest achieved the highest accuracy in both types of detection, with the combined-group detection achieving 98.75% accuracy (average accuracy of KNN and SVM is 99.4% and 99.46%, respectively). The table shows that Groups 03 and 06 with the longest registration time have the lowest accuracy. This indirectly confirms that the longer the zombie follower is cultivated, the more similar

Table 11: Detection accuracy of KNN for each feature category.

KNN	Basic	Content	Ecosystem	All features			
				Accuracy	Precision	Recall	F_1
01_72****37	98.55	92.05	90.16	99.57	96.48	99.30	97.87
02_72****04	99.83	—	97.89	99.82	95.90	95.41	95.65
03_72****91	99.07	91.36	87.27	99.14	95.04	97.66	96.34
04_72****62	99.77	—	—	99.77	54.84	70.83	61.82
05_72****21	99.89	—	—	99.89	50.00	100.00	66.67
06_72****43	98.00	93.25	67.06	97.99	88.88	90.60	89.73
07_72****47	99.28	—	80.28	99.36	94.72	99.00	96.81
08_72****28	98.74	90.59	90.74	99.64	97.11	99.16	98.12
All data	97.16	88.1792	66.32	97.59	95.56	97.85	96.69

Table 12: Detection accuracy of SVM for each feature category.

SVM	Basic	Content	Ecosystem	All features			
				Accuracy	Precision	Recall	F_1
01_72****37	98.38	92.32	90.21	99.55	97.01	98.48	97.74
02_72****04	99.82	—	97.96	99.83	96.76	94.71	95.72
03_72****91	99.25	98.08	88.10	99.06	98.71	93.32	95.94
04_72****62	99.86	—	—	99.86	78.95	85.71	82.19
05_72****21	99.93	—	—	99.93	73.68	93.33	82.35
06_72****43	98.19	92.64	90.26	98.28	94.30	87.59	90.82
07_72****47	99.38	—	91.62	99.43	95.33	98.99	97.12
08_72****28	98.89	92.11	90.84	99.73	98.52	98.63	98.57
All data	96.83	88.46	65.86	98.11	98.12	96.62	97.36

Table 13: Detection accuracy of Random forest for each feature category.

Random forest	Basic	Content	Ecosystem	All features			
				Accuracy	Precision	Recall	F_1
01_72****37	98.68	92.61	90.21	99.74	98.59	98.79	98.69
02_72****04	99.85	—	97.96	99.84	94.85	97.35	96.08
03_72****91	99.26	98.74	88.10	99.75	99.51	98.37	98.94
04_72****62	99.86	—	—	99.86	78.95	85.71	82.19
05_72****21	99.93	—	—	99.93	73.68	93.33	82.35
06_72****43	98.19	96.95	90.26	99.45	97.64	96.64	97.14
07_72****47	99.39	—	91.62	99.53	96.45	98.89	97.65
08_72****28	99.07	92.44	90.84	99.90	99.37	99.58	99.47
All data	97.16	92.11	65.86	**99.14**	**99.03**	**98.57**	**98.80**

it becomes to normal users and the more difficult its detection becomes.

6. Conclusion and Future Work

In this study, we focused on the features and detection of zombie followers from different companies. Through feature analysis, we described the current ecosystem of the zombie follower industry as follows: the ZF companies are constantly producing and cultivating zombie followers. Zombie followers that survive longer are more similar to normal users, thus lowering the detection rate by traditional methods. Furthermore, although the sources of zombie followers are different, the similar characteristics and the direct or indirect relationships between groups indicate that some zombie follower groups from different sources are actually controlled by the same organization. Finally, we used three different classification methods (i.e., KNN, SVM, and random forest) to detect zombie followers. The random forest performed the best with 99.14% accuracy. We also found that the richer the feature set, the greater the promotion of recall, precision, and F_1 of the detection results.

Interestingly, zombie follower services are only a small part of the black industry of malicious accounts. It also controls mass advanced social robot accounts, which have more advanced authentication and weight than normal users. Among them, some are compromised accounts and some have many real followers. In the future, we will conduct research on the features of such accounts.

Data Availability

The data used to support the findings of this study are included within the manuscript and its supporting information files.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the Sichuan Science and Technology Program, grant no. 2020YFG0076.

Supplementary Materials

All relevant data are included within the manuscript and its supporting information files. (*Supplementary Materials*)

References

[1] Sina IT, "Weibo posts the financial results for the fourth quarter and full year of 2019," 2020, https://tech.sina.com.cn/i/2020-02-26/doc-iimxyqvz6003265.shtml.

[2] S. Kumar, M. Jiang, T. Jung, R. J. Luo, and J. Leskovec, "MIS2: misinformation and misbehavior mining on the web," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 799-800, Los Angeles, CA, USA, February 2018.

[3] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, 2016.

[4] A. Thieltges, O. Papakyriakopoulos, J. M. Serrano, and H. Simon, "Effects of social bots in the Iran-debate on twitter," 2018, http://arxiv.org/abs/1805.10105.

[5] V. S. Subrahmanian, A. Azaria, S. Durst et al., "The DARPA twitter bot challenge," *Computer*, vol. 49, no. 6, pp. 38–46, 2016.

[6] E. E. Buckels, P. D. Trapnell, and D. L. Paulhus, "Trolls just want to have fun," *Personality and Individual Differences*, vol. 67, pp. 97–102, 2014.

[7] J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, and J. Leskovec, "Anyone can become a troll: causes of trolling behavior in online discussions," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pp. 1217–1230, Portland, OR, USA, February 2017.

[8] J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec, "Antisocial behavior in online discussion communities," in *Proceedings of the Ninth International AAAI Conference on Web and Social Media*, pp. 61–70, Oxford, UK, May 2015.

[9] S. Kumar, J. Cheng, J. Leskovec, and V. S. Subrahmanian, "An army of me: sockpuppets in online discussion communities," in *Proceedings of the 26th International Conference on World Wide Web WWW '17*, pp. 857–866, Perth, Australia, May 2017.

[10] S. K. Maity, A. Chakraborty, P. Goyal, and A. Mukherjee, "Detection of sockpuppets in social media," in *Proceedings of the Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pp. 243–246, Portland, OR, USA, February 2017.

[11] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, "Towards detecting compromised accounts on social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 447–460, 2017.

[12] E. Zangerle and G. Specht, "Sorry, I was hacked": a classification of compromised twitter accounts," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pp. 587–593, Gyeongju, South Korea, May 2014.

[13] J. P. Dickerson, V. Kagan, and V. S. Subrahmanian, "Using sentiment to detect bots on twitter: are humans more opinionated than bots?," in *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 620–627, Beijing, China, August 2014.

[14] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, "Detecting suspicious following behavior in multimillion-node social networks," in *Proceedings of the 23rd International Conference on World Wide Web*, pp. 305-306, Seoul, South Korea, April 2014.

[15] H. Jiang, Y. Wang, and M. Zhu, "Discrimination of zombie fans on Weibo based on features extraction and business-driven analysis," in *Proceedings of the 17th International Conference on Electronic Commerce*, p. 13, Seoul, South Korea, August 2015.

[16] M. Fazil and M. Abulaish, "Identifying active, reactive, and inactive targets of socialbots in twitter," in *Proceedings of the International Conference on Web Intelligence*, pp. 573–580, Leipzig, Germany, August 2017.

[17] M. Jiang, P. Cui, and C. Faloutsos, "Suspicious behavior detection: current trends and future directions," *IEEE Intelligent Systems*, vol. 31, no. 1, pp. 31–39, 2016.

[18] S. Kumar and N. Shah, "False information on web and social media: a survey," 2018, http://arxiv.org/abs/1804.08559.

[19] S. Lehmann and P. Sapieżyński, "You're here because of a robot," 2020, http://sunelehmann.com/2013/12/04/youre-here-because- of- a-robot.

[20] S. Stieglitz, F. Brachten, D. Berthelé, M. Schlaus, C. Venetopoulou, and D. Veutgen, "Do social bots (still) act different to humans?–comparing metrics of social bots with those of humans," in *Proceedings of the International Conference on Social Computing and Social Media*, pp. 379–395, Vancouver, Canada, July 2017.

[21] Y. Ji, Y. He, X. Jiang, J. Cao, and Q. Li, "Combating the evasion mechanisms of social bots," *Computers & Security*, vol. 58, pp. 230–249, 2016.

[22] M. Khademi, S. Hosseini Moghaddam, and M. Abbaspour, "An empirical study of the effect of profile and behavioral characteristics on the infiltration rate of socialbots," in *Proceedings of the 2017 Iranian Conference on Electrical Engineering (ICEE)*, pp. 2200–2205, Tehran, Iran, May 2017.

[23] S. Lee and J. Kim, "WarningBird: a near real-time detection system for suspicious URLs in twitter stream," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 3, pp. 183–195, 2013.

[24] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time URL spam filtering service," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, pp. 447–462, Berkeley, CA, USA, May 2011.

[25] S. Santhosinidevi, “Towards detecting compromised accounts on social networks,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 6, no. 4, pp. 71–73, 2018.

[26] Y. Shen, J. Yu, K. Dong, and K. Nan, “Automatic fake followers detection in Chinese micro-blogging system,” in *Advances in Knowledge Discovery and Data Mining*, pp. 596–607, Springer, Cham, Switzerland, 2014.

[27] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu, “Analyzing spammers’ social networks for fun and profit: a case study of cyber criminal ecosystem on twitter,” in *Proceedings of the 21st International Conference on World Wide Web*, pp. 71–80, Lyon, France, April 2012.

[28] Y. Zhang and J. Lu, “Discover millions of fake followers in Weibo,” *Social Network Analysis and Mining*, vol. 6, no. 1, p. 16, 2016.

[29] C. S.-H. Eom, W. Lee, J. J.-H. Lee, and W.-S. Cho, “Find spammers by using graph structure,” in *Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 278-279, Jeju, South Korea, February 2017.

[30] J. Wang, X. He, Q. Gong, Y. Chen, T. Wang, and X. Wang, “Deep learning-based malicious account detection in the momo social network,” in *Proceedings of the 2018 27th International Conference on Computer Communication and Networks (ICCCN)*, Hangzhou, China, July 2018.

[31] H. Shen and X. Liu, “Detecting spammers on twitter based on content and social interaction,” in *Proceedings of the 2015 International Conference on Network and Information Systems for Computers*, pp. 413–417, Wuhan, China, January 2015.

[32] M. Ikram, L. Onwuzurike, S. Farooqi et al., “Measuring, characterizing, and detecting facebook like farms,” *ACM Transactions on Privacy and Security (TOPS)*, vol. 20, no. 4, p. 13, 2017.

[33] M. Héder, “A black market for upvotes and likes,” *Információs Társadalom*, vol. 19, no. 4, pp. 18–39, 2020.

[34] Cristofaro, E. De, A. Friedman, G. Jourjon, M. A. Kaafar, and M. Zubair Shafiq, “Paying for likes?: understanding facebook like fraud using honeypots,” in *Proceedings of the 2014 Conference on Internet Measurement Conference*, pp. 129–136, Vancouver, Canada, November 2014.

[35] K. Lee, C. James, and S. Webb, “Uncovering social spammers: social honeypots + machine learning,” in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 435–442, Geneva, Switzerland, July 2010.

[36] X. Hu, J. Tang, and H. Liu, “Online social spammer detection,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence AAAI’14*, pp. 59–65, Québec City, Canada, July 2014.

[37] K. Lee, B. David Eoff, and C. James, “Seven months with the devils: a long-term study of content polluters on twitter,” in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, Barcelona, Spain, July 2011.

[38] C.-H. Xia, H.-K. Li, and G.-Z. Sun, “Microblogging malicious user identification based on behavior characteristic analysis,” *Computer Science*, vol. 45, no. 12, pp. 111–116, 2018.

[39] S. Junyi, “Jieba: Chinese text segmentation,” 2020, https://github.com/fxsjy/jieba/.

[40] H. Li, “Analysis and implementation of spammers detection method based on social network,” in *CNKI*Beijing Jiaotong University, Beijing, China, 2020, http://cdmd.cnki.com.cn/Article/CDMD-10004-1017086622.htm.

[41] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” *Communications of the ACM*, vol. 51, no. 1, pp. 117–122, 2008.

[42] J. A. K. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[43] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6682311, 13 pages
https://doi.org/10.1155/2021/6682311

Research Article

Ferry Node Identification Model for the Security of Mobile Ad Hoc Network

Zhifei Wang,[1,2] **Gang Xu**,[1,2] **Na Zhang**,[1,2] **Zhihan Qi**,[1,2] **Fengqi Wei**,[1,2] **and Liqiang He**[3]

[1] *College of Computer Science, Inner Mongolia University, Hohhot, China*
[2] *Inner Mongolia A.R. Key Laboratory of Wireless Networking and Mobile Computing., Hohhot, China*
[3] *Geomechanica Inc, Toronto, Canada*

Correspondence should be addressed to Gang Xu; csxugang@imu.edu.cn

Received 15 October 2020; Revised 2 December 2020; Accepted 11 December 2020; Published 6 January 2021

Academic Editor: Zhe-Li Liu

An opportunistic network is a special type of wireless mobile ad hoc network that does not require any infrastructure, does not have stable links between nodes, and relies on node encounters to complete data forwarding. The unbalanced energy consumption of ferry nodes in an opportunistic network leads to a sharp decline in network performance. Therefore, identifying the ferry node group plays an important role in improving the performance of the opportunistic network and extending its life. Existing research studies have been unable to accurately identify ferry node clusters in opportunistic networks. In order to solve this problem, the concepts of k-core and structural holes have been combined, and a new evaluation indicator, namely, ferry importance rank, has been proposed in this study for analyzing the dynamic importance of nodes in a network. Based on this, a ferry cluster identification model has been designed for accurately identifying the ferry node clusters. The results of the simulations conducted for verifying the performance of the proposed model show that the accuracy of the model to identify the ferry node clusters is 100%.

1. Introduction

An opportunistic network is a type of wireless mobile ad hoc network, which does not require a complete link between nodes and can realize communication between disconnected subdomains [1]. In contrast to the traditional networks, in an opportunistic network, messages rely on the encounter opportunities brought about by the node movement and are sent hop-by-hop in the network until they reach the destination node. Opportunistic networks are widely used in wildlife tracking, vehicle-mounted networks, remote areas, and communications in harsh environments [2].

Since the communication regions in an opportunistic network are fragmented most of the time, ferry nodes are placed between the disconnected areas in order to realize communication between the fragmented regions and enhance the overall performance of the opportunistic network. Ferry nodes moving between different regions connect to different areas. At present, research studies on ferry nodes mainly include routing algorithms based on ferry nodes [3–12], ferry node motion path planning [13–19], and ferry node network signal coverage [20–23].

In the actual application scenarios of opportunistic networks, ferry nodes play a vital role in maintaining communication between the separated areas of nodes. During network operation, if a ferry node withdraws from the network service due to cyberattacks, communication between different regions might get weakened or even get cut off. Therefore, identifying a ferry node cluster from an unfamiliar network environment and protecting these nodes play an important role in maintaining network security and improving the network performance. A few research studies have been conducted on ferry node identification in opportunistic networks, and the main idea in these studies has been the use of node importance evaluation indicators in a complex network for discovering ferry nodes in an opportunistic network. The existing node importance evaluation indicators mainly include degree centrality, betweenness centrality, and k-core indicator. A study in [24] proposed using the degree centrality of a node to measure the importance of the node and to estimate the position of the node in the network according to the number of neighbors of the node. Freeman [25] and Goh et al. [26] proposed the use of the betweenness centrality indicator to evaluate the importance of nodes by using the number of shortest paths through the nodes to the rest of the network. Kitsak et al. [27] proposed a node importance evaluation indicator based on the importance of the position of the node in the entire network and used the number of cores obtained by k-core decomposition as the basis for judging the importance of the node. However, this method is only suitable for complex networks with a static topology and cannot be used in opportunistic networks with constantly changing topologies. In a dynamically changing network, the degree of nodes changes constantly, and it is difficult to find a node that is always at the center of the network using methods based on degree centrality [28]. In a network consisting of moving nodes, the number of shortest paths through the nodes to other nodes is constantly changing. In this case, the betweenness centrality indicator is unable to accurately determine the number of times a node is in the shortest path. In a network involving a single propagation source, the k-core indicator is more accurate than the degree centrality and betweenness centrality indicators in identifying nodes with greater influence in the network. However, this method is ineffective in complex networks involving multiple propagation sources. Due to the features of the opportunistic network, such as its dynamic changing topology, a constant movement of nodes, and multiple propagation sources, none of the abovementioned node importance evaluation indicators are suitable for the identification of ferry node clusters in such networks.

With the aim of addressing these shortcomings of the existing methods, the concepts of structural holes of nodes and k-cores have been combined in this study, and the ferry importance rank (FIR) indicator has been proposed, which comprehensively analyzes the local and global importance of nodes in a network. In a changing and multipropagation source opportunistic network, the dynamic importance of nodes can be accurately evaluated by the proposed FIR indicator. On this basis, an FIR-based opportunistic network ferry node cluster identification model has been developed, which divides the operation information of the network over a period of time into equal-length time slices. In each time slice, the nodes that can have significant influence are calculated on the basis of the FIR indicator, and the ferry node cluster in the network is selected according to the nodes selected in the different time slices.

2. Related Work

Since opportunistic networks have the characteristics of unstable network topology, irregular node movement, nonfixed connections, and unpredictable encounters [29, 30], the existing key node mining methods based on static complex network analysis techniques cannot be applied to the discovery and selection of ferry nodes in opportunistic networks. In the existing key node mining algorithms of complex networks, researchers have primarily used indicators such as the degree centrality, betweenness centrality, near-centrality, and feature vector centrality for calculating the importance of nodes in complex networks from different perspectives. In opportunistic networks involving sparse nodes, the key node mining algorithms based on the degree centrality index are unable to find the bridge nodes [31]. The key node mining algorithm based on the betweenness centrality indicator has a high time complexity [32], and the feature vector centrality of a node ignores the influence of adjacent node changes on the importance of nodes. Therefore, the key node discovery algorithm based on complex networks cannot be applied to ferry node discovery in opportunistic networks.

A key node mining algorithm based on the node degree centrality index analyzes the number of neighbors of a node and the local importance of key nodes but ignores the importance of nodes in the global network topology [31]. On the basis of the degree centrality indicator, Chen [33] proposed a semilocal centrality key node identification method, which partially improved the degree centrality method by calculating the sum of the degrees of all nodes within a certain number of hops. This method partially improved the situation where the degree centrality method falls into a locally optimal solution. Based on the degree centrality and node deletion method, and in combination with the local connectivity of social networks and the shortest path between nodes, Li et al. proposed a connectivity centrality index to measure the influence of nodes in a network [34]. Their method integrated the global and local importance of nodes in the network and more comprehensively described the importance of nodes. However, due to its high computational complexity, their method cannot be used for ferry node discovery in opportunistic networks under dynamic topology.

The betweenness centrality index considers the importance of nodes from a global perspective and can efficiently judge the bridge nodes in a network [25]. The fast approximation algorithm based on the random sampling of the shortest path can quickly calculate the betweenness centrality and evaluation importance of all nodes in a large network [35]. A key node discovery algorithm based on the betweenness and closeness centrality for the shortest path used the closeness centrality to analyze the key nodes [36]. In an opportunistic network with social attributes, key nodes can be mined based on the fusion of the weight of the node in the social relationship and its position in the network topology [37]. However, none of the abovementioned key node mining algorithms can complete the discovery and selection of ferry nodes in an opportunity network, where the network topology changes dynamically.

Kitsak et al. [27] proposed that the importance of a node depends on the position of the node in the entire network. They calculated the number of node cores based on k-shell and used the k-core index to describe the propagation ability of the node, which can accurately identify the most influential node in the network. However, this method is unsuitable for opportunistic networks with multiple propagation sources. Burt proposed the structural hole theory [38], pointing out that a node with larger structural holes plays a more important role in the communication of the surrounding nodes in the network. The structural hole theory can calculate the structural relationship between multiple nodes and solves the problem of the k-core index not being able to reflect the structural characteristics of neighbors. Zhang and Zhang [39] estimated the importance of nodes by calculating the structure between them, fully considering the influence of the network structure on the importance of nodes. Su et al. [40] combined the structural hole importance of nodes and their neighborhood, which comprehensively considered the number of neighbors of the node and the topological structure between the neighbors. In addition, their method used structure holes to determine the key node, and the calculation range was extended from the neighboring nodes to the neighboring regions. However, in these studies, the problem of falling into the local optimal solution could not be avoided.

The importance of nodes in a network is affected by many factors, and the existing methods based on a single importance evaluation index cannot find the key nodes of a network accurately [41, 42]. In order to solve this problem, Zhou et al. merged the node efficiency, degree of the node, and the importance of adjacent nodes to form an importance evaluation matrix for mining key nodes in a network [43]. However, this method did not consider the impact of nonadjacent nodes having a high interdependence on the key nodes. Reference [44] combined the concepts of structural holes and the closeness centrality index for obtaining the influence matrix of the structural holes of a node and analyzed the global and local importance of nodes. However, the time complexity of calculating the node closeness centrality was relatively high. Reference [45] mined the key nodes by fusing the local, behavioral, and global characteristics of nodes and, based on the time slice method, transformed the dynamic topology in the opportunistic network into a static topology set, which provided a new method for mining the key nodes in an opportunistic network.

In summary, due to the mobility of nodes in a network, the topology of the network is also in an unstable state, and a message propagates via multiple propagation sources. Thus, the existing key node mining methods based on analyzing complex networks cannot be applied to an opportunistic network directly. In order to automatically find and select the ferry nodes in opportunistic networks, the Ferry_Importance_Rank (FIR) index has been proposed in this work. This index has been used for evaluating the dynamic importance of nodes in opportunistic networks on the basis of the importance of structure holes and the k-core index. In addition, based on FIR, a ferry node cluster identification model has been designed for the opportunistic networks, using which the analysis of the opportunistic networks can perform the discovery and selection of the ferry node group in unfamiliar opportunistic networks.

The remainder of this paper has been organized as follows: Section 3 provides a description of the three indicators for estimating the node importance in opportunistic networks. Details of the proposed FIR-based ferry node identification model for opportunistic networks, based on the importance of structural holes and k-core index, have been given in Section 4. Section 5 presents a comprehensive set of simulation results for various opportunistic network scenarios. A detailed discussion of the analysis of the results has also been given in this section. A summary of the present work and the conclusions drawn from it have been given in Section 6.

3. Preliminaries

In this section, some preliminary knowledge, including the structural holes theory and k-core importance theory, has been reviewed.

Definition 1. Structural holes

"Structural holes" is a classical theory of social networks developed by Burt [38], which is often used for evaluating the importance of nodes in local networks. If nodes B and C are neighbors of node A and nodes B and C are not adjacent and can only communicate via node A, there exists a structure hole between nodes B and C, or there is a structural hole on node A. The larger the number of structural holes possessed by a node is, the stronger its communication ability is.

3.1. Computing Methods for Estimating the Importance of Structural Holes. Suppose that the number of nodes in a network is n; then a matrix A of dimensions $n \times n$ is established. This matrix is used for representing the connection status of the nodes in a network. $a_{ij} = 0$ implies that nodes i and j are disconnected, whereas $a_{ij} = 1$ implies that nodes i and j are connected.

Suppose that $k(i)$ is the degree of node i; then $k(i)$ is calculated using equation (1), where G is the set of all the nodes in the topology map.

$$k(i) = \sum_{j \in G} a_{ij}, \tag{1}$$

and $Q(i)$ is the adjacency degree of node i. It is the sum of degrees of all the neighbors of node i, as expressed in equation (2).

$$Q(i) = \sum_{\varpi \in r(i)} k(\varpi). \tag{2}$$

In Equation 2, $r(i)$ is the set of neighboring nodes of node i.

The network constraint coefficients of nodes are related to multiple factors such as the connection of a node with the other nodes and the structure between a node and its neighbor. Therefore, the degree and the topology structure of its neighborhood, P_{ij}, should be taken into consideration when calculating the network constraint coefficient of nodes. P_{ij} is calculated using the expression given in

$$P_{ij} = \frac{Q(j)}{\sum_{v \in r(i)} Q(v)}. \tag{3}$$

The difficulty for a node to form structural holes is represented by the network constraint coefficient, RC_{i}, of the node, which is also a measure of the size of the structural holes of the node. The network constraint coefficient of a node is inversely proportional to its degree of structural holes and is calculated using

$$\mathrm{RC}_i = \sum_{i\in r(i)} \left(P_{ij} + \sum_{q} P_{iq} P_{qj} \right)^2, \tag{4}$$

where q is a node in the intersection of the neighbors of nodes i and j, which is not equal to i or j.

The constraint coefficient of node i is the structural hole importance index of the node, and the ratio of the sum of the constraint coefficients of all nodes in the network is calculated using

$$L_i = \frac{1 - \mathrm{RC}_i}{\sum_{j=1}^{n} \left(1 - \mathrm{RC}_j\right)}. \tag{5}$$

Definition 2. Structural hole constraint coefficient

The structural hole constraint coefficient, L_{ni}, of node i is defined as the ratio of the structural hole importance index value of node i to the sum of the structural hole importance index values of all nodes neighboring node i. L_{ni} is used for measuring the constraints for a node when forming structural holes. It is calculated using

$$L_{ni} = \frac{L_i}{\sum_{k\in r(i)} L_k}. \tag{6}$$

In this work, an algorithm for calculating the importance of structural holes in opportunistic networks has been developed, as shown in Algorithm 1, namely, the calculate structural hole importance (CSHI) algorithm.

Definition 3. K-core importance

Being a classical concept in graph theory, the k-core theory calculates the influence of nodes in a network based on the degree of nodes. The steps of k-core decomposition are as follows: recursively delete the nodes having a degree of k or less in the network and assign k-shell values to the deleted nodes. Repeat the process until all nodes in the network are assigned k-shell values. In the k-core decomposition algorithm, a large number of nodes are at the same network level, which leads to the incapability of the algorithm when further calculating their node importance.

3.2. Calculation of K-Core Importance.

3.2. Calculation of K-Core Importance. In the initial stage, $k_i^m = k(i)$ are recorded for every node, the nodes with the smallest k_i^m value are removed from the topology map, and k_i^m are assigned to K_{s_i} of these nodes. Subsequently, k_i^m of the remaining nodes are updated as $k_i^m = k_i^r + \lambda_k^e$, where λ is the adjustment factor and $0 \leq \lambda \leq 1$, ${}_k^e$ is the removed degree of the previous stage, and k_i^r is the degree of the remaining nodes. The above process is repeated until all the nodes obtain the K_{s_i} value. Then, K_{s_i} is the k-core index of node i.

According to the calculation method of k-core importance, this paper presents the calculation algorithm of the k-core index of nodes in an opportunistic network, as shown in Algorithm 2, namely, the calculate k-core importance (CKCI) algorithm.

The k-core importance of node i refers to the ratio of K_{s_i} of node i and the sum of K_{s_i} of all nodes, which can be calculated using equation (7) after the k-core index of node i is known:

$$M_i = \frac{K_{s_i}}{\sum_{j=1}^{n} K_{s_j}}. \tag{7}$$

4. Ferry Node Identification Model Based on FIR for an Opportunistic Network

The topology of an opportunistic network changes dynamically. Therefore, the existing key node mining algorithms based on static network indicators are inapplicable in an opportunistic network. In order to solve the problem of mining the key nodes in an opportunistic network, we have divided the network into several snapshots with equal runtime, established a static topology of the opportunistic network in the snapshots, mined the key nodes in each snapshot, and determined the ferry nodes in the opportunistic network based on the frequency with which the key nodes are selected.

The use of a single indicator cannot evaluate the importance of nodes accurately [41]. In order to fix the shortcomings of using a single index, the FIR_i index has been proposed in this work, which is an indicator of the fusion of structural hole importance of a node and k-core importance. This index comprehensively analyzes the local and global importance of nodes based on the ferry node group in the FIR_i election network.

4.1. Ferry Importance Rank Algorithm. The node importance evaluation model based on the FIR index has been used for measuring the importance of nodes in an opportunistic network. The model combines the k-core importance, M_i, of the node and the structural hole importance, L_{ni}, and calculates the FIR_i index. The larger the FIR_i value is, the higher the importance of the node is. To enable the mining of key nodes in the dynamic topology of an opportunistic network, the FIR-based model establishes a time-slice snapshot sequence for the running opportunistic network topology, conducts key node mining on the basis of the static topology graph sequence, and counts the number of times each node is selected as a key node. The selected node is the ferry node, and the node that has been selected for the highest number of times is the most important node among all ferry nodes. The FIR_i indicator is calculated using the expression given in

$$\mathrm{FIR}_i = \alpha M_i + \beta L_{ni}. \tag{8}$$

From the analysis of the simulations performed for verifying the proposed model, it is found that when α is 2

```
Input: node set $G$
Output: structural hole constraint coefficient, $L_{ni}$, of all nodes in $G$
for i ∈ G: //G is the node set of the opportunistic network
    Calculate k ($i$)
for i ∈ G:
    Calculate Q (i)
for i ∈ G:
    for $j \in r(i)$: //$r(i)$ is the set of all neighboring nodes of node $i$
        Calculate $\mathrm{P_{ij}}$
        for $q \in (r(i) \cap r(j))$:
            Calculate $P_{iq}$ and $P_{qj}$
        Calculate $\mathrm{RC}_i$
for i ∈ G:
    Calculate $L_i$
for i ∈ G:
    Calculate $L_{ni}$
return Ln
```

Algorithm 1: Algorithm for calculating the importance of structural holes.

```
Input: node set G
Output: K-core importance of all nodes in the set G
for i ∈ G:
    $k_i^m = k(i)$
$G_n = G$
while $G_n \neq \varnothing$:
    $k^e = 0$
    for $j \in G_n$: //$G_n$ represents the set of the remaining nodes in the topology graph
        if $k_j^m < \forall k_q^m$: $(q \in G_n, q \neq j)$
            $K_{s_j} = k_j^m$
            $k^e = k_j^m$
            $G_n = G_n \setminus \{j\}$
    for $l \in G_n$:
        $k_l^m = k_l^m + \lambda k^e$
return $K_s$
```

Algorithm 2: Algorithm for calculating the k-core index.

and β is 1, the node importance evaluation model achieves the best performance.

The FIR_i index is calculated by performing the following steps: the structural hole weight, L_n, and the k-core index, K_s, of each node in the node set G are calculated using Algorithms 1 and 2. Following this, the k-core importance, M_i, of each node is calculated according to the expression given in Equation 7. Finally, the FIR importance index, FIR_i, of node i in the node set G is calculated according to Equation 8. The calculation process is shown in Algorithm 3, namely, the calculate ferry importance rank (CFIR) algorithm.

To calculate the ferry node cluster in a network, the key nodes in each snapshot should be calculated separately. First, all nodes in the topology are inputted into set G, and the FIR values of all nodes in G are calculated according to Algorithm 3. The node with the highest FIR value in the topology map, which corresponds to this time slice, is inputted into the ferry node set. All the key nodes in the snapshot are calculated using the method described above, and the final set of ferry nodes is the ferry node group in the network. The ferry node identification algorithm for an opportunistic network is shown in Algorithm 4, namely, the ferry node identification (FNI) algorithm flow of the ferry node cluster identification algorithm is shown in (Figure1)

5. Experiments

The opportunistic network environment (ONE) simulator is an important experimental simulation platform for studying opportunistic networks. In this study, ONE1.4.1 was used for performing simulations, and the report is ConnectivityDtnsim2Report. The parameter settings of the simulation environment are listed in Table 1. In this study, three different simulation scenarios were set up and were compared with the betweenness-based algorithm (referred to as the VC model) [35]. The FIR-based model proposed in this study and the VC model were used for identifying the

```
Input: node set G
Output: FIR importance of all the nodes in the set G
$L_n$ = CSHI(G)
$K_s$ = CKCI(G)
for i $\in$ G:
    Calculate $M_i$ according to Equation 7
for i $\in$ G:
    $\mathrm{FIR}_i = \alpha M_i + \beta L_{ni}$
return FIR
```

Algorithm 3: Algorithm for calculating the FIR importance.

```
Input: the set of time slice (T)
Output: ferry nodes in the network
Ferry = {∅}
for $t \in T$:
    G = nodes in t
    FIR = CFIR(G)
    $\mathrm{Ferry} \cap \left\{ i | \mathrm{FIR}_i \geq \forall \mathrm{FIR}_j,\ i, j \in G \right\}$
return Ferry
```

Algorithm 4: Algorithm for identification of ferry nodes.

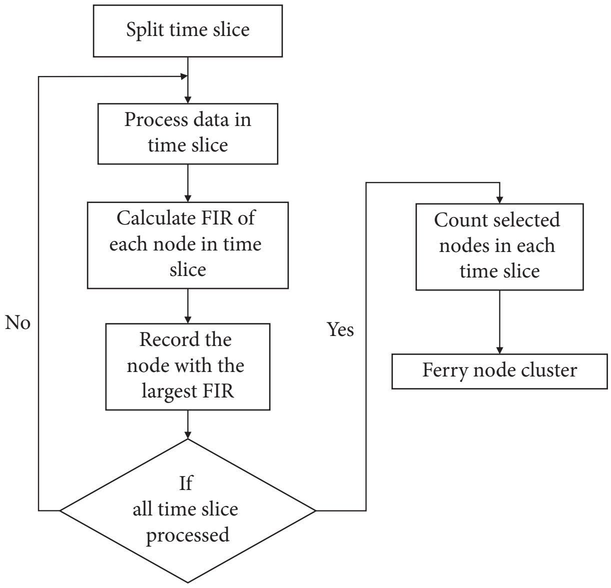

Split time slice
Process data in time slice
Calculate FIR of each node in time slice
Record the node with the largest FIR
If all time slice processed
No
Yes
Count selected nodes in each time slice
Ferry node cluster

Figure 1: Flowchart of the ferry node cluster identification algorithm.

ferry nodes in different scenarios, and the performance of the models in identifying the nodes in different environments was analyzed.

The three simulation scenarios used in this study are an opportunistic network with sparse nodes, an opportunistic network with dense nodes, and an opportunistic network with star-shaped distribution of nodes (referred to as scenarios 1, 2, and 3, respectively). Scenario 1 was used for simulating the nodes that are sparsely distributed in operating environments, for example, the opportunistic networks in scenarios such as grasslands, remote villages, or agricultural and pastoral areas. Scenario 2 was used for simulating an operating environment with densely distributed nodes, such as campus environments and opportunistic networks in urban environments. Scenario 3 was used for simulating the operating environment where the nodes are unevenly distributed, such as an opportunistic network in the mountains, forests, and other scenarios,

Table 1: The parameter of the simulation scenario.

Category	Parameter	Values
Computer configuration	CPU	i7 9700K
	OS	Windows 10 Professional
	RAM	8 G
Scenario settings	Simulation area size	$200 * 200\ m^2$
	Simulation time	24 h (86400 s)
	Message transmission carrier	Bluetooth device
	Message transmission range	50 m
	Nodes movement model in the region	MapRouteMovement (MRM)
	Node movement model between regions	RandomWaypoint (RWP)
	Number of nodes in the region	10
	Nodes moving speed in the region	1 m/s
	Nodes moving speed between regions	5 m/s
Sparse multiparallel opportunistic network	Number of experimental regions	4
	Number of nodes between regions	4
Dense multiparallel opportunistic network	Number of experimental regions	6
	Number of nodes between regions	6
Star-shaped multiparallel opportunistic network	Number of experimental regions	5
	Number of nodes between regions	4

where the terrain is more restricted. The simulation environment setup in this study covered various practical application scenarios of the opportunity network and comprehensively verified the effect of using the FIR model in an actual operating environment. The schematics of the simulation environment setups are shown in Figures 2–4.

Figure 2 shows the schematic of an opportunistic network involving a relatively sparse node distribution. In this scenario, the number of node groups is small, and the geographical distribution of the node groups is sparse. Figure 3 shows the schematic of an opportunistic network with densely distributed nodes. Compared to scenario 1, this scenario has a larger number of node groups, and the node groups are densely distributed. Figure 4 shows an opportunistic network with a star-shaped node distribution. This scenario is used for simulating a scene where the distribution of nodes is irregular due to the terrain constraints such as mountains and river valleys. Simulations were carried out using opportunistic networks with different node distributions, and the results thus obtained are shown in Figures 5–16.

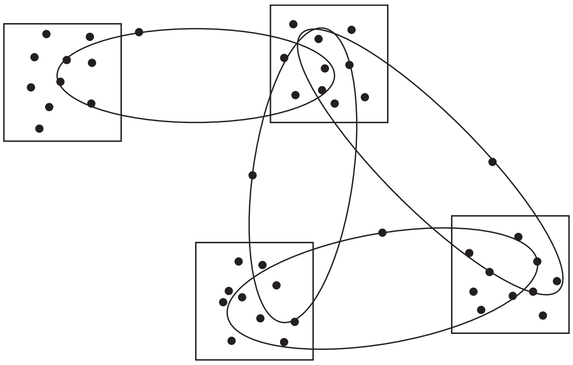


Figure 2: Schematic of the sparse multiparallel ferry opportunistic network.

Figures 5–7 show the comparison of the results obtained from the FIR-based model developed in this work and the VC model election ferry node group in opportunistic networks in different scenarios. In Figures 5–7, the node identified in the upper right corner is the ferry node set marked in the scene, and the blue line represents the other nodes in the scene. From the analysis of the simulation results of different scenarios in the opportunistic network, it can be observed that when the time slice length is less than 200 s, the FIR-based model proposed in this work has a lower accuracy of mining ferry nodes in different scenarios. This is because when the time slice length is less than 200 s, the distribution of nodes is sparse, and the ferry nodes are ignored because they do not form effective connections between regions. The error rate is significantly reduced for time slices greater than 500 s. When the time slice length is greater than 1000 s, the accuracy rate is higher, and the rate of recognizing the ferry nodes in the opportunistic network in the above three scenarios is 100%.

Further, the FIR-based model and the VC model were used for performing simulations in the above scenarios for time slices of 1200, 1800, and 2400 s, respectively. The results obtained from these simulations are shown in Figures 8–16 below.

Figures 8–10 show the comparison results of the simulation performed for scenario 1. In this, the simulation results obtained by applying the FIR-based model and the VC model have been compared for the case of sparse multiparallel opportunistic network, when the time slice length was set to 1200, 1800, and 2400 s, respectively, and the nodes h43, g32, f21, and e10 were set as the ferry nodes. From Figure 8, it can be seen that the FIR-based model can accurately identify all ferry nodes, whereas the VC model can only identify a few of them. Thus, the analysis proves the effectiveness of the use of the FIR-based model in opportunistic networks involving sparse nodes.

Figures 11–13 are the simulation results for scenario 2. In this simulation, the time slice length was set to 1200, 1800, and 2400 s, respectively, and the ferry nodes were set

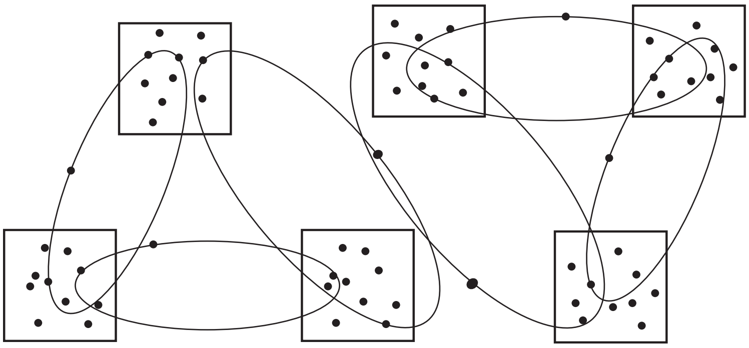


Figure 3: Schematic of the dense multiparallel ferry opportunistic network.

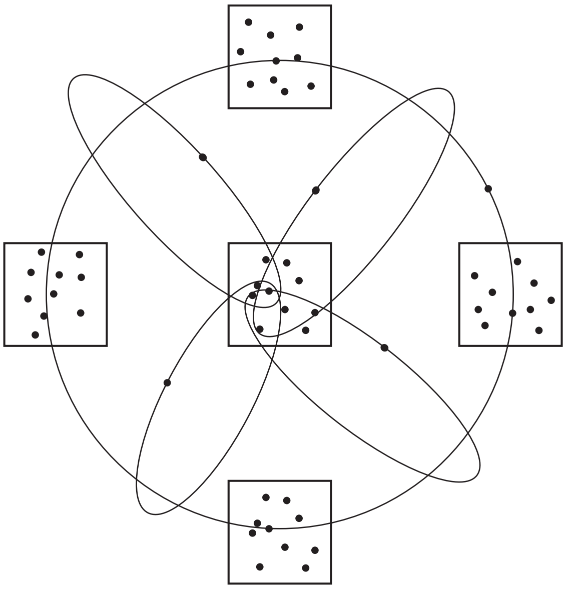


Figure 4: Schematic of the star-shaped multiparallel ferry opportunistic network.

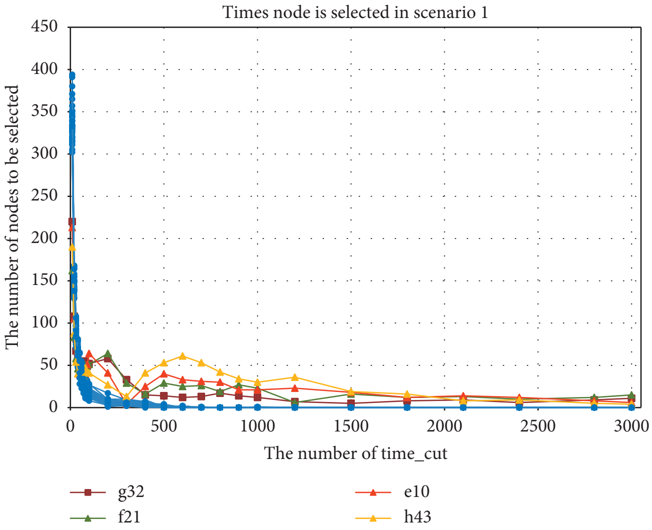


Figure 5: Comparison of the simulation results for mining ferry nodes in a sparse multiparallel opportunistic network.

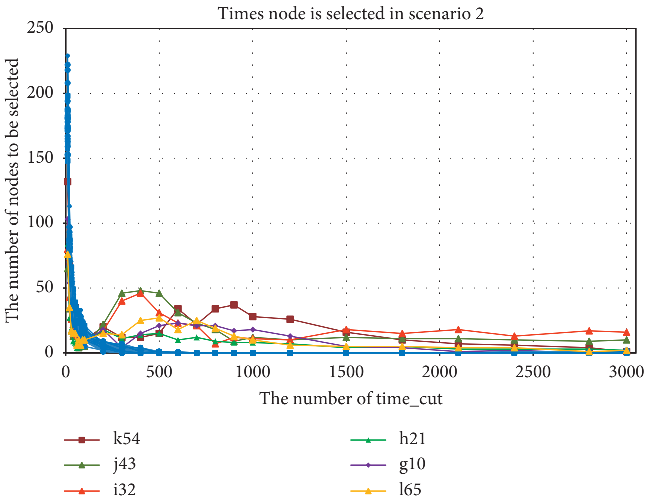


Figure 6: Comparison of the simulation results for mining ferry nodes in a dense multiparallel opportunistic network.

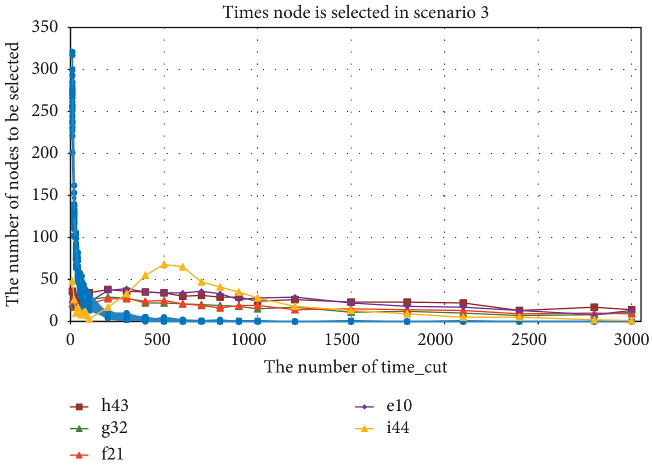


Figure 7: Comparison of the simulation results for mining ferry nodes in a star-shaped multiparallel opportunistic network.

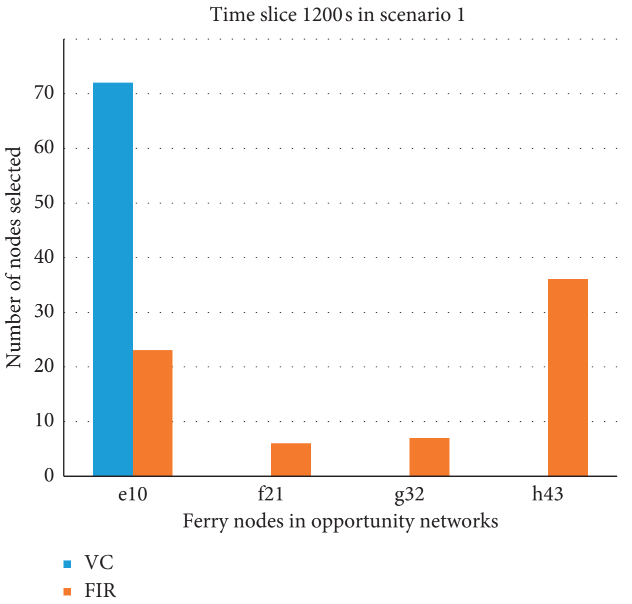


Figure 8: Comparison of the simulation results for the discovery of the ferry nodes in scenario 1 when the time slice length is 1200 s.

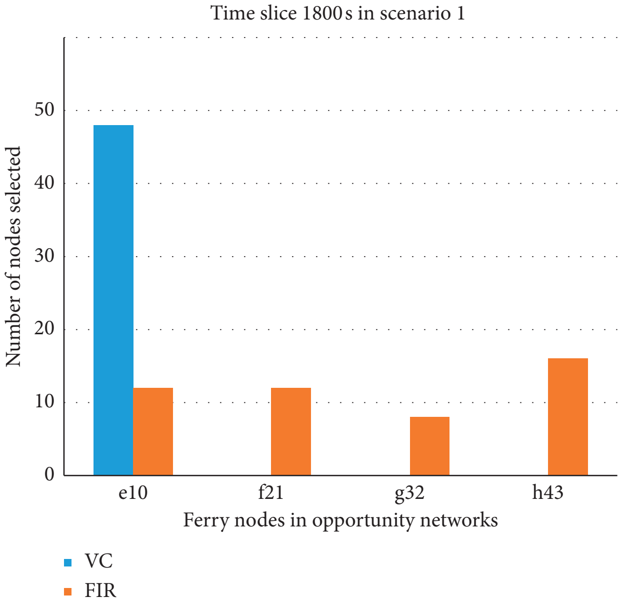


Figure 9: Comparison of the simulation results for the discovery of the ferry nodes in scenario 1 when the time slice length is 1800 s.

to g10, h21, i32, j43, k54, and l65, respectively. From the simulation results, it can be seen that, in scenario 2, the FIR-based model and the VC model exhibit similar results in identifying the ferry nodes. This is because when the distribution of nodes in the opportunistic network is dense, the topology between the nodes is more stable. In this case, the VC model, which is based on the betweenness centrality indicator, can achieve better results. When the time slice of 2400 s was taken, the number of ferry nodes g10, h21, i32, j43, k54, and l65 selected by the FIR-based model was 2, 3, 13, 10, 6, and 4 and the hit rates were 5%, 8%, 34%, 26%, 16%, and 11%, respectively. Further, the higher the ferry node hit rate is, the more important its role in the ferry node group is and the greater its impact on the network is.

In the opportunistic network of scenario 3, the distribution of nodes is star-shaped. The comparison of the simulation results of the FIR-based model and the VC model is shown in Figures 14–16. In scenario 3, the nodes e10, f21, g32, h43, and i44 were set as the ferry nodes. The simulation results show that the FIR-based model is able to identify all ferry nodes, whereas the VC model recognizes only node i44. This indicates that the FIR algorithm can accurately find the ferry nodes groups in the opportunistic network of scenario 3.

In summary, the VC model, which is based on the betweenness centrality indicator, can only identify all ferry node groups in opportunistic networks involving dense nodes, whereas the FIR-based model proposed in this work is able to find the ferry nodes groups in various

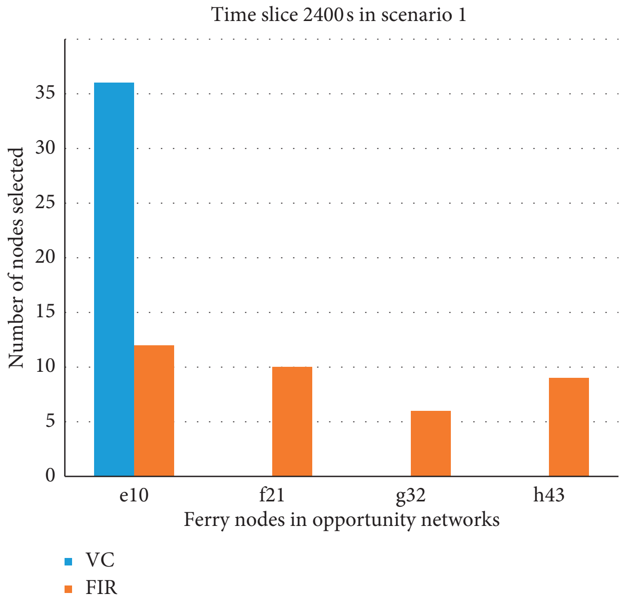


Figure 10: Comparison of the simulation results of the discovery of the ferry nodes in scenario 1 when the time slice is 2400 s.

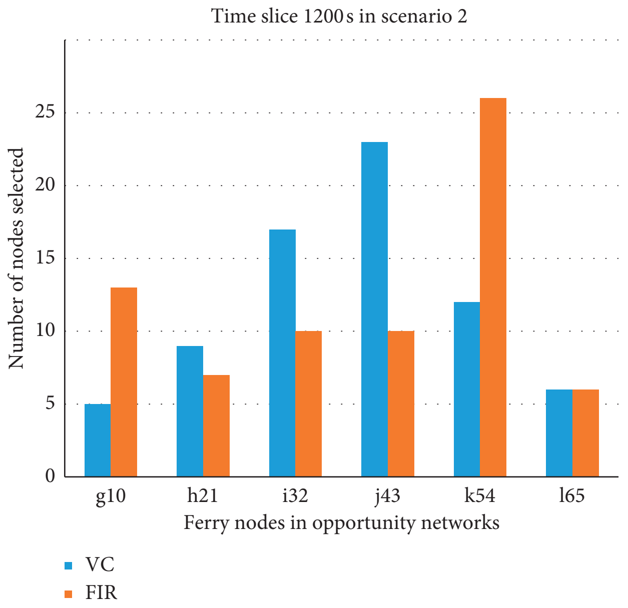


Figure 11: Comparison of the simulation results for the discovery of ferry nodes in scenario 2 when the time slice is 1200 s.

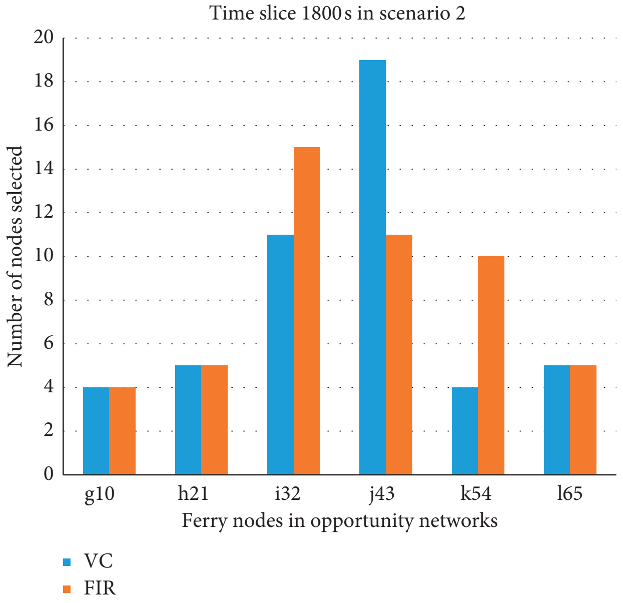


Figure 12: Comparison of the simulation results for the discovery of ferry nodes in scenario 2 when the time slice is 1800 s.

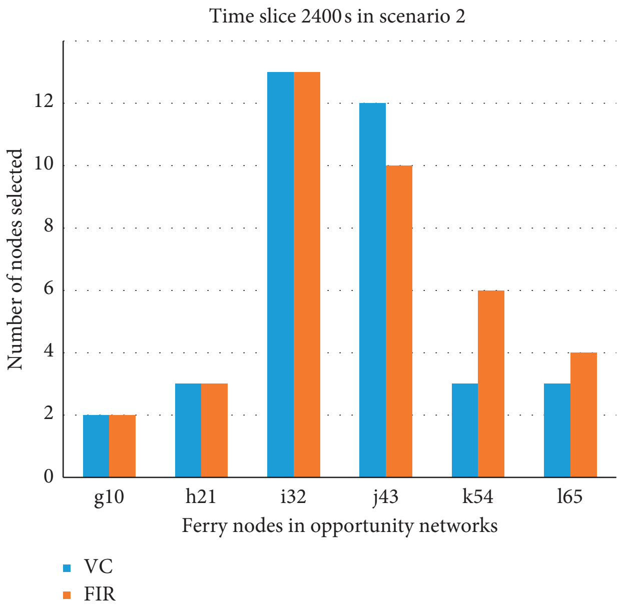


Figure 13: Comparison of the simulation results for the discovery of ferry nodes in scenario 2 when the time slice is 2400 s.

common opportunistic networks. This effectively fixes the problem of omission of the existing ferry node mining algorithm based on the betweenness centrality indicator in scenarios 1 and 3. The FIR-based model exhibits a 100% success rate in identifying the ferry nodes in all three scenarios, whereas the VC model only reaches 100% identification of ferry nodes groups in scenario 2. In scenarios 1 and 3, the recognition rate of ferry nodes groups by the VC model is only 25%. Thus, it can be seen that the ferry node identification model in an opportunistic network based on FIR can find the ferry nodes in the opportunistic networks effectively and reliably.

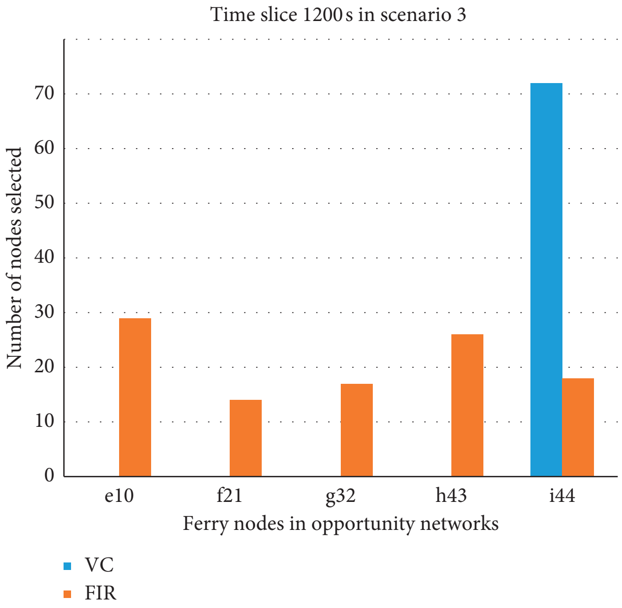


Figure 14: Comparison of the simulation results for the discovery of ferry nodes in scenario 3 when the time slice is 1200 s.

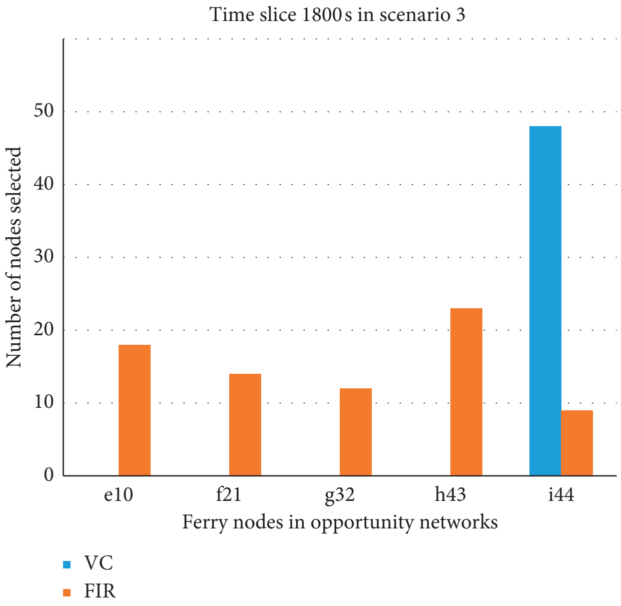


Figure 15: Comparison of the simulation results for the discovery of ferry nodes in scenario 3 when the time slice is 1800 s.

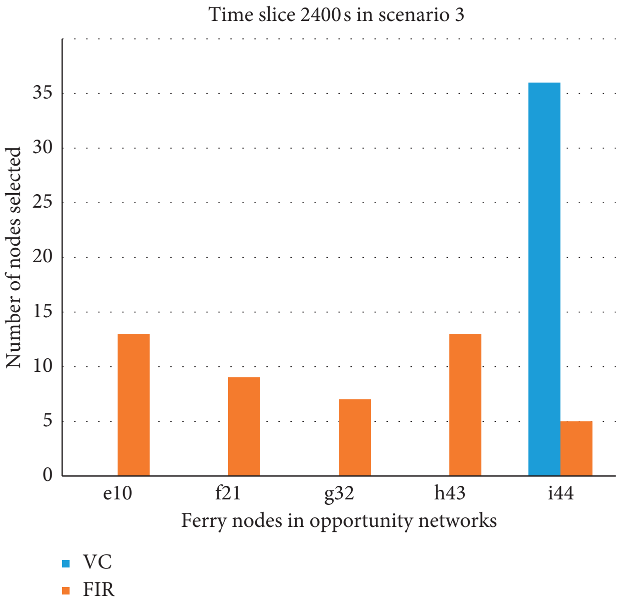


Figure 16: Comparison of the simulation results for the discovery of ferry nodes in scenario 3 when the time slice is 2400 s.

6. Conclusion

Based on the importance of structural holes and k-cores, an FIR indicator has been proposed in this study for evaluating the importance of nodes in the opportunistic network. Based on this indicator, an FIR-based opportunistic network ferry node identification model has been proposed. Compared to the VC model based on the betweenness centrality index, the FIR-based model is able to accurately identify the ferry node groups in a variety of application scenarios. An analysis of the local importance of nodes has been done in this study through the structural hole constraint coefficients, and the k-core importance of nodes has been used to analyze the global importance of nodes. Further, the FIR indicators have been proposed by fusing the structural hole attributes and k-core importance of nodes, which can evaluate the importance of nodes in an opportunistic network with dynamic topology changes and then identify the ferry nodes in the opportunistic network. Results of the simulations performed for verifying the proposed model have proved that the FIR-based model proposed in this work can accurately and efficiently identify the ferry nodes in opportunistic networks under low time complexity. In addition, it solves the problem of missing ferry nodes in the VC model and provides an important research foundation for an opportunistic network to automatically identify ferry nodes, protect these nodes in a targeted manner, and maintain network security.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grants 62061036, 61841109, and 61661041 and Natural Science Foundation of Inner Mongolia under Grant 2019MS06031 and in part by the CERNET Innovation Project under Grant NGII20170622.

References

[1] E. A. Abdellaoui Alaoui, H. Zekkori, and S. Agoujil, "Hybrid delay tolerant network routing protocol for heterogeneous networks," *Journal of Network and Computer Applications*, vol. 148, Article ID 102456, 2019.

[2] Y.-P. Xiong, L.-M. Sun, J.-W. Niu, and Y. Liu, "Opportunistic networks," *Journal of Software*, vol. 20, no. 1, pp. 124–137, 2009.

[3] C. Tao and J. Gao, "Modeling mobile application test platform and environment: testing criteria and complexity analysis," in *Proceedings Of the 2014 Workshop On Joining AcadeMiA and Industry Contributions To Test Automation And Model-Based Testing*, pp. 28–33, San Jose, CA, USA, February 2014.

[4] P. A. Poole-Wilson, G. A. Langer, J. Cheng, and T. Uehara, "Effect of pH on ionic exchange and function in rat and rabbit myocardium," *The American Journal of Physiology*, vol. 229, no. 3, pp. 570–581, 1975.

[5] S. Li, W. Qu, C. Liu, T. Qiu, and Z. Zhao, "Survey on high reliability wireless communication for underwater sensor networks," *Journal of Network and Computer Applications*, vol. 148, Article ID 102446, 2019.

[6] K. Ikenoue and K. Ueda, "Routing method based on data transfer path in DTN environments," in *Proceedings of the Transfer Path in DTN Environments. International Conference on Broadband and Wireless Computing, Communication and Applications*, pp. 544–552, Cham, Switzerland, 2019.

[7] K. J. Buhmeyer, A. R. Hutson, W. Li, and F. Zeng, "MEDEX South Carolina: a progress report," *Journal of the South Carolina Medical Association (1975)*, vol. 71, no. 11, pp. 337-338, 1975.

[8] S. Krug, M. Helbig, and J. Seitz, "Poster: utilization of additional nodes in hybrid DTN-MANET scenarios," in *Proceedings Of the 12th Workshop On Challenged Networks*, pp. 35–37, Snowbird, Utah, USA, October 2017.

[9] R. Vallikannu, A. George, and S. K. Srivatsa, "Routing and charging scheme with ferry nodes in mobile Adhoc networks," in *Proceedings of the 2017 International Conference on Intelligent Computing and Control (I2C2)*, pp. 1–4, Coimbatore, India, June 2017.

[10] Z. Liu, J. Li, S. Lv et al., "EncodeORE: reducing leakage and preserving practicality in order-revealing encryption," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[11] C. Wu, T. Yoshinaga, D. Bayar, and Y. Ji, "Learning for adaptive anycast in vehicular delay tolerant networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 1379–1388, 2019.

[12] Z. Du, C. Wu, X. Chen, X. Wang, T. Yoshinaga, and Y. Ji, "A VDTN scheme with enhanced buffer management," *Wireless Networks*, vol. 26, no. 3, pp. 1537–1548, 2020.

[13] R. Anguswamy, M. Thiagarajan, and C. H. Dagli, "Systems Methodology and Framework for problem definition in Mobile ad hoc networks," in *Proceedings of the 2nd Annual IEEE Systems Conference*, pp. 1–7, Montreal, Quebec, April 2008.

[14] T. Wang and C. P. Low, "Reducing message delay with the general Message Ferry Route (MFR∗) problem," in *Proceedings of the IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 380–387, New York, USA, October 2011.

[15] A. S. Ali, K. R. Mahmoud, and K. M. Naguib, "Optimal caching policy for wireless content delivery in D2D networks," *Journal of Network and Computer Applications*, vol. 150, Article ID 102467, 2020.

[16] K. K. Ahmed, *A Mobile Agent and Message Ferry Mechanism Based Routing for Delay Tolerant Network*, University Utara Malaysia, Changlun, Malaysia, 2018.

[17] C. Hu, H. Lin, Y. Hsu, S. Huang, L. Hui, and Z. Zhang, "Message forwarding with ferries in delay-tolerant networks," in *Proceedings of the 2019 28th Wireless and Optical Communications Conference (WOCC)*, pp. 1–5, Beijing, China, 2019.

[18] C. Diwaker, "Saini, shikha, "an enhanced cluster based movement model using multiple ferries nodes in VANET," *International Journal of Management, IT and Engineering*, vol. 6, no. 10, pp. 68–78, 2016.

[19] J. Li, H. Yan, Y. Zhang et al., "Efficient identity-based provable multi-copy data possession in multi-cloud storage," *IEEE Transactions on Cloud Computing*, p. 1, 2019.

[20] C. Peng, W. Li, Z. Wang et al., "All coverage and low-delay routing algorithm based on message ferry in opportunistic networks," *Application Research of Computers*, vol. 34, no. 03, pp. 819–823, 2017.

[21] X. WuC. Zhang et al., "Clustering routing protocol based on improved PSO algorithm in WSN," *Journal on Communications*, vol. 40, no. 12, pp. 114–123, 2019.

[22] J. Li, Y. Huang, Y. Wei et al., "Searchable Symmetric Encryption with Forward Search Privacy," *IEEE Transactions on Dependable & Secure Computing*, p. 1, 2019.

[23] Y. Huang, B. Li, Z. Liu, J. Li, and B. B. Gupta, "ThinORAM: towards practical oblivious data access in fog computing environment," *IEEE Transactions on Services Computing*, vol. 99, p. 1, 2019.

[24] H. Zhao-Long, L. Jian-Guo, and R. Zhuo-Ming, "Analysis of voluntary vaccination model based on the node degree information," *Acta Physica Sinica*, vol. 62, no. 21, pp. 218901–219590, 2013.

[25] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[26] K.-I. Goh, E. Oh, B. Kahng, and D. Kim, "Betweenness centrality correlation in social networks," *Physical Review E*

Statal Nonlinear & Soft Matter Physics, vol. 67, no. 1, Article ID 017101, 2003.

[27] M. Kitsak, L. K. Gallos, S. Havlin et al., "Identification of influential spreaders in complex networks," *Nature Physics*, vol. 6, no. 11, pp. 888–893, 2010.

[28] S. Yuping and N. Jing, "Effect of variable network clustering on the accuracy of node centrality," *Acta Physica Sinica*, vol. 65, no. 2, 2016.

[29] F. Zhang, *Research on Opportunistic Network Routing Algorithm Based on Cellular Learning Automata*, Shaanxi Normal University, Xi'an, China, 2016.

[30] S. Pal, "Evaluating the impact of network loads and message size on mobile opportunistic networks in challenged environments," *Journal of Network and Computer Applications*, vol. 81, pp. 47–58, 2017.

[31] Z. Hu, J. Liu, and Z. Ren, "Analysis of voluntary vaccination model based on the node degree information," *Acta Physica Sinica*, vol. 62, no. 21, pp. 512–517, 2013.

[32] Y. Song and J. Ni, "Effect of variable network clustering on the accuracy of node centrality," *Acta Physica Sinica*, vol. 65, no. 02, pp. 379–386, 2016.

[33] D. Chen, L. Lü, Y.-C. Zhang, and T. Zhou, "Identifying influential nodes in complex networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 391, no. 4, pp. 1777–1787, 2012.

[34] Z. Li, Z. Yang, and H. Wang, "An influence measure of nodes based on structures of social networks," *Acta Electronica Sinica*, vol. 44, no. 12, pp. 2967–2974, 2016.

[35] M. Riondato and E. M. Kornaropoulos, "Fast approximation of betweenness centrality through sampling," in *Proceedings Of the 7th ACM International Conference on Web Search and Data Mining*, pp. 413–422, New York, NY, USA, February 2014.

[36] Z. Zhang, Z. Y. Zhang, and M. Song, "Important node searching algorithm based on shortest-path betweenness," *Computer Engineering and Applications*, vol. 49, no. 21, pp. 98–100+132, 2013.

[37] H. Chen, G. Wang, P. Zhang et al., "Key nodes mining algorithm in sina weibo social network based on hadoop cloud platform," *Journal of Southeast University(Natural Science Edition)*, vol. 48, no. 4, pp. 590–595, 2018.

[38] R. S. Burt, M. Kilduff, and S. Tasselli, "Social network analysis: foundations and frontiers on advantage," *Annual Review of Psychology*, vol. 64, no. 1, pp. 527–547, 2013.

[39] H. Zhang and M. Zhang, "Node importance evaluation of communication network based on structural hole index," *Computing Technology and Automation*, vol. 35, no. 01, pp. 101–103, 2016.

[40] X. Su and Y. Song, "Leveraging neighborhood "structural holes" to identifying key spreaders in social networks," *Acta Physica Sinica*, vol. 64, no. 02, pp. 5–15, 2015.

[41] H. Yu, Liu, Z. Liu, and Y. J. Li, "Key nodes in complex networks identified by multi-attribute decision-making methon," *Acta Physica Sinica*, vol. 62, no. 02, pp. 54–62, 2013.

[42] G. Hu, H. Gao, and X. Xu, "Identify important nodes in complex network based on aggregation of multi-attribute preference information," *Journal of Zhejiang Sci-Tech University*, vol. 41, no. 4, pp. 482–488, 2019.

[43] X. Zhou, F. Zhang, K. Li et al., "Finding vital node by node importance evaluation matrix in complex networks," *Acta Physica Sinica*, vol. 61, no. 05, pp. 1–7, 2012.

[44] C. Zhu, X. Wang, and L. Zhu, "A novel method of evaluating key nodes in complex networks," *Chaos, Solitons & Fractals*, vol. 96, pp. 43–50, 2017.

[45] J. Shu, W. L. Jiang, and L. L Liu, "Critical nodes evaluation of opportunistic networks based on topological condensation graph," *Journal of Beijing University of Posts and Telecommunications*, vol. 42, no. 02, pp. 57–62, 2019.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 6679304, 11 pages
https://doi.org/10.1155/2020/6679304

Research Article

Defending Application Layer DDoS Attacks via Multidimensional Parallelotope

Xiaolin Zhao,[1] Hui Peng,[1] Xiang Li,[2] Yue Li,[1] Jingfeng Xue,[1] Yaoyuan Liang,[1] and Mingzhe Pei[1]

[1] *Beijing Institute of Technology, Beijing 100081, China*
[2] *National Key Laboratory of Science and Technology on Information System Security, Beijing 100101, China*

Correspondence should be addressed to Xiang Li; lixiangxts@163.com and Jingfeng Xue; xuejf@bit.edu.cn

Received 31 October 2020; Revised 17 November 2020; Accepted 18 December 2020; Published 30 December 2020

Academic Editor: Zhe-Li Liu

The Internet is more and more integrated into people's life; because of the complexity and fragility of the network environment, network attack presents a more and more serious trend. Application Layer DDoS (AL-DDoS) attack is the most complex form of DDoS attack, which is hindering the availability for the legitimate users by taking up a large number of requests of web server. The paper introduced the concept of behavior utility to portray the network. The concept of attack and defense utility was defined by a specific property which was the manifestation of the network risk after the offset of attack and defense. In the utility model, traffic metrics were mapped to the multidimensional parallelotope in the Euclidean space to express as a diagonal matrix. To determine the threshold status, the defense strategies of load balancing and limiting the maximum number of connections were used with different attack scales. Finally, the attack and defense utility value was calculated to evaluate the network risk level. The proposed method can master the capacity of network system against each attack means and the defense capability of network system. Its availability and accuracy are verified by comparing with the relevant works.

1. Introduction

With the rapid development of network technology, the field of network security is facing hacker attacks. The intensity of attacks is gradually increasing, and illegal attackers achieve improper goals. DDoS attacks are the main means. AL-DDoS attacks are different from traditional network layer DDoS attacks. It mainly uses existing protocol loopholes, such as HTTP and SMTP, and consumes existing network resources, so that the target server cannot provide conventional services. The intensity and accuracy of this attack are higher, and the threat to security is also greater. The number of attackers and the required attack traffic are much lower than traditional AL-DDoS attacks, which also means that AL-DDoS attacks are easier to launch, and attackers can accurately attack specific applications, so the attack threat is great.

The measurement of network system security by most of today's methods cannot reach the stage of quantitative calculation. Most security measurement methods rely on a certain technology while relying on the human experience of experts to measure whether it is safe. These methods are not accurate and objective enough, so people are always looking for a method that can quantitatively, dynamically, and objectively measure network security. Although the network is static, it is always changing. In order to measure attacks more accurately, a network security measurement method that can dynamically describe and warn attacks is needed. Boyer et al. [1] propose a network security evaluation framework based on D-S evidence theory, but this method has some problems such as large calculation. Ramaki et al. [2] propose a network security risk assessment method based on Bayesian network. Although this method has a strong capacity to process a large amount of data, it inevitably affected by some subjective factors, so the method must be properly trained to obtain relevant parameter. In the paper of Mukherjee et al. [3], a new security metric based on attack graph, namely, attack difficulty, has been proposed which includes the position factor. Wen et al. [4] propose a network security situation prediction method based on the hidden Markov model. The change rules and trend changes were analyzed by describing the dependence of security conditions in different periods. Wang et al. [5] propose an improved base metric algorithm based on dependency relationship graph and CVSS, aiming at the problem that the existing network security measurement based on CVSS could not accurately measure the probability and the impact of network attack at the same time.

In order to find a way that can dynamically measure network security and does not rely on expert experience, this paper attempts to find performance metrics that can describe AL-DDoS attacks. This paper analyzes the principles of attack and defense types, summarizes the characteristics of attack and defense targets, and proposes metrics for measurement. In order to evaluate the impact of AL-DDoS attacks, the parameters and calculation methods used in various technologies are analyzed to find better performance metrics to describe the impact of the attack. The main contributions are as follows:

(1) This paper puts forward the definition of AL-DDoS attack and defense utility combined with the definition of related concepts from the perspective of sociology and network. It is the first time that the concept of utility combines with attack and defense to measure network security.

(2) This paper selects 6 metrics and conducts a large number of simulation experiments combining type and intensity changes. The selected metrics can be used to do various experiments with different attack and defense effect.

(3) This paper proposes a calculation model that uses the concept of hyperparallel to construct multidimensional space for utility calculation. The model can accurately represent the impact of the attack and quantitatively determine the impact value of the attack and defense utility. This paper verifies the credibility and accuracy of the calculation model.

2. Related Research

2.1. DDOS Attack and Detection Technology. DDoS attacks evolved from DoS attacks and are divided into network layer DDoS attacks and AL-DDoS attacks. The traditional network layer DDoS attack means that hackers invade and control a large number of puppet machines through various loopholes and then use puppet machines to attack the target server. AL-DDoS attack refers to the attacker sending a large number of requests from the victim computer to the database to disable the server [6]. AL-DDoS attacks are usually divided into two types: flooding attacks and slow attacks [7].

Currently, DDoS attacks occur frequently, so it is necessary to detect attacks in time. AL-DDoS attacks generate a large amount of request data in a short period of time. This attack method is similar to the behavior of a large number of users suddenly and normally accessing. Therefore, this condition needs to be distinguished from normal access by a large number of users. The purpose of the AL-DDoS attack is to make the applications on the server unable to provide normal services to legitimate users and deny their access [8]. Intrusion Detection System (IDS) is one of the most effective detection and defense mechanisms for DDoS attacks [9]. IDS is an application-type system that monitors suspicious events in the network, generates reports, and forwards them to administrators for action. There are many traditional IDS/IPS technologies, such as feature-based detection methods and abnormal behavior-based detection methods [10].

2.2. AL-DDOS Attack Assessment Method. Among current researches on DDoS attacks, most of them are based on the network layer, but there are few researches on the impact of AL-DDoS attacks [11]. The paper of Pallavi et al. [12] finds that over 45% of these applications do not implement measures to protect BLE data, and that cryptography is sometimes applied incorrectly in those that do. Application layer data is extremely vulnerable. The paper of Wei Zhou et al. [13] finds that smart home devices are vulnerable to attacks by network traffic interception. While bringing unprecedented convenience and accessibility, they also introduce various security hazards to users.

Kumar et al. [14] propose a method to measure the impact of AL-DDoS attacks on web server performance. The authors modify the Webtraf module in NS-2 to generate attack traffic to simulate legitimate user behavior. They analyze the impact of different server processing strategies and queue lengths on the attack. In this method, Wang et al. [15] have developed a prototype of SkyShield and evaluated its effectiveness using real attack data collected from large web clusters. Experimental results show that SkyShield can quickly reduce malicious requests while having limited impact on legitimate users. In the method of Sahoo et al. [16], an information distance-based flow discriminator framework has been discussed, which can discriminate DDoS traffic during flash events in a software-defined network (SDN) environment, that is, legitimate traffic that looks similar. The information distance metric is used to describe the variations of traffic behavior of such events. The simulation results show that the information distance metric can effectively identify the DDoS traffic [17]. The paper of Jiahao Cao et al. [18] systematically studies the impacts of attack on various network applications in a real SDN test bed. Experiments show that the attack significantly degrades the performance of existing network applications and causes serious network anomalies, e.g., routing black hole, flow table resetting, and even network-wide DoS.

Procopiou et al. [19] propose ForChaos, a lightweight detection algorithm for IoT devices, which is based on forecasting and chaos theory to identify flooding and DDoS attacks. In NS-3, the detection algorithm is evaluated through a series of experiments in flooding and slow-rate DDoS attacks. Sardana et al. [20] propose an integrated honeypot framework for active detection, characterization, and redirection of DDoS attacks at the ISP level. The authors evaluate the impact of DDoS flood attack on effective throughput, average transaction failure interval time, and average response time as parameters under different operation modes and use the framework to defend against high-rate DDoS attack by referring to impact value.

Most of the relevant studies rely on subjective empirical judgment and lack of objectivity and use fewer metrics, so the conclusions may be biased. In this paper, a measurement method of AL-DDoS attack and defense utility is proposed to calculate the effects of AL-DDoS attack. By comparing the simulation experiment data with the related technical data, the effectiveness, objectivity, and accuracy of the method are verified.

3. AL-DDOS Attack and Defense Utility and Calculation Model

Attack and defense behaviors are defined in the network: from the perspective of network objects and their interconnection, attack or defense behavior refers to a series of state changes caused by attacks or defense methods in the network. The network status change caused by the attack process can be described as an attack behavior. The weakening of the attack effect caused by the defensive means leading to the change of the network state can be defined as defensive behavior. The attack behavior is composed of five basic elements, among which the behavior subject is the attack initiator, that is, the hacker. The object of behavior is the target of attack, namely, the network system. The behavior environment is the network environment where the attack process is located. The behavior means are the resources used in the network when the attack is launched. The behavior result is the agreement degree between the expected attack result and the actual attack result during the attack. By analogy, the basic elements of defensive behavior are as follows. The behavior subject is the defensive measure. The behavior object is the target of defense, that is, the source of attack. The behavior environment refers to the network environment in the process of defense. The behavior means is the resource in the network when the defense measures are launched. The behavior result is the agreement degree between the expected defense result and the actual defense result after the attack process [21].

Based on the analysis of sociology and network behavior, attack and defense behavior have the following features: (1) Purpose: the occurrence of attack and defense behavior must be accompanied by purpose, in which the attack behavior will not affect the network for no reason, and the defense behavior will not work without being attacked. (2) Persistence: attack and defense behaviors each point to their own goals. Generally, attack and defense behavior will not terminate until the goal is completed. When attack and defense behaviors are in effect, they may change their behaviors due to the difficulty of achieving the goal, but attack and defense behaviors are persistent. (3) Variability: the mode of attack and defense behaviors may be gradually optimized with the continuous update of technology, and new technology may be used to achieve their goals.

Attack and defense behavior impact refers to the network status change caused by attack and defense behavior through a series of operations. Before a multitude of attack and defense behaviors function to the network system, the network status value needs to be set to S. The network status value after attack and defense behavior is set to S', and the status value rate is used to calculate comprehensive attack and defense forces $F_{\mathrm{AD}} = S'/S$. Attack and defense behavior is shown in Figure 1.

As is shown in Figure 2, the essence of network system security is a balance between attack and defense. In a specific network scene, network attack and defense are regarded as behaviors, which can establish network system security judgments and identification of the behavior utility standards.

The definition of attack and defense utility is as follows: Suppose the attack function is A, the defense function is D, and the combined attack and defense force is $F_{\mathrm{AD}} = A - D$ (if F is greater than 0, it indicates that the defense function cannot resist the attack; if F is less than or equal to 0, the defense function can resist the current scale of attack).

After calculating the attack and defense force, the attack and defense utility is the sum of the effects caused by the attack and defense forces, which is used to represent the comprehensive effect of the network system after the combination of attack and defense in the process [22]. The attack and defense utility is shown in Figure 3.

By calculating the utility value of the attack and defense behavior of the network system, the effects of various AL-DDoS attack and defense methods can be accurately and objectively obtained. The calculated value of attack and defense utility can provide a more complete and objective evaluation standard for researchers in network security measurement field. The mathematical methods of evaluating attack and defense behavior are more objective and easy to compare with other methods.

3.1. Attack and Defense Utility Calculation Model.

When an attack occurs, the relevant metric status of the network will change. No single network metric can completely represent an attack. Therefore, the network status can be obtained by combining the metric values according to the changes of various metrics in the network during the attack.

In this method, the metrics in the network are taken as dimensions in the n-dimensional space, and each metric corresponds to a vector in the space. The change of the metric means that the length of the vector will change.

The AL-DDoS attack utility calculation model is divided into three steps: the first step is to determine the corresponding metric items according to the attack and defense features and map each metric to the n-dimensional parallelotope to calculate the parallel volume. The second step is to obtain the status value of the network system by the parallel volume. The third step is to calculate the change rate of the parallel volume according to the selection of threshold value and the calculation method of attack and defense utility and then compare the change rate with the threshold value to obtain the attack and defense utility value. Finally, the attack and defense utility value during the attack is obtained by calculating the average attack and defense value.

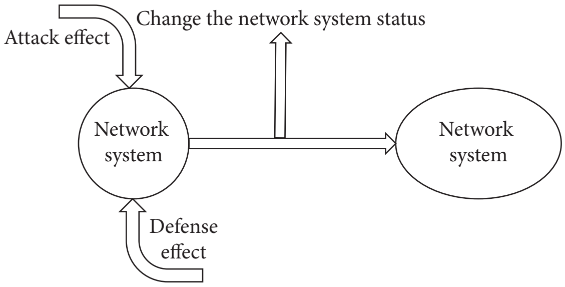


Figure 1: Attack and defense behavior effect in network system.

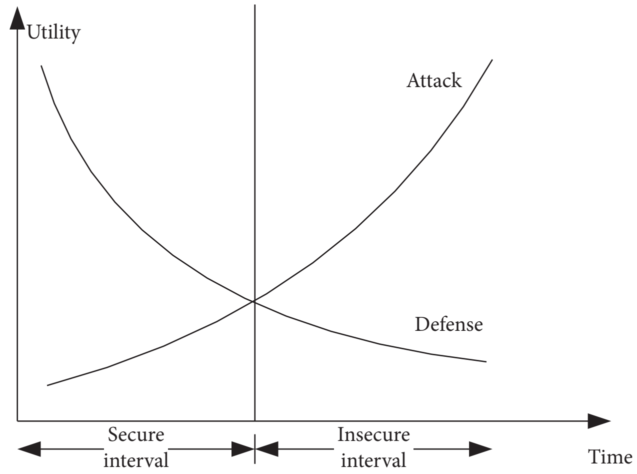


Figure 2: Utility criteria for network system security.

In this calculation method, the attack and defense effect should be compared with the threshold. The calculation method of threshold set is the calculation result of node status value by measuring the compound metric value without attack and defense effect. According to the average record, after an AL-DDoS attack, the average time for the defender to detect the attack is 1–5 hours. It is recommended to collect far more than the operation status of the server within 5 hours. Through the analysis of server data records, network analysis, application services, and other relevant indicators, the status value within 1–5 hours that is the most stable data fluctuation and relatively consistent with the legal access behavior track is selected as the threshold.

In the simulation, the attack strength is preset to 1, and the defense base strength is preset to 0. The current attack method and the effect of attack strength are obtained by calculating the change rate of the node status value caused by the attack effect. Under the condition that the initial value of node status is set to 0, that is, without adding defense measures, the attack effect is equal to the node status value. The calculation method of the defense effect can be derived in this way. Based on the variability of attack and defense behaviors, in order to ensure the universality of the calculation results of attack and defense effects, it is necessary to obtain the average attack effect within a certain period of time after the change of attack intensity is obtained by analyzing attack effect at each moment. In this way, the peak and minimum values of the attack effect are eliminated to fully describe the attack effect.

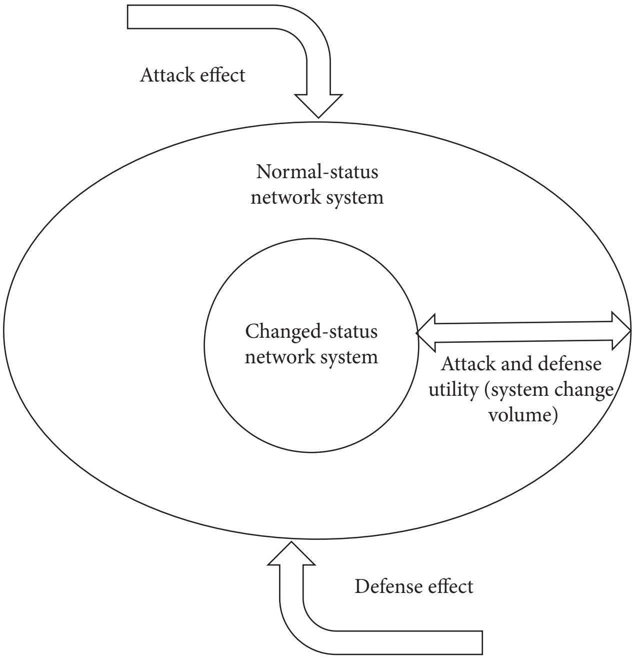

Attack effect
Normal-status network system
Changed-status network system
Attack and defense utility (system change volume)
Defense effect

Figure 3: Attack and defense utility in attack and defense process.

On the premise that the time span remains unchanged, the calculation method of attack and defense is shown in Table 1. Table 1 lists the change value of the attack effect caused by the change of the attack strength and the change amount of the defense effect caused by the accumulation of defense measures. The data in the table are abstract values.

The attack and defense utility value can be used to describe the total attack on the network system during the attack and the total defense capability during the attack and defense. In AL-DDoS attacks, the attack traffic is not constant under normal circumstances, and the influence of network traffic is limited by many factors. Therefore, the expression of the attack effect can obtain the attack size at a certain point, but it cannot reflect the impact of the attack in the entire attack process.

On the basis of threshold value, the size of the attack effect at each moment can be obtained, and then the attack utility can be used to illustrate the impact value caused by the attack type during the entire attack process. In other words, the utility is the sum of the effects and the cumulative amount of the attack effect changing with time. If the attack effect size is F, when setting and selecting the threshold, the attack effect is 0, and the attack effect is also 0. If an attack occurs, set t_1 attack effect to be F_1, t_2 attack effect to be F_2, and t_n attack effect to be F_n. Based on the threshold setting, the calculation method of attack utility E is

$$E = \sum_{F_{t1}}^{F_{\mathrm{tn}}} F. \qquad (1)$$

The attack scale and intensity may change during the attack, but when the defense measures are attacked, it is

Table 1: Attack and defense parameter simulation calculation.

Time span	Attack strength	Defense strength	Node status	Attack effect	Defense effect
1	1	0	1	1	0
1	1	1	0	1	1
1	2	0	2	2	0
1	2	1	1	2	1

necessary to know whether the defense effect is constant. The defense effect can be obtained by comparing different attack effects. In this case, the average attack force will be calculated.

3.2. Utility Calculation Metric Selection. It is important to select appropriate metric to measure and analyze the impact value of attacks. As for the selection of metrics, the existing metrics when calculating the impact of the current DDoS attack are as follows. Sardana et al. [20] select effective network throughput, average access failure time and average response time as metrics in the measurement method. Dantas et al. [23] use the volume of traffic as the measurement metric. In the calculation method of influence value based on user service quality, the selected metrics are successful transaction rate, average response time, number of connections, average service rate, and request rate.

Comprehensive consideration of network traffic, hardware performance and other related indicators will be more suitable for analyzing the impact of DDoS attacks. 6 metrics were selected using the proposed metric selection [24].

(1) Network throughput rate: network throughput rate is used to describe the total number of data packets received and sent by the network card in the server during the attack. These data packets not only include the packets generated by normal user access, but also calculate the packets generated by attackers.

(2) TCP data segment transmission rate: AL-DDoS attacks may take advantage of the three-handshake mechanism of the TCP protocol to attack the server. The number of TCP segments is used to describe the number of TCP segments at any time during the attack.

(3) IP datagram transmission rate: in attack detection technology, IP datagrams are usually used to analyze the structure of datagram to detect the occurrence of attacks. Because of the nature of the TCP/IP protocol, it is essential to analyze the TCP data segment transmission rate and the IP datagram transmission rate.

(4) Transaction failure rate: when there are a large number of attacker requests in the server, the successful server accessing rate will be greatly affected due to bandwidth, server performance, and other factors. The failure rate is the ratio of the number of failed accesses to the total number of accesses at any time. According to the purposes and characteristics of AL-DDoS attacks, the access failure rate is the most important metric to detect attack.

(5) Average traffic arrival time: the average traffic arrival time refers to the time it takes to successfully access the server. When an AL-DDoS attack occurs, the average server response time must increase. As the attack strength increases, access timeouts may occur.

(6) Server CPU utilization: when an attack sends a large number of high-frequency service requests to the target server, the server will be busy providing response resources to the attacker, and the occupation of resources will inevitably affect the performance of the server hardware.

3.3. Network State Value Calculation. Given the combination of the current attack size and defense effect, the server status value represents the server status at each moment. In the calculation process, the average value of server status at each moment in the entire attack and defense process is selected as the calculation result [25]. The specific calculation steps are as follows.

The first step is to construct an $\mathbf{n}$-dimensional matrix. In this calculation model, the values of the six metrics are, respectively, set as $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4, \mathbf{m}_5, \mathbf{m}_6$. As mentioned above, each metric is a linearly independent vector in the $\mathbf{n}$-dimensional Euclidean space $\mathbf{V}$. Therefore, six metrics are selected to map to the vector dimension in the six-dimensional space, and the vectors are expressed as

$$\begin{bmatrix} \mathbf{m}1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{m}2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \mathbf{m}3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{m}4 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{m}5 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{m}6 \end{bmatrix}. \tag{2}$$

Therefore, the 6-dimensional matrix composed of these six vectors can be expressed as a diagonal matrix, namely,

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}1 & 0 & 0 \\ 0 & \ldots & 0 \\ 0 & 0 & \mathbf{m}6 \end{bmatrix}. \tag{3}$$

As is shown in Figure 4, this is a 4-dimensional parallelotope. The second step is to calculate the volume of parallelotope in 6-dimensional space. The volume of parallelotope composed of six vectors in six-dimensional Euclidean space is as follows:

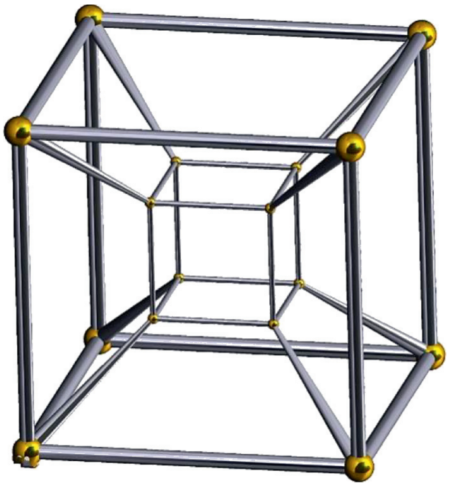


Figure 4: 4-dimensional parallelotope graphic model.

$$\mathbf{V}(\mathbf{m}_1, \mathbf{m}_2, \dots \mathbf{m}_6) = \mathbf{V}(\mathbf{m}_1, \mathbf{m}_2, \dots \mathbf{m}_5) \times \mathbf{g}\mathbf{h}_6, \qquad (4)$$

where $\mathbf{h}_6$ represents the length of the orthogonal component of $\mathbf{m}_6$ in the subspace generated by vectors $\mathbf{m}_1, \mathbf{m}_2, \dots \mathbf{m}_6$, and $\mathbf{V}_1(\alpha_1) = |\alpha_1|$. It has been proved that$\mathbf{V}(\mathbf{m}_1, \mathbf{m}_2, \dots \mathbf{m}_6) = \sqrt{\mathbf{G}(\mathbf{m}_1, \mathbf{m}_2, \dots \mathbf{m}_6)} = |\mathbf{D}|$, where $\mathbf{D}$ is the determinant of coordinates on a set of standard bases of $(\mathbf{m}_1, \mathbf{m}_2, \dots \mathbf{m}_6)$.

Therefore, the network status value at a certain time $\mathbf{t}$ is the determinant of the diagonal matrix $\mathbf{M}$. Set the network status value to $\mathbf{S}_\mathbf{t}$, and the specific calculation method is

$$\mathbf{S}_\mathbf{t} = |\mathbf{M}|. \qquad (5)$$

The third step is to calculate the arithmetic average of the network status during the attack. During the duration of attack and defense effect of the network system, the network status value at each moment is, respectively, $\mathbf{S}_{\mathbf{t}_1}, \mathbf{S}_{\mathbf{t}_2}, \mathbf{S}_{\mathbf{t}_3}, \dots, \mathbf{S}_{\mathbf{t}_\mathbf{n}}$. Therefore, the arithmetic average of the network status is as follows:

$$\mathbf{S} = \frac{\sum_{\mathbf{i}=1}^{\mathbf{n}} \mathbf{S}_{\mathbf{t}_\mathbf{i}}}{\mathbf{n}}. \qquad (6)$$

3.4. Effect Calculation. According to the calculation model of the status value, a part of the normal operation status of the network is selected and the status value $\mathbf{S}_0$ is calculated. When an attack effect occurs, the change of network status value can be described by the attack effect. When the attack strength is $\mathbf{X}$, the network status value is $\mathbf{S}_\mathbf{i}$ at time $\mathbf{t}_\mathbf{i}$. Then, the attack effect $\mathbf{A}_{\mathbf{Xt}_\mathbf{i}}$ is calculated as follows:

$$\mathbf{A}_{\mathbf{Xt}_\mathbf{i}} = \frac{\mathbf{S}_\mathbf{i}}{\mathbf{S}_0}, \quad \mathbf{i} = (1, 2, \dots, \mathbf{n}). \qquad (7)$$

The attack effect value is averaged to reasonably describe the attack force of attack type during the attack. The calculation method of average attack force is as follows:

$$\overline{A} = \frac{\sum_{\mathbf{i}=1}^{\mathbf{n}} \mathbf{A}_{\mathbf{Xt}_\mathbf{i}}}{\mathbf{n}} = \frac{\sum_{\mathbf{i}=1}^{\mathbf{n}} \mathbf{S}_{1_\mathbf{i}}/\mathbf{n}}{\sum_{\mathbf{i}=1}^{\mathbf{n}} \mathbf{S}_{0_\mathbf{i}}/\mathbf{n}}. \qquad (8)$$

There is a defense effect before the attack, but it is impossible to measure the defense effect. When the status value of the attacked system is obtained under the defense status of 0, the defense function is gradually added under the premise that the attack effect remains unchanged, and the defense effect is obtained through the system status value when each defense effect occurs.

If the attack intensity is X with no defense measures, the network status value is S_1, then the attack effect at time t_1 can be set to A_{Xt_1}. If defense measures are added to the system at this time, the network status value is S_1, the attack effect is changed to A_{Xt_1}, and the change value of attack effect is denoted as $\Delta A = A_{Xt_1} - A'_{Xt_1}$. According to the formula, the calculation method of defense effect at time t_1 is expressed as

$$\mathbf{D}_{\mathbf{t}_\mathbf{i}} = \frac{\mathbf{S}_\mathbf{i} - \mathbf{S}'_\mathbf{i}}{\mathbf{S}_0}, \quad \mathbf{i} = (1, 2, \dots, \mathbf{n}). \qquad (9)$$

The calculation method of the average defense force is as follows:

$$\overline{\mathbf{D}} = \frac{\sum_{i=1}^{n} \mathbf{S}_{\mathbf{Xt}_\mathbf{i}}}{\mathbf{n}}, \quad \mathbf{i} = (1, 2, \dots, \mathbf{n}). \qquad (10)$$

3.5. Utility Calculation. The attack and defense utility value represents the cumulative value of attack and defense effects. In the process of attack and defense, the role of attack and defense means changes at any time in practice. Therefore, in the evaluation process, it is necessary to obtain the utility value of two kinds of effects in the whole process; that is, the utility value can represent the total amount of the attack and defense effects in the process.

According to the calculation results obtained in (7) and (9), if the attack duration is t, the attack intensity is X, and the defense intensity is Y, and the average attack force is denoted as $\overline{A_X}$, and the average defense force is $\overline{D_X}$, then the calculation methods of attack utility E_A and defense utility E_D are as follows:

$$\mathbf{E}_\mathbf{A} = \overline{\mathbf{A}_\mathbf{X}} \cdot \mathbf{t}, \qquad (11)$$

$$\mathbf{E}_\mathbf{D} = \overline{\mathbf{D}_\mathbf{X}} \cdot \mathbf{t}. \qquad (12)$$

According to the calculation result of the attack and defense utility value, it can be concluded that the defense measures can resist a certain attack effect, and then the concept of defense efficiency is proposed. According to (11), when the defense is 0, it can be concluded that the attack utility value is E_A at the certain time t. After adding defense measures to the current system, the attack utility is E_A. Then, the calculation method of defense efficiency D_E is

$$\mathbf{D_E} = \frac{\mathbf{E_D}}{\mathbf{E_A}} \times 100\%. \qquad (13)$$

4. Experiment Design and Analysis

4.1. Experiment Design. According to the general attack methods and the application range of defense measures in the AL-DDoS attack types, HTTP/POST attack was selected as the attack type in the simulated attack and defense experiment, and load balancing and limiting the maximum number of connections were adopted as defense measures. Under the premise of using the same attack threads, 10, 20, and 30 attack nodes were selected to gradually increase the attack intensity. The program was designed for 10 combinations of attack and defense. The network topology of the attack and defense experiment is shown in Figure 5.

The environment configuration is shown in Tables 2 and 3.

The specific implementation process is as follows. The first step is to set the combination of attack and defense scale. The specific design of 10 scales was as follows: without attack and defense effect, and 10, 20, and 30 attack nodes, respectively, in three defense modes: no defense, load balancing, and limiting the maximum number of connections.

The second step is to collect experiment data. Performance monitoring tools JMeter and Spotlight are used to collect experiment data. According to the level of detail and accuracy of the tool, the experiment data were collected separately. In this simulated experiment, JMeter was used to collect three metrics, namely, server failure rate, average server access time, and peak server access traffic, while Spotlight was mainly responsible for collecting four metrics, namely, packet volume, TCP segment number, IP datagram number, and server CPU utilization.

The third step is to analyze the experiment results. Through the proposed calculation model, the data collected by simulation experiment of attack and defense were calculated and the calculated results were obtained. The calculation results were compared with the existing technical results, and the rationality and correctness of the calculation results were analyzed.

It is aimed at calculating various utility values of attack and defense after the occurrence of AL-DDoS attack, so the main purpose of constructing the experiment environment is to simulate the actual AL-DDoS attack. In order to better simulate distributed attacks, 30 attack nodes were constructed in the simulation experiment, and one of the small servers deployed web applications as the target. The environment was divided into three subnets by IP, which were connected by switches, respectively, and the network topology is a star structure.

4.2. Experiment Calculation Result

4.2.1. Effect Calculation Result. According to the experiment steps, two defense methods were selected for comparison experiments, namely, load balancing and limiting the maximum number of database connections. Because load balancing defense refers to reduce server pressure through distributed deployment, the average defense effect should be calculated to evaluate its defense effect. As the attack intensity changes, the average of HTTP/POST attack effects on the network system of the current experiment is shown in Table 4.

When the number of load balancing distributed deployment machines is constant, the effect of HTTP/POST attack on the network system changes with the attack intensity. Particularly, the average defense effect of load balancing and limiting the maximum number of connections is shown in Table 4 with the change of attack intensity.

4.2.2. Utility Calculation Result. According to the concept that the attack and defense utility is defined as 0 under the threshold status, the utility value and attack and defense efficiency under each combination can be calculated by using the calculation model of attack and defense utility. And the attack utility value under each attack intensity can be calculated.

Since at least 60 attack and defense data were continuously collected in this attack experiment, the time was set to 60.

4.3. Experiment Analysis. The defense efficiency of load balancing and limiting the maximum number of connections in the combination of HTTP/POST attack intensity are shown in Figure 6.

According to Figure 6, when the attack intensity is small, the defense effect of load balancing is larger, and the defense effect of limiting the maximum number of connections is relatively negligible. As the attack strength increases, the defense utility of limiting the maximum number of connections increases exponentially, far exceeding the defense utility of load balancing. This shows that when the attack intensity is small, the load balancing defense effect is good, and limiting the maximum number of connections can resist large-scale DDoS attacks.

As is shown in Figures 6 and 7, when the attack intensity is 10 nodes, the maximum number of connections obviously exceeds the number of connections sent by the attacker, and the defense effect is close to 0. When the attack intensity is 20 nodes, it can be seen that the set maximum number of connections can resist some attacks. When the attack intensity is 30 attack nodes, it can be observed that the system can almost completely resist the current scale attacks. This also raises the question of how to set the number of connections. If the limit is set too high, the system may not be able to resist the current strength of the attack. If the limit is set too small, it will easily lead to congestion or overflow and other abnormal phenomena, affecting the normal operation of the system. Therefore, we can judge how to set the maximum number of connections based on the existing

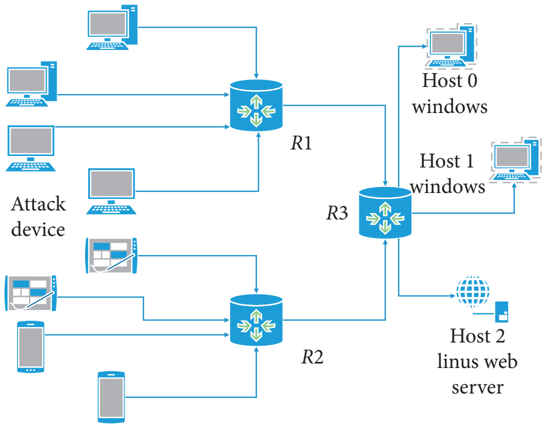


Figure 5: Attack and defense experiment network topology.

Table 2: Hardware and software environment configuration of attack nodes.

Device/software	Performance/function
ROM	2 GB
Processor	1
RAM	60 GB
Windows, Ubuntu operating system	Install test software and perform access service
Kali attack tool	Simulate various attacks and penetration tests
Wireshark	Capture packets and get messages
Burp Suite	Intercept log messages
LOIC	Package tool
JMeter	Website stress test tool
VMware Workstation	Simulate attack nodes

Table 3: Hardware and software environment configuration of target drone.

Device/software	Performance/function
ROM	4 GB
Processor	8
RAM	500 GB
Spotlight on Windows	Monitor server performance indicators
bWAPP	Target drone used for attack experiments
Java, php + MySQL + Apache	Build a web server environment
VMware Workstation	Build a virtual environment and software load balancing

experiment data and analyze the calculation results of this experiment.

4.4. Experiment Comparison. Through the study of the current AL-DDoS attack effect evaluation technology, a method is selected that Mirkovic [26] puts forward for the use of the user's quality of service (QoS) as a measure model. In the comparison technique, the author chooses the transaction failure rate as a measure of the QoS of various services. The author puts forward the method of calculating the amount of customer QoS degradation.

$$N = \frac{(d - t)}{t}. \tag{14}$$

N is the QoS degradation, t is threshold, and d is the value greater than the threshold value.

The value of QoS degradation N means that the service of transaction failure is N times the service that the user can tolerate. As calculated by the experiment data, the transaction threshold of failure rate is 0.075%. In order to show clearly, the attack utility is reduced by 10^{14} times in the figure.

As is shown in Figure 8, the utility method is more effective to determine the impact of an attack. The calculation model comprehensively considers the metric values of the impact of various AL-DDoS attack on the network system. Therefore, this method can obtain more comprehensive, accurate, and reasonable results compared with the existing methods.

Table 4: Average attack effect and defense effect.

	HTTP/POST attack	Load balancing	Limiting connections
10 nodes	4.651×10^{13}	2.895×10^{13}	0.031×10^{13}
20 nodes	1.076×10^{15}	6.951×10^{14}	7.521×10^{14}
30 nodes	1.437×10^{16}	7.714×10^{15}	1.405×10^{16}

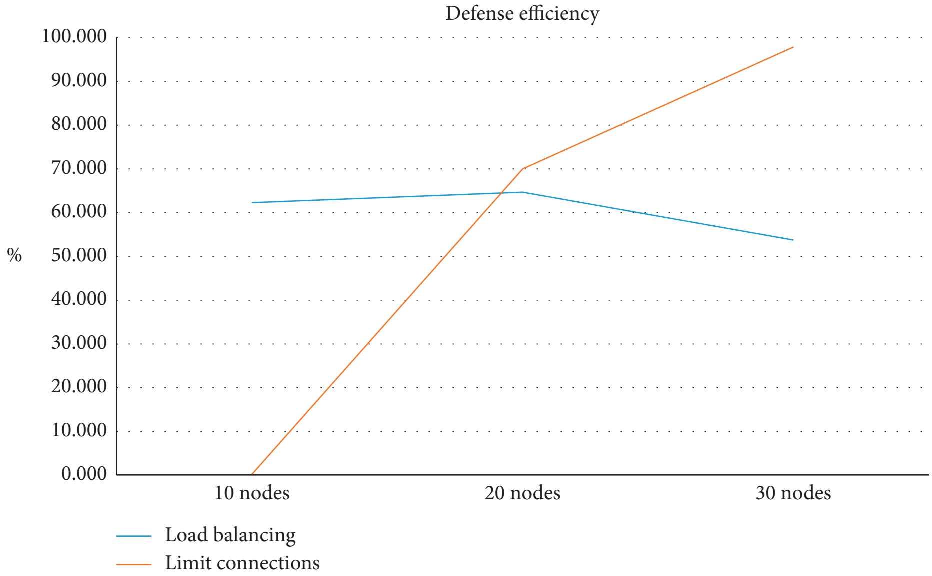


Figure 6: Defense efficiency based on the load balancing and limiting the maximum number of connections.

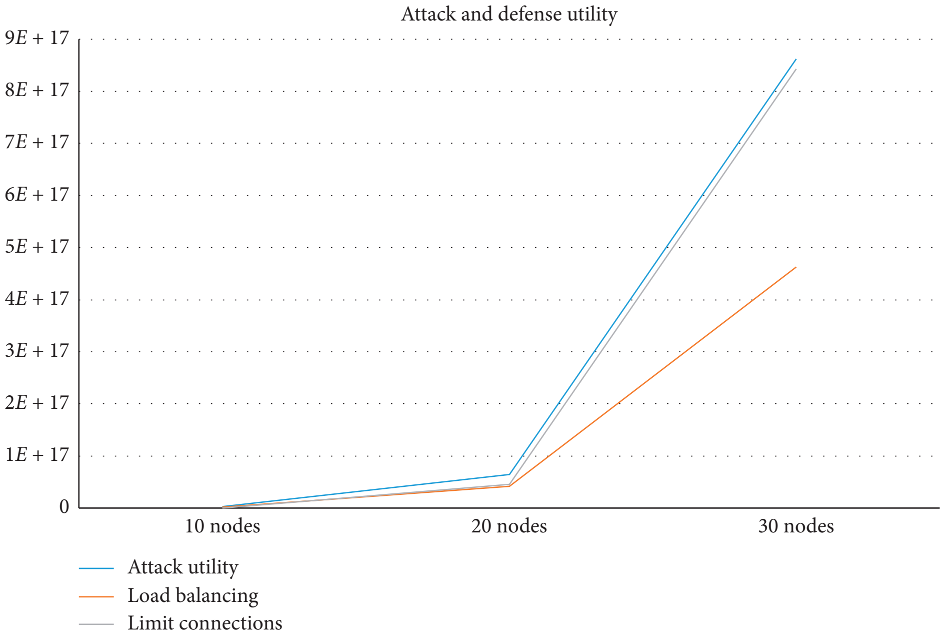


Figure 7: Attack and defense utility based on the load balancing and limiting the maximum number of connections.

By analyzing the attack and defense measures used in this experiment, we can obtain the attack utility value caused to the application layer of the current network under different attack intensity and calculate the defense utility value of each defense method under different attack intensity. The proposed attack and defense utility measurement method can quickly obtain the attack and defense utility value, without manual calculation and judgment, and can

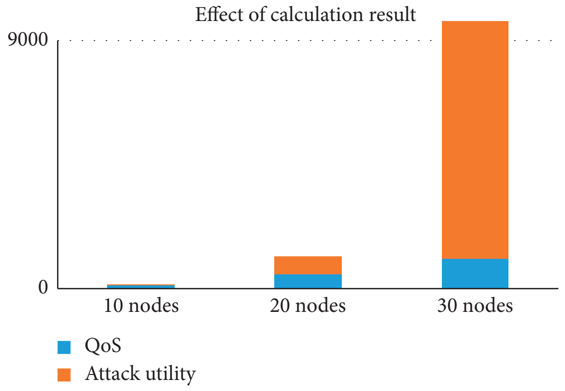


Figure 8: Influence of different attack strengths on calculation results under the same attack technology and defense strength.

accurately obtain the attack and defense impact value. This method has certain accuracy and objectivity and can judge the impact of attack and defense and the occurrence of potential attacks in real time.

5. Conclusion

Some factors are analyzed such as the characteristics of AL-DDoS attack and defense utility and the selection method of the existing indicators, and then the 6 metrics are selected. The attack and defense impact on the network itself and the impact on users are both considered, so as to accurately and objectively describe the network attack and defense behavior. For various network attacks, the metrics may be different, but traffic attacks also can use these metrics. Different models can be compared, so the method is objective.

Existing evaluation methods used to describe the measure results of the proposed concept lack theoretical support, while the concept of utility is used to describe the attack effect value. The concept of attack and defense utility is put forward by analyzing the characteristics of network behavior and combining traditional theory. The theoretical support makes the utility more reliable. At the same time, it is proposed to use hyperparallel volume to describe the network status. Based on this method, any number of indicators can be combined and calculated. Hyperparallel is used to map network space, which is an innovative attempt to network structure and interactive mathematical modeling.

The measurement is based on some certain information such as traffic, so the conclusion is objective. For example, the role and defense efficiency of defense technology in the attack can be analyzed separately, the attack effect on the network at a certain time can be analyzed, the status value that can be used to describe the network can be obtained, the attack and defense efficiency value can be calculated, and finally the attack and defense utility value can be calculated.

This method is more objective and effective than traditional methods. This method has smaller deviation and higher accuracy compared with single or few index methods. This method also provides a reference for various attack and defense methods. The use of mathematical methods to evaluate attack and defense behavior is more objective and easier to compare. More attack and defense methods can be used to conduct combined experiments on network systems for security evaluation against AL-DDoS attacks, so as to verify the utility of other attack and defense methods on the target network.

These 6 metrics are selected only for AL-DDoS attacks. These metrics are mainly suitable for traffic attacks. For other types of attacks, the calculation model is suitable but the metrics may be different. In the experiment design, the attack strength change. Attack and defense types are limited. There are only three changes in attack strength. The subdivision of the experiment is not enough.

In future work, we may try other different types of datasets, change other metrics, and conduct attack and defense experiments. Attack and defense experiments with smaller gradient changes and more types will be designed.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2016QY06X1205.

References

[1] S. Boyer, O. Dain, and R. Cunningham, “Stellar: a fusion system for scenario construction and security risk assessment,” in *Proceedings of the Third IEEE International Workshop on Information Assurance*, March 2005.

[2] A. A. Ramaki, M. Khosravi-Farmad, and A. G. Bafghi, “Real time alert correlation and prediction using Bayesian networks,” in *Proceedings of the 2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, Rasht, Iran, Sepetember 2016.

[3] P. Mukherjee and C. Mazumdar, “Attack difficulty metric for assessment of network security,” in *Proceedings of the ARES 2018*, Hamburg, Germany, August 2018.

[4] Z. Wen, C. Cao, and H. Zhou, “Network security situation assessment method based on naive Bayes classifier,” *Journal of Computer Applications*, vol. 35, no. 8, pp. 2164–2168, 2015.

[5] J. X. Wang, Y. Feng, and R. You, “Network security measurement based on dependency relationship graph and common vulnerability scoring system,” *Journal of Computer Applications*, vol. 39, no. 6, pp. 1719–1727, 2019.

[6] M. T. Manavi, “Defense mechanisms against distributed denial of service attacks: a survey,” *Computers & Electrical Engineering*, vol. 72, pp. 26–38, 2018.

[7] H. Luo, Y. Lin, H. Zhang, and M. Zukerman, “Preventing DDoS attacks by identifier/locator separation,” *IEEE Network*, vol. 27, no. 6, pp. 60–65, 2013.

[8] X. Ma and Y. Chen, "DDoS detection method based on chaos analysis of network traffic entropy," *IEEE Communications Letters*, vol. 18, no. 1, pp. 114–117, 2014.

[9] S. Bravo and D. Mauricio, "DDoS attack detection mechanism in the application layer using user features," in *Proceedings of the 2018 International Conference on Information and Computer Technologies ICICT*, March 2018.

[10] Y. J. Li, B. Y. Liu, S. Zhai, and M. R. Chen, "DDoS attack detection method based on feature extraction of deep belief network," *IOP Conference Series Earth and Environmental Science*, vol. 252, Article ID 032013, 2019.

[11] A. Praseed and P. S. Thilagam, "DDoS attacks at the application layer: challenges and Research perspectives for safeguarding web applications," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 661–685, 2019.

[12] P. Sivakumaran and J. Blasco, "A study of the feasibility of co-located app attacks against BLE and a large-scale analysis of the current application-layer security landscape," in *Proceedings of the 28th USENIX Security Symposium*, pp. 1–18, Santa Clara, CA, USA, August 2019.

[13] W. Zhou, Y. Jia, Y. Yao et al., "Discovering and understanding the security hazards in the interactions between IoT devices, mobile apps, and clouds on smart home platforms," in *Proceedings of the 28th USENIX Security Symposium*, pp. 1133–1150, Santa Clara, CA, USA, August 2019.

[14] M. Kumar and A. Bhandari, "Performance evaluation of web server's request queue against AL-DDoS attacks in NS-2," *International Journal of Information Security and Privacy*, vol. 11, no. 4, pp. 29–46, 2017.

[15] C. Wang, T. T. N. Miu, X. Luo, and J. Wang, "SkyShield: a sketch-based defense system against application layer DDoS attacks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 559–573, 2018.

[16] K. S. Sahoo, S. K. Panda, S. Sahoo, B. Sahoo, and R. Dash, "Toward secure software-defined networks against distributed denial of service attack," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 4829–4874, 2019.

[17] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602–622, 2016.

[18] J. Cao, Q. Li, R. Xie et al., "The crossPath attack: disrupting the SDN control channel via shared links," in *Proceedings of the 28th USENIX Security Symposium*, pp. 19–36, Santa Clara, CA, USA, August 2019.

[19] A. Procopiou, N. Komninos, and C. Douligeris, "ForChaos: real time application DDoS detection using forecasting and chaos theory in smart home IoT network," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 8469410, 14 pages, 2019.

[20] A. Sardana and R. C. Joshi, "An integrated honeypot framework for proactive detection, characterization and redirection of DDoS attacks at ISP level," *Journal of Information Assurance and Security*, vol. 1, pp. 1–15, 2018.

[21] K. Tang and X. Zhang, "Construction of human subject: expansion and limitation of the perspective of network sociology," *Social Sciences in Hunan*, vol. 5, pp. 67–74, 2017.

[22] C. Hu, "Calculation of the behavior utility of a network system: conception and principle," *Engineering*, vol. 4, no. 1, pp. 171–185, 2018.

[23] Y. G. Dantas, V. Nigam, and I. E. Fonseca, "A selective defense for application layer DDoS attacks," in *Proceedings of the 2014 IEEE Joint Intelligence and Security Informatics Conference*, The Hague, Netherlands, September 2014.

[24] K. Singh, P. Singh, and K. Kumar, "User behavior analytics-based classification of application layer HTTP-GET flood attacks," *Journal of Network and Computer Applications*, vol. 112, pp. 97–114, 2018.

[25] X. Zhao, Q. Chen, J. Xue, Y. Zhang, and J. Zhao, "A method for calculating network system security risk based on a lie group," *IEEE Access*, vol. 7, pp. 70610–70623, 2019.

[26] J. Mirkovic, A. Hussain, B. Wilson et al., "Towards user-centric metrics for denial-of-service measurement," in *Proceedings of the Workshop on Experimental Computer Science*, San Diego, CA, USA, June 2007.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8858444, 15 pages
https://doi.org/10.1155/2020/8858444

Research Article

A Framework of Abnormal Behavior Detection and Classification Based on Big Trajectory Data for Mobile Networks

Haiyan Zhang,[1,2] Yonglong Luo,[1,2] Qingying Yu,[1,2] Liping Sun,[1,2] Xuejing Li,[1,2] and Zhenqiang Sun[1,2]

[1]School of Computer and Information, Anhui Normal University, 241002 Wuhu, Anhui, China
[2]Anhui Provincial Key Laboratory of Network and Information Security, 241002 Wuhu, Anhui, China

Correspondence should be addressed to Yonglong Luo; ylluo@ustc.edu.cn

Received 28 September 2020; Revised 8 November 2020; Accepted 24 November 2020; Published 22 December 2020

Academic Editor: Zhe-Li Liu

Big trajectory data feature analysis for mobile networks is a popular big data analysis task. Due to the large coverage and complexity of the mobile networks, it is difficult to define and detect anomalies in urban motion behavior. Some existing methods are not suitable for the detection of abnormal urban vehicle trajectories because they use the limited single detection techniques, such as determining the common patterns. In this study, we propose a framework for urban trajectory modeling and anomaly detection. Our framework takes into account the fact that anomalous behavior manifests the overall shape of unusual locations and trajectories in the spatial domain as well as the way these locations appear. Therefore, this study determines the peripheral features required for anomaly detection, including spatial location, sequence, and behavioral features. Then, we explore sports behaviors from the three types of features and build a taxi trajectory model for anomaly detection. Anomaly detection, including sports behaviors, are (i) detour behavior detection using an algorithm for global router anomaly detection of trajectories having a pair of same starting and ending points; this method is based on the isolation forest algorithm; (ii) local speed anomaly detection based on the DBSCAN algorithm; and (iii) local shape anomaly detection based on the local outlier factor algorithm. Using a real-life dataset, we demonstrate the effectiveness of our methods in detecting outliers. Furthermore, experiments show that the proposed algorithms perform better than the classical algorithm in terms of high accuracy and recall rate; thus, the proposed methods can accurately detect drivers' abnormal behavior.

1. Introduction

Big data analysis is the detection of massive data and a type of thinking process, technology, and resource. The trajectory data for the mobile networks, which is a branch of big data, comprise a rich sequence of geospatial locations with timestamps and carry the information of the moving object's actual movement. They have the characteristics of time and space, spatially static but temporally dynamic [1]. A massive amount of vehicle trajectory data is collected by GPS-embedded vehicles. The "big trajectory data" under the mobile networks have contributed to the emergence of many data-driven trajectory-based applications such as route recommendation [2], transit time estimation [3, 4], traffic dynamic analysis [5], fraud detection [6], and city planning [7]. Analysis on such data to serve fields including intelligent transportation and smart cities has attracted the interest of a large number of researchers [8].

An abnormality generally implies that a data object is extremely deviant from the remaining set of retrieved data due to some of its unusual features. An abnormal trajectory differs clearly from most trajectories scrutinized under a similarity evaluation mechanism. An obvious rare pattern may indicate an abnormal event [9]. The detection results can help identify suspicious activities of vehicles and be used in many applications such as security surveillance, scheduling, and city planning [7, 10]. In the surveillance application, vehicle trajectories can be used in automatic visual surveillance [11], traffic management [12], suspicious activity detection [13], sports video analysis [14], video summarization [14], synopsis generation [15], and video-to-text descriptors [16], among others. Ngan et al. [17] adopted a Dirichlet process mixture model (DPMM) for detecting outliers in large-scale urban traffic data. Kingan and Westhuis [18] presented a regression model approach for average daily traffic. Therefore, outlier detection is an important analysis task.

Every day, thousands of people are victims of traffic accidents, which are generally directly related to driver behavior. According to the World Health Organization, the total number of road traffic deaths worldwide is approximately 1.3 million per year. The primary causes of accidents include speeding, drunk driving, unsafe lane switching, and incorrect turns, among others [19, 20]. Therefore, we divide anomalous trajectories into three categories: (1) global router anomaly, (2) local speed anomaly, and (3) local shape anomaly. Specifically, the global router anomaly means that the driver adds extra travel for some reasons. This anomaly occurs in various instances; for example, the taxi driver fraudulently increases the itinerary of the customer to obtain additional benefits, or the driver chooses another route to avoid traffic congestions or road repairs. Local speed anomaly refers to the vehicle exceeding the speed limitation specified for some road sections, such as roads near schools and hospitals. Local shape anomaly refers to the local shape of the trajectory meandering. This anomaly can occur due to several reasons such as drunk driving, in case of which the alcohol content in the blood of the vehicle driver exceeds a specified limit, leading to the slowing down of the reflex arc nerve and decrease in the tactile ability of the driver; these problems in turn cause instability of the steering wheel, eventually forcing the driver to continuously change lanes during driving.

To address the aforementioned problems, this study proposes three abnormal trajectory detection algorithms. The main contributions of this study are summarized as follows:

(1) This study systematically analyzes the abnormal behaviors of taxis, including detour behavior, speed anomaly, and local shape anomaly. According to the different anomalies causing abnormal taxi trajectory, different solutions are proposed.

(2) Most people who are new to the city will choose to take a taxi. Considering the situation of taxi detours and planned delay and safety issues, this study proposes a novel global router anomaly detection algorithm. When the starting and ending points are the same, the trajectory that greatly differs from the conventional route is considered a detour anomaly.

(3) In road sections where provision to take pictures to detect speed limit violation is absent, in order to detect taxis travelling at abnormal speed, this study proposes a method for detecting speed abnormal trajectories on the basis of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm.

(4) Considering the situation of unstable driving direction of the taxi caused by drunk driving or incorrect turns, this study proposes a method for detecting local shape abnormity on the basis of the local outlier factor (LOF) algorithm.

(5) Considerable research on urban monitoring has been conducted by analyzing videos or images. The detection framework proposed in this study detects anomalies in urban traffic by analyzing big trajectory data for the mobile networks. Thus, our framework can reduce the cost of urban monitoring and provide services for smart cities.

The rest of the paper is organized as follows. Section 2 introduces existing related work on abnormal trajectory detection. Section 3 introduces the assumptions adopted in this paper. Taxi abnormal trajectory detection algorithms are detailed in Section 4. Experimental setup and results are presented in Section 5, and the conclusions are presented in Section 6.

2. Related Work

Several abnormal trajectory detection approaches have been developed so far and continue to be explored; the existing approaches can be mainly classified into four categories: classification-based methods [21–24], distance-based methods [25–30], density-based methods [26, 27, 31–34], and statistical methods [35, 36].

2.1. Classification-Based Methods. A classification-based approach is a supervised learning algorithm. Commonly used classification algorithms include logistic regression [23], k-nearest neighbor algorithm [21], decision tree [22], and support vector machine [24]. The basic concept of the classification-based method is to train the algorithm through existing training samples to obtain an optimal model and then use this model to map all the inputs to the corresponding outputs, thereby making simple judgments on the output to achieve the purpose of the classification. It is an efficient method to classify unknown data. Therefore, when supervised learning algorithms are used, the dataset is divided into three categories during the experiments: training, test, and verification sets. The training set learns a classifier through labeled data; the test set is used to evaluate the performance of the algorithm; and the verification set can obtain appropriate hyperparameters. Both training and verification sets of this method must be labeled, and the classification effect of the classifier depends on the training dataset. Trajectory anomaly detection is difficult owing to the lack of ground truth data. Many researchers use human experts to label training data. However, in some cases, labeling data is impossible or difficult, which makes classification algorithms unreliable. In addition, given that the anomalies in the evolutionary trajectory are usually unknown and time dependent, it is impossible to obtain training data covering all anomalous instances in practical applications. Therefore, the classification-based anomaly detection method is not suitable for online anomaly detection of trajectory flow.

2.2. Distance-Based Methods. In the distance-based method, the trajectory in the trajectory dataset with a long distance, such as the Euclidean distance, Manhattan distance, and dynamic time warping (DTW) distance, from most trajectories is regarded as abnormal [25, 26, 30]. Knorr et al. introduced the concept of distance-based trajectory anomaly detection and, by conducting validation experiments using a database, proved that the method based on this concept can process high-dimensional data [25]. Hence, scholars have developed novel schemes on the basis of distance-based trajectory anomaly detection. This method uses a partition detection framework to detect abnormal trajectories and Hausdorff distance to measure the distance between two subtrajectories [26, 29]. Recently, San Román et al. proposed an abnormal trajectory detection method based on context-aware distance [28], wherein human trajectories are detected by video surveillance systems. First, the appropriate representation of each trajectory is selected by the polar coordinates of the trajectory. Then, the context-aware distance between trajectories is determined by the angle difference, the Euclidean distance, and the weighted average of the number of points in each trajectory. Subsequently, the trajectory distance matrix formed uses an unsupervised learning method to extract cohorts (clusters) of trajectories. Finally, an outlier detection method is used to detect anomalous trajectories in each cluster. Although distance-based detection methods are suitable for high-dimensional data, they are computationally expensive and time consuming. Moreover, they only adjust the abnormal behavior of the trajectory itself based on location information, ignoring the trajectory that is obviously different from its temporal and spatial neighbors in terms of nonlocation information.

2.3. Density-Based Methods. In density-based methods, outliers are objects in low-density areas [31, 32, 34]. Breunig et al. [31] defined a local outlier factor (LOF), which depends on the degree of isolation of an object relative to the surrounding neighborhood and has many desirable properties. For example, due to the local approach, LOF can identify outliers in a dataset that would not be outliers in another area of the dataset. The LOF and DBSCAN are similar, so some scholars used DBSCAN to detect outliers. A spatiotemporal (ST)-outlier detection method based on DBSCAN was proposed by adding the time dimension to a scheme presented by Kut and Birant [33]. First, a modified DBSCAN clustering algorithm is run on the tested data with two main modifications: (1) to support the time aspect, the tree is traversed to determine the space and time neighborhood of any object in a given radius; (2) to identify outliers, the algorithm allocates density factors to each cluster and compares the average value of clustering with the new clustering, when the clustering has different densities. After clustering, the potential outliers are detected. Furthermore, by checking the spatial neighbors, the objects are verified to be spatial exception values. Subsequently, the temporal neighbors of the spatial outliers identified in the previous step are checked. If the eigenvalue of the spatial anomaly is not significantly different from its temporal neighbor, it is not an ST exception. Otherwise, it is confirmed as an ST-outlier. The proposed scheme adds a limitation of the sliding window in the time dimension [37] and then divides trajectory anomaly detection into two categories: detection of abnormal trajectory points (PN-outliers) and detection of the entire trajectory (TN-outliers). This approach improves the detection efficiency, but the accuracy of trajectory detection is reduced. The time and space complexity of density clustering-based methods are linear or close to linear, so the detection of outliers is highly effective. The difficulty lies in the choice of the number of clusters and the existence of abnormal points. Extremely different results or effects are produced by different cluster numbers. Furthermore, a coarse quality greatly affects the quality of the outliers generated, and each clustering model is only suitable for specific data types.

2.4. Statistical Methods. In a statistical method, trajectory points are first modeled by assuming that a certain distribution is obeyed. Then, an abnormality is determined by checking whether the trajectory complies with the distribution model of trajectory points. The most frequent assumptions are Gaussian distributions [35] and multivariate Gaussian distributions [36]. When sufficient data and prior knowledge exist, using statistical methods for outlier detection can be very effective and efficient. However, such methods rely on a pathognomonic distribution model obtained with the used dataset, and it is difficult to select the parameters of the model. At present, few scholars use this method for abnormal trajectory detection. At the same time, most statistics-based outlier detection techniques use a single attribute. The current important question is how to model multivariate data (with multiple attributes).

The aforementioned abnormal trajectory detection methods can only detect abnormal trajectories in the driving range and driving direction and do not combine the characteristics of the abnormal driving behavior. In this study, we determine the peripheral features required for anomaly detection, including spatial location, sequence, and behavioral features. Then, we explore sports behaviors from the three types of features and build a taxi trajectory model for anomaly detection. The model systematically analyzes abnormal behaviors of drivers, but detection of such abnormalities is difficult.

3. Problem Description and Related Definitions

3.1. Problem Description. In this study, a given road network is denoted as $G(V, E)$ and a given trajectory dataset is denoted as RTS; we design algorithms to determine abnormal taxi trajectories and determine to which category of abnormal behavior of taxi drivers the detected trajectory belongs.

3.2. Related Definition. In this section, we provide the formal definitions of the parameters required for the algorithm.

Definition 1 (road network). The road network, denoted as $G(V, E)$, is a directed graph, where V represents the node set (i.e., the starting point and the ending point of the road section) and E is the edge set (i.e., the road section). For road $e \in E$, $e.s \in V$ is the starting point of the road section and $e.d \in V$ is the ending point of the road section.

Definition 2 (free trajectory point). Let t be a timestamp and (x, y) be a location in $\mathfrak{R}^2$. A free trajectory point is defined as a triple (x, y, t), implying that an object is at location (x, y) at time t.

Definition 3 (restrained trajectory point). Let e be a road section, and appending it to the free trajectory point generates a restrained trajectory point. A restrained trajectory point is defined as a quadruple (x, y, t, e). The trajectory points mentioned here onward are restrained trajectory points.

For example, the coordinates of restrained trajectory points A and B are, respectively, denoted as (x_a, y_a, t_a, e_1) and (x_b, y_b, t_b, e_1), where e_1 represents the *Wangfujing* section; this implies that locations (x_a, y_a) and (x_b, y_b) are on road section e_1 at times t_a and t_b, respectively.

Definition 4 (restrained trajectory). A restrained trajectory, denoted as RT, represents a set of multiple restricted trajectory points:

$$\mathrm{RT} = \{\mathrm{Tid}, (x_1, y_1, t_1, e_1), (x_2, y_2, t_2, e_2), \ldots, (x_n, y_n, t_n, e_r)\}, \quad (1)$$

where Tid is the identification of a trajectory, time stamp t is arranged in the ascending order, implying that $t_s < t_{s+1}$, t $(1 \leq s < n)$, n is the number of sampling, also called the length of the trajectory, and r is less than or equal to n. A road section can contain multiple locations, i.e.,(x_1, y_1, t_1, e_1), (x_2, y_2, t_2, e_1), and (x_3, y_3, t_3, e_1), but a location belongs to only one road section; in addition, the road sections of adjacent locations are either identical or adjacent.

Furthermore, if e_1 and e_r values of two trajectories are the same, they are called neighbors. Consider a set of m trajectories $\mathrm{RTS} = \{\mathrm{RT}_1, \mathrm{RT}_2, \ldots, \mathrm{RT}_m\}$, where $\mathrm{RT}_i = \{\mathrm{i}\ (x_1^i, y_1^i, t_1^i, e_1^i), \ldots, (x_q^i, y_q^i, t_q^i, e_r^i)\}$ represents the ith trajectory in RTS, $1 \leq i \leq n$.

Definition 5 (direction deflection angle). The direction deflection angle is defined as the degree of change at the position of a restrained trajectory point. Consider three consecutive trajectory points, denoted by $k(x_{i-1}, y_{i-1})$, $q(x_i, y_i)$, and $p(x_{i+1}, y_{i+1})$ (only the position of the direction deflection angle is considered in the calculation). Then, the direction deflection angle at trajectory point q is denoted as θ_q and is given by

$$\theta_q = \frac{\mathrm{distance}(p, q)^2 + \mathrm{distance}(q, k)^2 - \mathrm{distance}(p, k)^2}{2 * \mathrm{distance}(p, q)^* \mathrm{distance}(q.k)}. \quad (2)$$

Figures 1(a) and 1(b), respectively, show direction deflection angles less than 0 and greater than 0. Let A be the direction deflection angle of point q_1 on trajectory RT_1 and B be the direction deflection angle of point q_2 on trajectory RT_2. From equation (2), it can be seen that A is less than 0 and B is greater than or equal to 0.

Definition 6 (deme). Two trajectories RT_i and RT_j with their starting and ending points on the same road section are considered to be in a deme. Each deme includes two attributes, namely, the starting and ending road sections. According to the road section attributes of the starting and ending points, trajectories can be divided into various demes. The rth deme is denoted as D_r, with attributes $D_r.\mathrm{se} \in E$ and $D_r.\mathrm{de} \in E$, respectively.

Definition 7 (ATD-outlier). ATD-outlier includes three types of anomalies, namely, global router anomaly, local speed anomaly, and local shape anomaly. If a trajectory is an ATD-outlier, it can be at least one of the above anomalies.

4. ATD-Outlier Detection Algorithms

4.1. Framework Overview. In this section, we present an overview of our proposed framework. It contains two stages: the preprocessing stage and the anomaly detection stage, which contains three algorithms. As shown in Figure 2, in the trajectory data preprocessing stage, a map matching algorithm based on AntMapper [38] is used to match the trajectory points to the road sections. This method considers both local geometric/topological information and global similarity measures and uses an ant colony optimization algorithm, which mimics the pathfinding process of ants transporting food in nature. In addition, local heuristics and global fitness are used to search for the global optimal value of the model with high matching accuracy. The framework of the anomaly detection phase is described as follows:

Global router anomaly detection algorithm: the similarity between trajectories in the same deme is used as the input to the isolation forest (iForest) algorithm that trains a suitable model to determine global router anomaly trajectories.

Local speed anomaly detection algorithm: the instantaneous velocities of trajectory points are clustered by DBSCAN for each road section. A trajectory having a sufficient number of speed anomaly points will be marked as a local speed anomaly trajectory.

Local shape anomaly detection algorithm: the direction deflection angle of each trajectory point is calculated. The deflection angle of the trajectory point on the same road section is used as the input of the LOF algorithm to determine the abnormal trajectory of the lane change.

4.2. Global Router Anomaly Detection Algorithm. Currently, taxi charges for public are calculated on the basis of a standard mileage. In order to make extra profits, some taxi drivers take their passengers via long routes to their destinations in the urban road network, thereby fraudulently increasing mileage. However, traffic authorities cannot

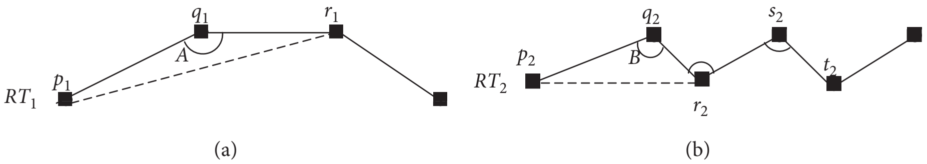


Figure 1: Examples of the direction deflection angle.

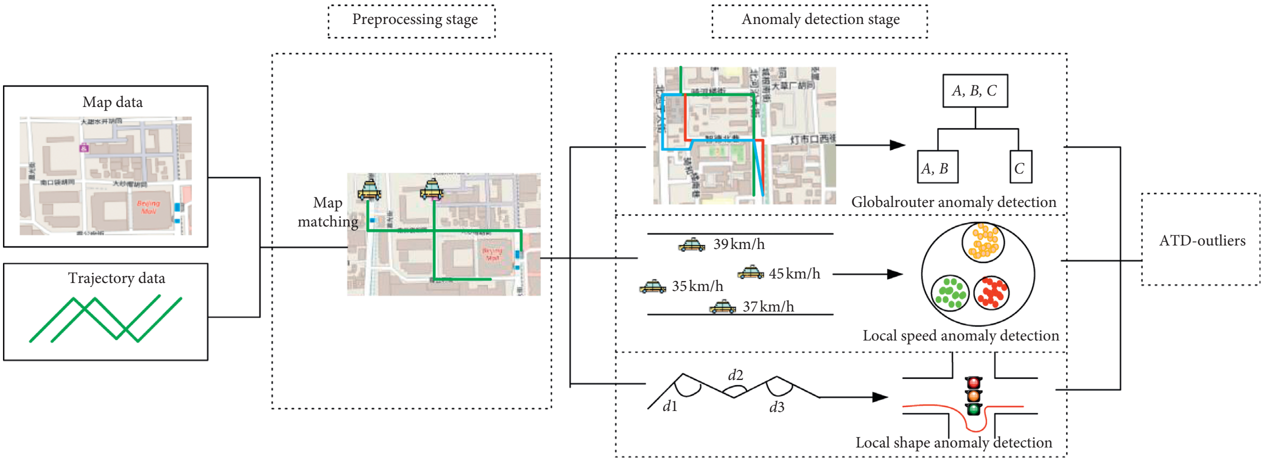


Figure 2: System framework diagram.

manually investigate and deal with such illegal behavior of taxi drivers. Therefore, a global router anomaly detection algorithm is proposed.

For example, as shown in Figure 3, a number of trajectories exist in a deme. The red line indicates an odd path that is different from the others (indicated by the yellow lines), for example, in terms of the length. This odd route is considered the global router anomaly trajectory as the driver has taken a very long route.

The distance between two trajectories determined by dynamic time warping (*DTW*) gives the similarity between the trajectories. The dynamic time regularity algorithm measures the similarity between two different time series. Using the distance function to determine the similarity between two trajectories that do not have a similar trip time is not feasible. However, if two trajectories have the same starting and ending points and a very similar time taken for the trips, they are comparable in terms of the distance function. Hereafter, in this paper, the reference to similarity between objects implies that the objects are in the same deme.

For example, let us consider comparison of a template trajectory sequence Q with an actual sampled trajectory sequence C; because of the different route patterns, both trajectories cannot be aligned. However, the first sampling value and the last sampling value of the two trajectories are taken such that they correspond pairwise to each other. Then, the process of calculating similarity is as follows.

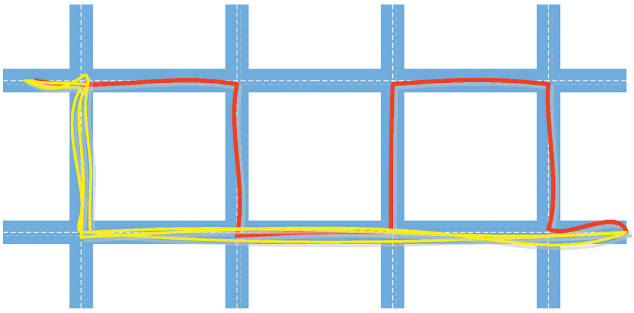


Figure 3: Example of the global router anomaly trajectory.

Step 1: we construct a $n \times m$ matrix, with elements $d(i, j) = \text{distance}(q_i, c_j)$. Without loss of generality, we utilize the Euclidean distance as the distance measure.

Step 2: the shortest path from $d(1, 1)$ to $d(m, n)$ is searched with dynamic programming. Because Q and C are both time series, there are only three directions to search.

Step 3: the similarity between trajectories Q and C is calculated by the shortest path from $d(1, 1)$ to $d(m, n)$.

Definition 8 (trajectory similarity). The similarity between two trajectories Q and C, denoted as SIM, is calculated as follows:

$$\mathrm{SIM}_{Q,C} = \gamma(i,j) = d\left(q_i, c_j\right) + \min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}. \tag{3}$$

In the equation, q_i and c_j, respectively, represent the ith point of Q and the jth point of C, and the similarity between Q and C is given by the value of $\gamma(n,m)$, where n and m, respectively, represent the lengths of Q and C, such that $1 \le i \le n$, and $1 \le j \le m$.

Definition 9 (deme similarity matrix). The deme similarity matrix, denoted as SM, is established for each deme. If $\mathrm{RT}_i, \mathrm{RT}_{i+1}, \ldots, \mathrm{RT}_k \in D_r$, $\forall \mathrm{RT}_i.e.s = \mathrm{RT}_j.e.s = D_r.\mathrm{se}$, $\mathrm{RT}_i.e.d = \mathrm{RT}_j.e.d = D_r.\mathrm{de}\,(i < j < k)$, the similarity matrix of the rth deme, denoted as SM_r, is calculated as follows:

$$\mathrm{SM}_r = \begin{bmatrix} \mathrm{SIM}_{1,1} & \ldots & \mathrm{SIM}_{1n} \\ \vdots & \ddots & \vdots \\ \mathrm{SIM}_{n1} & \ldots & \mathrm{SIM}_{nn} \end{bmatrix}, \tag{4}$$

where $\mathrm{SIM}_{i,j}$ is calculated by equation (3) and n is the number of trajectories in D_r.

When the number of trajectories is especially large in a deme, the dimension of this matrix is difficult to predict. Therefore, we set a constraint as follows:

$$\mathrm{maxtrix\_dimension}_r = \begin{cases} 10, & \text{if } \mathrm{len}\left(D_r\right) > 10, \\ \mathrm{len}\left(D_r\right), & 0 < \text{otherwise} \le 10, \end{cases} \tag{5}$$

where maxtrix_dimension$_r$ represents the dimension of the similarity matrix of D_r. When the number of trajectories in D_r is greater than 10, the dimension is set to 10. Otherwise, it is set equal to the number of trajectories. Another reason for setting the limitation is that it is not reasonable to use the attributes of trajectories with high similarity for anomaly detection. Consequently, we sort the similarity of each trajectory in the ascending order and select the attributes of the lowest ten similarities and use them as the input to the iForest algorithm. Theoretically, the length of the trajectory marked as abnormal should be longer than the length of the normal trajectories.

The iForest algorithm is an unsupervised anomaly detection method suitable for continuous data. It was first proposed by Professor Zhihua Zhou of Nanjing University in 2008 [39], and an improved version was proposed in 2012 [40]. Different from the other anomaly detection algorithms, which portray the degree of dissimilarity between samples is through distance, density, and other indicators, the iForest algorithm detects outliers by isolating sample points. Specifically, the algorithm isolates a sample using a binary search tree structure called the isolation tree or, in short, iTree. Because the number of outliers is small and alienated from most samples, the outliers are isolated earlier, that is, the outliers are close to the root node of iTree, whereas the normal samples are placed far from the root node. In addition, compared to traditional algorithms such as LOF and K-means, the iForest algorithm is more robust to high-dimensional data. Therefore, it is suitable for detour trajectories, which have ten dimensions. The specific Algorithm is as follows.

The time complexity of Algorithm 1 depends on the following aspects: (a) the time for calculating the *SM* matrix, whose time complexity is $(d \times s \times m)$, where d is the size of a deme and m and s are the length of trajectories, respectively; (b) the time of the iForest algorithm, whose time complexity is $o(n)$. To be precise, n is the largest size of a deme; therefore, the total time complexity is $o(\mathrm{len} \times s \times m \times n)$, where *len* is the size of a deme. The spatial complexity is $o(n^2)$, which is mainly due to storing of the *SM* matrix. The parameters of the iForest algorithm are the same as those used in the literature [37], so they are not listed.

4.3. *Local Speed Anomaly Detection Algorithm.*

The local speed refers to the instantaneous speed of each trajectory point. Owing to the road section attribute, trajectory points are also classified. Then, the instantaneous velocity of trajectory points of each road section is clustered using the DBSCAN algorithm.

DBSCAN is a classical density-based clustering algorithm, having the following main characteristics: (1) the number of clusters does not need to be specified in advance when clustering and (2) the number of clusters is uncertain.

The correlative concept definitions of this section are presented as follows.

Definition 10 (instantaneous velocity of the trajectory point). Consider two consecutive trajectory points, denoted as $p(x_i, y_i, t_i)$ and $q(x_{i+1}, y_{i+1}, t_{i+1})$; the instantaneous velocity of point q is obtained as follows:

$$\Delta v_q = \frac{\mathrm{distance}\,(p,q)}{t_{i+1} - t_i}. \tag{6}$$

Definition 11 (core point). The core point indicates a point within a radius Eps that contains more than ε points (where ε is the minimum number of points to form a dense region). A point that contains less than ε points within a radius Eps falls in the neighborhood of the core point, which is also called the boundary point. Moreover, the noise point is neither a core nor a boundary point.

As shown in Figure 4 if ε is set to 3, according to Definition 11, the red points are core points because there are three points in the red circles with a radius Eps. The radius of all circles is Eps. There are two points in the blue circle centered on point B, so B is not the core point, but it falls within the red circle, so point B is the boundary point. Because the number of points in the green circle with C as the center is less than 3, C is not the core point, and because C does not fall within the red circle, it is not the boundary point, so C is the noise point.

Definition 12 (Eps-neighborhood). The neighborhood within a given object radius Eps is called Eps-neighborhood of the object. We denote the set of points within a radius Eps of point p as N_Eps(p):

```
Input: RTS (a dataset of restrained trajectories)
Output: $OUT$ (a dataset of the number of abnormal trajectories)
$MD \longleftarrow$ Extract the road section identifier value of the starting and
    ending points of all trajectories in RTS;
$MD \longleftarrow$ Delete duplicate pairs of road section;
$deme \longleftarrow$ Classify RT by $MD$;
$len \longleftarrow |deme|$;
for $i \longleftarrow$ 1 to len do:
    Initialize the matrix $SM$;
    Calculate $SM$ of the trajectories included in $deme$[i] using (4);
    $OUT$.append(iForest($SM$));
endfor
return $OUT$;
```

Algorithm 1: *GRAD*: global router anomaly detection algorithm.

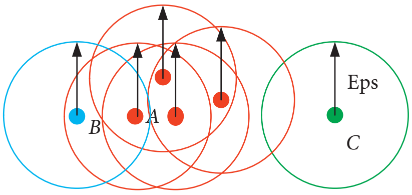


Figure 4: Example of core point, boundary point, and noise point.

$$N_{\mathrm{Eps}}(p) = \{q|qt\epsilon n\, qD, distance(p,q) < \mathrm{Eps}\}. \qquad (7)$$

D is a given object set.

Definition 13 (density connection). Given an object set D if p is in the Eps-neighborhood of q and q is a core object, then object p is defined as directly density reachable. If there is an object chain $p_1, p_2, \ldots, p_n$, $p_1 = q$, $p_n = p$, $p_i \in D (1 \le i \le n)$, where p_{i+1} is the direct density from p_i about Eps and ε, then object p is defined as density reachable from object q with Eps and ε. If there is an object $O \in D$ and objects p and q are density reachable from about Eps and ε, then objects p and q are defined as density connection about Eps and ε.

Principle of judging abnormal points:

Step 1: clusters are created by checking Eps-neighborhood of each point in the dataset; if the Eps-neighborhood of point P contains more than points, a cluster is created with P as the core object.

Step 2: then, the aggregation of iterations from these core objects gives directly density reachable objects; this process involves merging of some density reachable clusters.

Step 3: when no new points are added to any cluster, the clustering process ends. Points that are not classified into any class are suspected anomaly points.

Step 4: each trajectory is checked for whether consecutive points are marked in it. If so, such trajectories are outliers.

Speed of urban taxis is subject to traffic laws in various road sections, but it is difficult to accurately obtain the speed limit of each road section. However, clustering the instantaneous speed of moving objects on various road sections is a feasible solution. If the instantaneous speed of a moving object differs greatly from those of other moving objects, that object may be regarded as an abnormal one. The detection result can be applied to detection of over speeding in real life. The specific algorithm is as follows.

Line 6 of Algorithm 2 clusters the instantaneous speed of trajectory points for each road section and records the location of noise points in *CLUSTER*. The time complexity of Algorithm 2 depends on the following aspects: (a) the time for clustering, whose time complexity is $o(n \times \log n)$ in low dimensions, where n is the number of trajectory points; (b) the time of checking trajectories, whose time complexity is $o(n)$. Therefore, the total time complexity of Algorithm 2 is $o(n^2 \times \log n)$. The spatial complexity is $o(n^2)$, which is mainly due to storing of *SPEED* and *CLUSTER* matrixes.

4.4. Local Shape Anomaly Detection Algorithm. The implication of a local shape anomaly on the trajectory is an abrupt change in the direction of the trajectory. Such deviations are considered illegal if they occur at an intersection or successively.

The LOF algorithm is an unsupervised outlier detection method proposed by Berning et al. [31] in 2000. It is a representative algorithm among outlier detection methods based on density. The algorithm calculates an outlier factor LOF for each point in the dataset and determines whether it is an outlier factor by judging whether the LOF is close to 1. If the LOF is much greater than 1, it is considered an outlier factor, whereas if it is close to 1, it is a normal point. Herein, we only provide a brief introduction to the concept of the algorithm, see [31], for further details.

This study mainly uses the LOF algorithm to detect the anomaly of the deflection angle of the track point on the same road section. The direction deflection angle of a trajectory point is evaluated by equation (2). The specific algorithm is shown as follows.

The time complexity of Algorithm 3 depends on the following aspects: (a) the time of calculating the direction deflection angle of trajectory points; (b) the time required to

```
Input: $E$ (a dataset of road sections), RTS (a dataset of restrained trajectories)
Output: $OUT$ (a dataset of number of trajectories with speed anomaly)
Initialize the two matrixes: $SPEED$ and $CLUSTER$;
Initialize the array $OUT$;
$SPEED \longleftarrow$ calculate instantaneous velocity of each trajectory point;
$p \longleftarrow |E|$;
for $i \longleftarrow 1$ to $p$ do:
    Cluster instantaneous velocity of RTS on the $i$-th road section;
    Delete the trajectory that only have an exception in a period of time;
endfor
Store the restrained trajectory number with abnormal speed at $OUT$;
return $OUT$;
```

Algorithm 2: *LADA* local anomaly detection algorithm.

```
Input: RTS (a dataset of restrained trajectories)
Output: $OUT$ (a dataset of number of abnormal trajectories)
Initialize the array $OUT$;
$p \longleftarrow |E|$;
for $i \longleftarrow 1$ to $p$ do:
    Initialize the $ANGLE$ matrix;
    Initialize the array LOF;
    $ANGLE \longleftarrow$ calculate the direction deflection angle of each trajectory point on the road section i
    LOF $\longleftarrow$ local outlier factor()/* The direction deflection angle of all track points on the road section $i$ is used as the input of the local outlier factor function, and it is judged whether a trajectory point is abnormal according to the set threshold; the abnormal point is $-1$*/;
    If there are more than two abnormal points near the determined abnormal point, the track where the abnormal point is located is considered abnormal;
endfor
Store the restrained trajectory number with local shape anomaly at $OUT$;
return $OUT$;
```

Algorithm 3: LSAD local shape anomaly detection algorithm.

execute the LOF algorithm. The LOF algorithm must calculate the distance between the two data points, resulting in the time complexity of the entire algorithm, $o(n^2)$, where n is the total number of all track points on road section i. The time complexity of calculating the direction deflection angle of trajectory points is $o(n)$; therefore, the total time complexity of local shape anomaly detection is $o(n^2 * p)$. The spatial complexity is $o(n^2)$, which is mainly contributed by the storing of the array *ANGLE*.

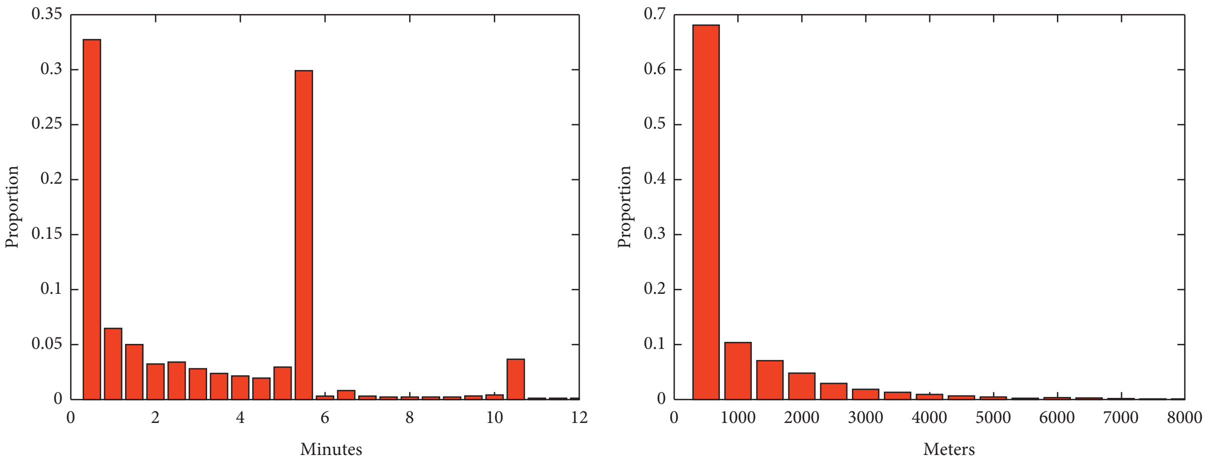
In this section, we introduced three different anomaly detection algorithms to detect three illegal behaviors of taxi drivers and established the algorithms by analyzing the characteristics of taxi trajectories, such as the characteristics of road segments and local characteristics of trajectories. Combining the three anomaly detection algorithms will save considerable labor cost and ensure safety and convenience of human travel.

5. Experiments

Experiments were conducted using Python3.7, with the software and hardware environment being Intel Core i5 @ 2.30 GHz quad-core CPU, 16G memory, and Windows 10 operating system.

The dataset is described in Section 5.1. We compare the three detection algorithms with a previous algorithm for trajectory neighbor (TN)-outlier [37].

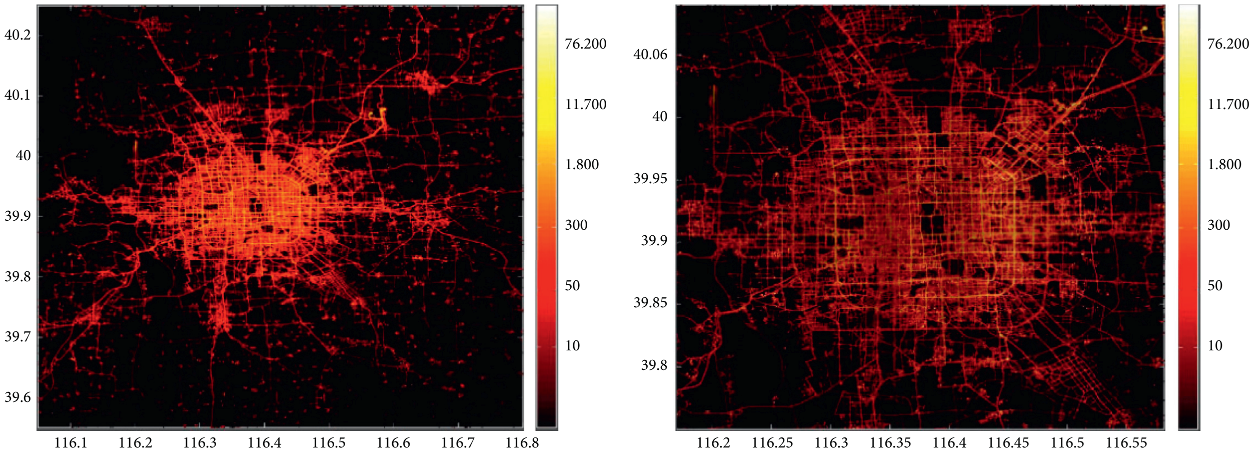
5.1. Dataset Selection. The dataset contains the GPS trajectory data of 10357 taxis in Beijing for a period from February 2 to February 8, 2008. A total of approximately 15 million points are present in the dataset. The total distance of the trajectory reaches 9 million kilometers. Figure 5 plots the distribution of time interval and distance interval between two consecutive points. The average sampling interval between two points is approximately 177 s, and the average distance between two points is approximately 623 m. The figure indicates that the sampling frequency has approximately 50% of the trajectory points within three minutes. Figure 6 shows the density distribution of the GPS points in the dataset. The dataset provides all data for each taxi. Therefore, we considered that the user trajectory changes when the



(a)

(b)

Figure 5: Histograms of time and distance intervals between two consecutive points: (a) time interval and (b) distance interval.



(a)

(b)

Figure 6: Distribution of GPS points, where the color gradient indicates the density of the points: (a) data overview in Beijing and (b) data within the 5th ring road of Beijing.

sampling interval is greater than 10 min. Moreover, the experimental data with the taxi trajectory length less than 10 are disqualified from the dataset.

To evaluate the precision of trajectory outlier detection, we selected the GPS trajectory of 5000 taxis, of which 7000 trajectories were screened out. We used the trajectory anomaly detection algorithm presented in the literature [25, 26, 31] to mark the 7000 trajectories. The trajectories marked as normal by these algorithms are regarded as true normal trajectories, whereas those marked as abnormal are regarded as true abnormal trajectories. In this manner, 3192 trajectories were marked as normal and 1186 trajectories were marked as abnormal. Due to the particularity of local shape anomaly detection and because the dataset used in this study has a low sampling frequency, we used cubic spline interpolation to interpolate the original trajectory.

5.2. Parameter Setting. In Algorithm 1, we selected the lowest ten similarities as trajectory attributes to detect anomaly. An enormous trajectory that did not have a neighbor was deemed abnormal. For a trajectory having a number of neighboring trajectories, we selected the minimum number of trajectory neighbors in the dataset as the number of attributes. However, the number of attributes must be greater than or equal to 2 and less than or equal to 10 because if the data dimension is extremely large, prediction by iForest may not be suitable.

The parameters Eps and ε vary for different applications and datasets. The first road data derived from OpenStreetMap show that a latitude and longitude of 0.00046 corresponds to an actual distance of 39.3 m. In Algorithm 2, we used a taxi speed of 40 mph as the standard speed, which is 64.2 km/h, given that 1 mile is 1605 m. We set ε between 1 and 10 m/s and converted it to latitude and longitude, that is, the Eps range was 1.17048×10^{-5} to 1.17048×10^{-4}. To facilitate the calculation, we expanded the data by 10,000 times such that Eps was normalized between 0 and 1. Then, Eps was set between 1 and 6.

5.3. Experimental Results. Some results of our methods were compared with the results of the TN-outlier detection algorithm, which is one of the most popular trajectory outlier detection algorithms. The trajectory data in this study was required to be preprocessed for map matching. We used the AntMapper algorithm [38] to match the trajectory points to the road section and then classified the starting and ending pairs. This algorithm uses an ant colony optimization algorithm that mimics the pathfinding process of ants transporting food in nature. It uses local heuristics and global fitness to search for the global optimal value of the model. For a 5-min sampling frequency, this algorithm could achieve a matching accuracy of 93.97%.

5.4. Visual Display of the Global Router Anomaly Detection Result. To illustrate that each trajectory can find corresponding neighbors under a large data volume, we randomly selected trajectories of three taxis and categorized them. When two sampling points of a taxi trajectory exceeded 10 min, the other trajectory was considered to have begun. The different colored lines in Figure 7 represent different trajectories, and each submap represents a taxi journey from February 2 to February 8, 2008. Clearly, most of the trajectories are concentrated in certain places where the passengers are transported back and forth; therefore, it is reasonable to classify the trajectories according to the departure and destination locations.

In Algorithm 1, for learning using the iForest algorithm, the parameters used in the literature [39] were adopted. Figures 8(a) and 8(b), respectively, illustrate the detour detection results of the global router anomaly detection algorithm and TN-outlier algorithm on the real taxi trajectory dataset.

Owing to the large number of demes in this dataset, we selected some demes to show the detection results in Figure 8. In the figure, the red lines indicate the trajectories that are detour or without neighbors and, hence, anomalies. The blue lines indicate normal trajectories. From the figure, we can observe that the abnormal trajectory is longer than the normal and that, in the middle of the trip, the abnormal trajectory increases the distance to the destination by taking some other road sections than normal.

The number of abnormal trajectories shown in Figure 8(b) is less than that in Figure 8(a). TN-outlier detection, as shown in Figure 8(b), can also detect trajectories without neighbors but not detour trajectories. This is because the TN-outlier detection algorithm does not account for the fact that a taxi increases the distance to the destination by detour, but only analyzes the shape characteristics of the trajectory and trajectory point neighbors. The global router anomaly corresponds to the long detour behavior of taxi drivers for gaining a higher profit than the profit without a detour. Of course, the long detour may be chosen by the taxi driver due to traffic jams or road repairs. In case of force majeure, most drivers may choose a longer trip, so the taxi driver's neighbors can be found in the route; therefore, this situation is not a global router anomaly. The global router anomaly detection can be used to track the itinerary of taxis or cars hired through online booking as a measure to protect the interests and safety of passengers.

5.5. Local Speed Anomaly Detection. The abnormal detection of speed cannot be achieved in the trajectory. Furthermore, a restrained trajectory was marked with a road section label, while the velocity of each trajectory was clustered to determine the abnormal speed. In order to illustrate the different detection results with different values of Eps and ε, we set Eps between 0 and 1 and ε between 1 and 6. Figure 9 shows the average number of trajectories with abnormal speed on each road section.

Algorithm 2 was used to detect trajectory outliers by DBSCAN to cluster the instantaneous velocity of the trajectory points. Because of the large number of clusters in this dataset, we selected clustering results of three road sections between 13 : 00 and 14 : 00 on February 2, 2008, as shown in Figure 10. The blue points represent the normal and the black points represent exceptions. After the completion of the clustering, we checked each trajectory for whether it contained consecutive points that were marked. Such trajectories, if any, were outliers. The point where the speed is abnormal is not distinguishable by observing the trajectory; hence, we do not show the speed anomaly detection results. Local speed anomaly can be used for over speed detection without video surveillance. Installing video surveillance in every corner of the city requires considerable manpower, financial resources, and regular maintenance of the equipment. However, the speed detection of vehicles is crucial because numerous accidents of individual or multiple vehicles occur due to over speeding every year. When the instantaneous speed of a vehicle at multiple consecutive moments is substantially different from the speed of other vehicles in the same lane, it is regarded as a local speed abnormality. In the final calculation of the precision and recall rates, we add the results of speed detection.

5.6. Local Shape Anomaly Detection. Algorithm 3 provides local shape anomaly detection. Based on equation (2), we calculated the directional deflection angle for the real dataset. Then, we used the direction deflection angle of the track point on each road segment as the input of the LOF algorithm to detect the trajectory points with an abnormal deflection angle. Furthermore, trajectories that contain two or more such points were marked as abnormal. In the LOF algorithm, we assigned different values of k, d, and f to compare the total precision, and the final parameters were set to $k = 5$, $d = 0.5$, and $f = 0.15$.

Figure 11(a) shows the detection result of trajectory outliers based on the LSAD algorithm; the outliers are indicated by red lines, while normal trajectories are indicated by blue lines. In the left-middle of Figure 11(a), several trajectories with a zigzag shape are marked as abnormal, but

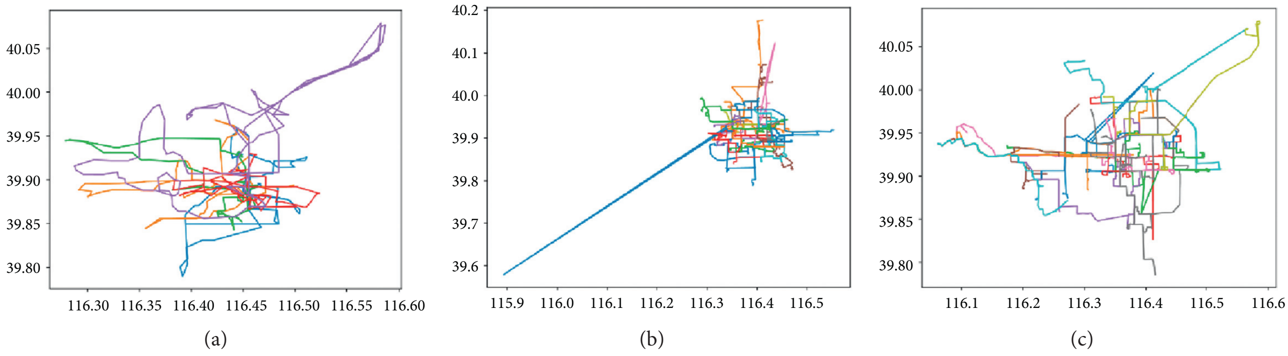


Figure 7: Trajectory slice: (a) taxi 1, (b) taxi 2, and (c) taxi 3.

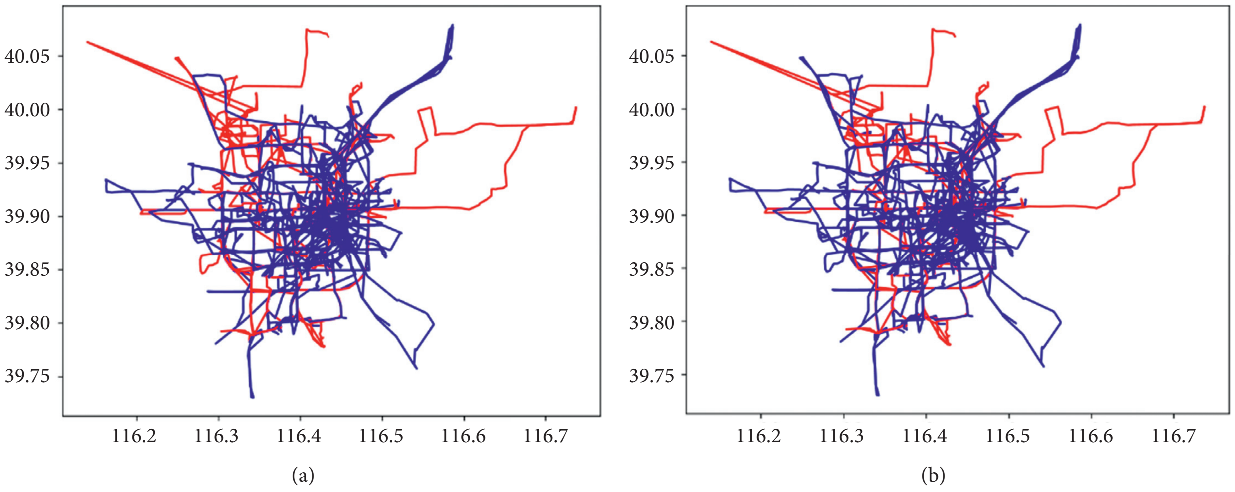


Figure 8: Detour detection by (a) GRAD (global router anomaly detection algorithm) and (b) TN-outlier detection algorithm.

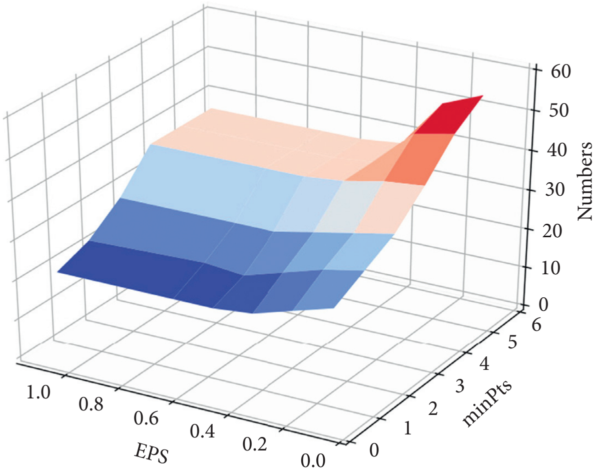


Figure 9: Average number of trajectories with abnormal speed.

in Figure 11(b) they are normal. These trajectories are determined by the LSAD algorithm as abnormal local shape. The LSAD algorithm combined with the road network analyzes whether the position of the point with a large degree of continuous curvature is at the intersection. The TN-outlier algorithm does not consider the feature. The road network data are too large to be clearly visible even after expanding the map, so the map is not displayed. Local shape anomaly may be caused by drunk driving or sudden sharp turns. Although the traffic inspection department considers

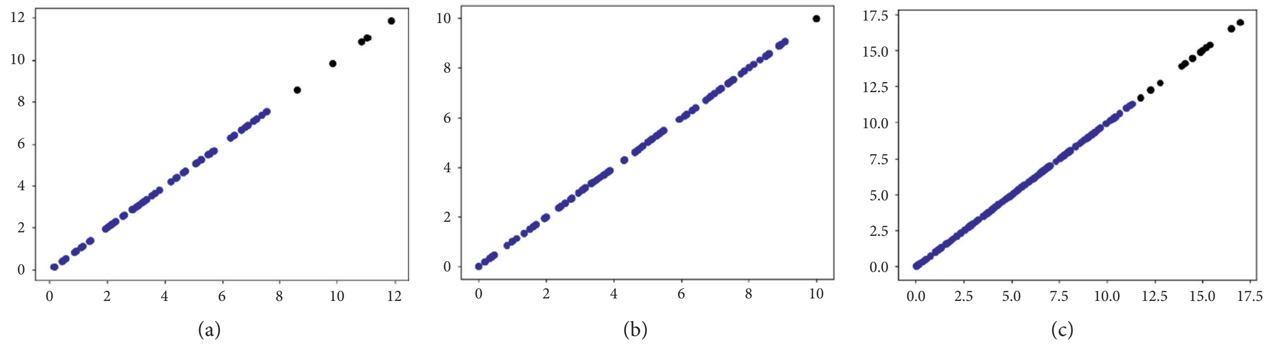


Figure 10: Speed cluster results: (a) road section 1, (b) road section 2, and (c) road section 3.

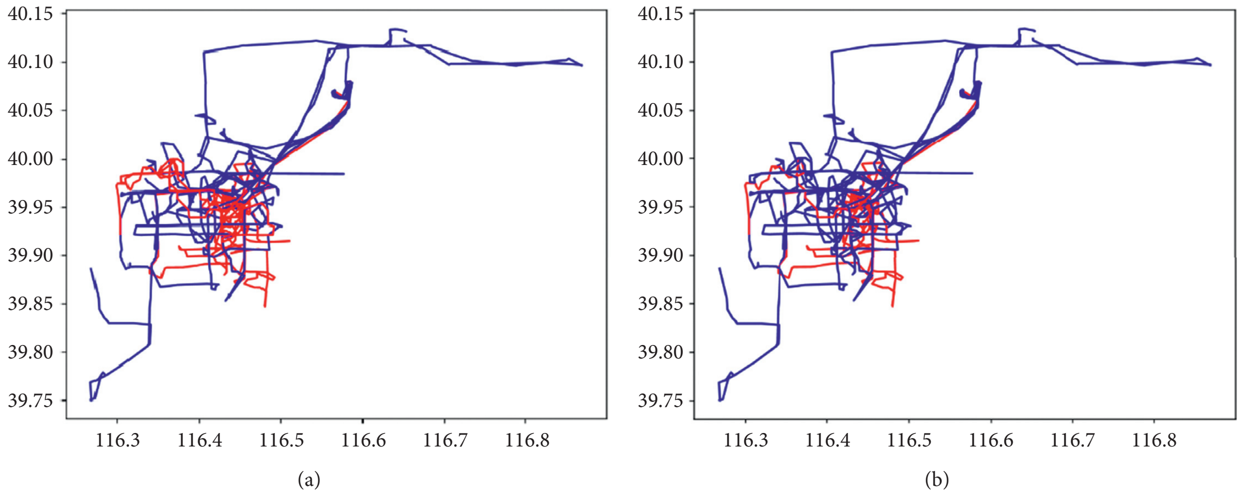


Figure 11: Deflection angle anomaly detection. (a) LSAD and (b) TN-outlier.

drunk driving a serious problem, it has always been tested manually, which requires considerable manpower and time. Local shape anomaly detection can be combined with trajectory semantics to determine whether the vehicle driver is drunk driving. If the starting point of the local abnormal trajectory is in a certain hotel, the driver is very likely to be drunk driving. The possibilities for local abnormalities are numerous, so we did not classify them specifically. If the local shape anomaly detection algorithm is applied to the traffic supervision system and combined with trajectory semantics, it can be further classified in detail.

5.7. Accuracy and Recall Rate of Abnormal Trajectory Detection. Next, we used precision and recall to measure the performance of abnormal trajectory detection. When calculating the precision and recall rate, the classification of abnormalities is not considered, but only whether the trajectory is abnormal is considered. Precision and recall are defined as follows:

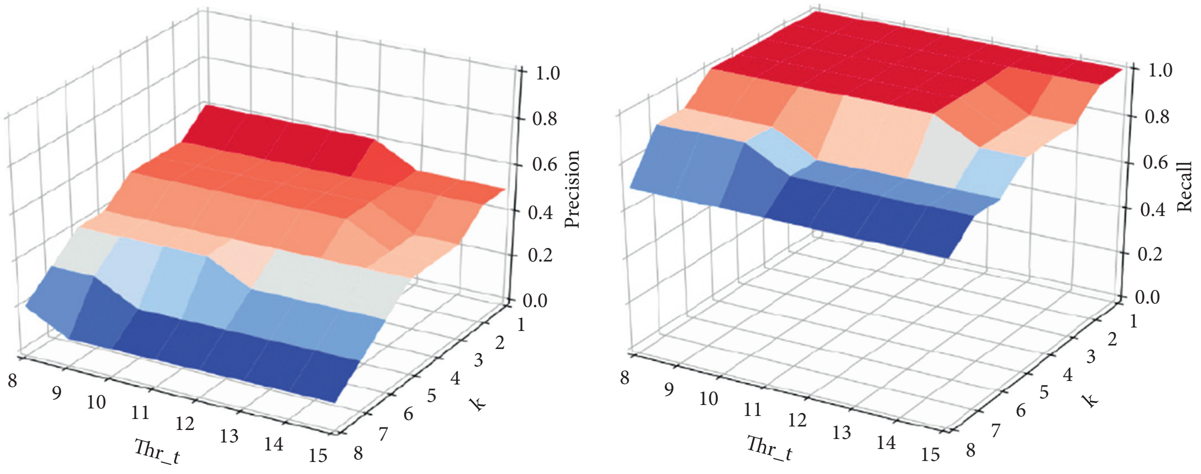
$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \end{aligned} \tag{8}$$

where TP represents the number of detected abnormal trajectories, TN represents the number of detected normal trajectories, FP indicates the number of normal trajectories that are falsely detected as abnormal, and FN represents the number of abnormal trajectories that are falsely detected as normal trajectories.

Figures 12(a) and 12(b) show the precision and recall rate of the TN-Outlier detection algorithm. Although the recall is nearly 100%, the abnormal detection precision of the TN-outlier detection algorithm for taxi trajectories is not ideal.

Figure 13 shows the precision and recall rate of ATD-outlier. The number of abnormal trajectories in the precision and recall rate calculation is obtained by the union of the results of the three algorithms. The x-axis and y-axis of Figure 13(a), respectively, represent Eps and ε, and the z-axis represents the precision. In Figure 13(b), the z-axis represents the recall rate.

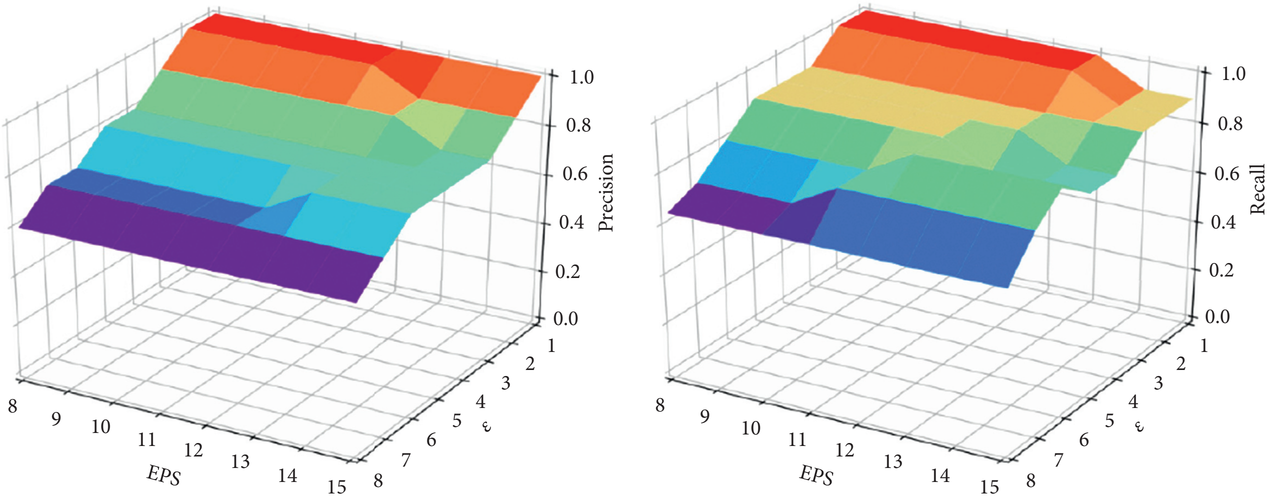
Although its recall rate is comparable to our ATD-outlier detection algorithm (Figure 13(b)), the TN-outlier algorithm considers a trajectory of a taxi as an outlier only if the taxi always moves alone. Moreover, the behavior of a taxi driver will be considered abnormal only if the driver always moves to regions that other taxi drivers hardly visit. Therefore, the TN-outlier detection algorithm frequently misclassifies trajectories of taxis.



(a)

(b)

Figure 12: (a) Precision and (b) recall of the TN-outlier detection algorithm.



(a)

(b)

Figure 13: (a) Precision (b) recall of the ATD-outlier detection algorithm.

6. Conclusions

This study mainly focused on different anomalous features of trajectories and road network environment and proposed three corresponding detection methods. (1) Global router anomaly detection algorithm: according to the road section attributes of starting and destination points, the trajectories were first classified; then, abnormal trajectories in each deme were detected. (2) Local speed anomaly detection algorithm: the instantaneous speed of each trajectory was calculated; then, clustering algorithm was used to determine trajectories with abnormal speed. (3) Local shape anomaly detection algorithm: the trajectories with an abnormal deflection direction were determined on the basis of the direction deflection angle of trajectory points. Our framework contributes to city monitoring by analyzing big trajectory data under the mobile networks. Experiments to verify the algorithms were conducted using the Beijing taxi trajectory dataset of 2008. The results indicate that the proposed algorithms are better than an existing method tested for comparison. In general, the proposed methods can be applied in the construction of smart cities. The algorithm in this study roughly divides the abnormal trajectories into three categories according to the abnormal behavior of users. However, in actual situations, the classification of abnormal trajectories is complicated, and there are more than the three categories. In future work, we will perform further detailed anomaly classification for each type of anomaly and integrate time attributes and semantics, analyze road traffic, and provide personalized route recommendations because research based on real-time traffic of road sections is more meaningful.

Data Availability

The data used to support the findings of the study are available at https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61972439, 61672039, and 61702010). and the Key Program in the Youth Elite Support Plan in Universities of Anhui Province (gxyqZD2020004).

References

[1] Y. Huang, B. Li, Z. Liu et al., "ThinORAM: towards practical oblivious data access in fog computing environment," *IEEE Transactions on Services Computing*, vol. 13, no. 4, p. 602, 2020.

[2] W. Luo, H. Tan, L. Chen, and L. M. Ni, "Finding time period-based most frequent path in big trajectory data," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, June 2013.

[3] Z. Liu, B. Li, Y. Huang, J. Li, Y. Xiang, and W. Pedrycz, "NewMCOS: towards a practical multi-cloud oblivious storage scheme," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 4, pp. 714–727, 2019.

[4] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, August 2014.

[5] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1679–1693, 2012.

[6] S. Liu, L. M. Ni, and R. Krishnan, "Fraud detection from taxis' driving behaviors," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 464–472, 2013.

[7] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, Beijing, China, September 2011.

[8] F. Meng, G. Yuan, S. Lv, Z. Wang, and S. Xia, "An overview on trajectory outlier detection," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2437–2456, 2019.

[9] J. Li, Y. Huang, Y. Wei et al., "Searchable symmetric encryption with forward search privacy," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2019.

[10] Z. Liu, J. Li, S. Lv et al., "EncodeORE: reducing leakage and preserving practicality in order-revealing encryption," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[11] Y. Djenouri, A. Belhadi, J. C.-W. Lin, D. Djenouri, and A. Cano, "A survey on urban traffic anomalies detection algorithms," *IEEE Access*, vol. 7, no. 7, pp. 12192–12205, 2019.

[12] J. Bian, D. Tian, Y. Tang, and D. Tao, "Trajectory data classification," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 4, pp. 1–34, 2019.

[13] T. Xiang and S. Gong, "Video behavior profiling for anomaly detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 893–908, 2008.

[14] A. Rehman and T. Saba, "Features extraction for soccer video semantic analysis: current achievements and remaining issues," *Artificial Intelligence Review*, vol. 41, no. 3, pp. 451–461, 2014.

[15] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1971–1984, 2008.

[16] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence—video to text," in *Proceedings of the International Conference on Computer Vision*, Santiago, Chile, December 2015.

[17] H. Y. T. Ngan, A. G. O. Yung, and A. G. Yeh, "Outlier detection in traffic data based on the Dirichlet process mixture model," *Iet Intelligent Transport Systems*, vol. 9, no. 7, pp. 773–781, 2015.

[18] R. J. Kingan and T. B. Westhuis, "Robust regression methods for traffic growth forecasting," *Transportation Research Record*, vol. 1957, no. 1, pp. 51–55, 2006.

[19] F. Chang, M. Li, P. Xu, H. Zhou, M. Haque, and H. Huang, "Injury severity of motorcycle riders involved in traffic crashes in Hunan, China: a mixed ordered logit approach," *International Journal of Environmental Research and Public Health*, vol. 13, no. 7, p. 714, 2016.

[20] R. Paleti, N. Eluru, and C. R. Bhat, "Examining the influence of aggressive driving behavior on driver injury severity in traffic crashes," *Accident Analysis & Prevention*, vol. 42, no. 6, pp. 1839–1854, 2010.

[21] J. Gou, W. Qiu, Z. Yi, Y. Xu, Q. Mao, and Y. Zhan, "A local mean representation-based K -nearest neighbor classifier," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 3, pp. 1–25, 2019.

[22] T. Kim, Y. Yue, S. Taylor, and I. Matthews, "A decision tree framework for spatiotemporal sequence prediction," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, August 2015.

[23] T. Lu, Z. Dunyao, Y. Lixin, and Z. Pan, "The traffic accident hotspot prediction: based on the logistic regression method," in *Proceedings of the 2015 International Conference on Transportation Information and Safety (ICTIS)*, Wuhan, China, June 2015.

[24] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008.

[25] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," *The VLDB Journal*, vol. 8, no. 3-4, pp. 237–253, 2000.

[26] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: a partition-and-detect framework," in *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, Cancun, Mexico, April 2008.

[27] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, Beijing, China, June 2007.

[28] I. San Román, I. Martín de Diego, C. Conde, and E. Cabello, "Outlier trajectory detection through a context-aware distance," *Pattern Analysis and Applications*, vol. 22, no. 3, pp. 831–839, 2019.

[29] Q. Yu, Y. Luo, C. Chen, and X. Wang, "Trajectory outlier detection approach based on common slices sub-sequence," *Applied Intelligence*, vol. 48, no. 9, pp. 2661–2680, 2018.

[30] Z. Zhu, D. Yao, J. Huang, H. Li, and J. Bi, “Sub-trajectory-and trajectory-neighbor-based outlier detection over trajectory streams,” in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Melbourne, Australia, June 2018.

[31] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: identifying density-based local outliers,” in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, TX, USA, May 2000.

[32] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Portland, OH, USA, August 1996.

[33] A. Kut and D. Birant, “Spatio-temporal outlier detection in large databases,” *Journal of Computing and Information Technology*, vol. 14, no. 4, pp. 291–297, 2006.

[34] J.-G. Lee, J. Han, X. Li, and H. Gonzalez, “TraClass,” *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 1081–1094, 2008.

[35] G. G. Hazel, “Multivariate Gaussian MRF for multispectral scene segmentation and anomaly detection,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 3, pp. 1199–1211, 2000.

[36] S. A. Shaikh and H. Kitagawa, “Efficient distance-based outlier detection on uncertain datasets of Gaussian distribution,” *World Wide Web*, vol. 17, no. 4, pp. 511–538, 2014.

[37] Y. Yu, L. Cao, E. A. Rundensteiner, and Q. Wang, “Detecting moving object outliers in massive-scale trajectory streams,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, August 2014.

[38] Y.-J. Gong, E. Chen, X. Zhang, L. M. Ni, and J. Zhang, “AntMapper: an ant colony-based map matching approach for trajectory-based applications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 390–401, 2017.

[39] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, December 2008.

[40] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation-based anomaly detection,” *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1–39, 2012.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8868550, 6 pages
https://doi.org/10.1155/2020/8868550

Research Article

Secrecy Performance Analysis of the NOMA System on High-Speed Railway

Wenwu Xie, Jinxia Yang, Xinzhong Liu, Zhihe Yang, Xin Peng, Jianwu Liao, and Tingyu Huang

School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang, Hunan 414006, China

Correspondence should be addressed to Xinzhong Liu; liuxinzhong@hnist.edu.cn

Received 15 August 2020; Revised 20 November 2020; Accepted 8 December 2020; Published 22 December 2020

Academic Editor: Zhe-Li Liu

High-speed railway (HSR) wireless communications are required to ensure strict security. In this work, we study the secrecy performance of a nonorthogonal multiple access- (NOMA-) aided HSR wireless communication system in the case of an eavesdropping user. Specifically, applying NOMA technology to the HSR communication system can effectively improve the data rates. Therefore, we study the secrecy performance of the downlink NOMA system under the HSR wireless communication. In particular, the exact analytical results for the secrecy outage probability (SOP) based on no small-scale channel state information (CSI) are derived. We also provide all of the parameterizations for the proposed channel model. Finally, the correctness of theoretical derivation is verified via simulations. Results show the positive effect of utilizing the NOMA for enhancing wireless systems secrecy performance.

1. Introduction

Recently, the rapid development of HSR has brought great convenience to people's travel. At the same time, high quality and data rate wireless communication are required for passenger services. The high-speed movement of HSR has brought about problems such as the wireless Doppler effect and signal shielding in train compartments, which will cause a series of phenomena, such as the difficulty of a mobile phone call and poor voice signal quality. Thus, railway transportation communication becomes an interesting topic. Researchers have carried out research on improving the data rate on HSR wireless communication in various aspects [1–3]. For example, the traditional multiple antenna and beamforming technology are introduced into the HSR scenario for the first time in [1] where the security capacity and SOP are compared and analyzed. In addition to the security requirements, the high broadband services for passengers are required. Therefore, the authors in [2] proposed the tapped delay line model for a MIMO channel in HSR scenarios. Compared with the existing channel model, the MIMO channel model is efficient and yet flexible in the HSR environment. More recently, to satisfy the fifth generation (5G) on HSR, the authors in [3] focus on the study of channel modeling combined with the 5G technology such as multiple-input multiple-output (MIMO) and millimeter-wave (mm wave). Hence, applying 5G technology to HSR wireless communication is to be a significant work for future wireless systems.

As a major technology of 5G, nonorthogonal multiple access (NOMA) can schedule multiple users with the same time-frequency resources. Combined with the successive interference cancellation (SIC) technology, multiple signals can be sent and demodulated at the same time, which can effectively improve the spectrum efficiency [4]. In [5], the performance of downlink NOMA is studied by calculating the BER of perfect and imperfect SIC conditions; compared to the orthogonal multiple access (OMA) techniques, the NOMA technique can provide better performance gains. Thus, the authors in [6] analyzed and compared the performance of different NOMA schemes applied in HSR scenarios and proved that choosing the right NOMA scheme has a great effect on improving the data rate in HSR. However, in practice, it is challenging to ensure the security of data transmission in HSR wireless communication.

In recent years, based on Wyner's eavesdropping channel model, the main indicators to measure the secrecy of wireless communication systems are secrecy outage probability (SOP) and average secrecy capacity (ASC). In the research of physical layer security, the CSI of each channel affects the secrecy performance of the system. Generally, the sources knew the CSI of the main channel. For the CSI of the eavesdropping channel, if the eavesdropping terminal is active eavesdropping, we use ASC to quantify the security performance of the system. Otherwise, we use SOP to measure communication security [7]. The authors in the literature [8] analyse the secrecy performance of the different systems by calculating SOP and ASC.

In this work, we study the secrecy performance of the downlink NOMA system for HSR. More specifically, we assume that no small-scale CSI of the channels is known; hence, the channels are sorts according to distances between the base station and the legitimate users. Besides, we consider that the system model exists as a static eavesdropper near the base station. To measure the secrecy performance of the system, the paper provides a closed-form expression for the SOP, and the simulation results show the positive effect of applying the NOMA for improving the system secrecy performance for HSR.

2. System Methods

In the HSR scenario, the secure communication mode of the downlink NOMA system is shown in Figure 1. The base station (BS) is located at the track side, and the height of the BS transmitting antenna relative to the horizontal plane is h_z. The legitimate user is the passenger on the HSR. We denote the length of the HSR is L_t, the vertical distance between the base station and HSR is l_z, the distance between the legitimate user and the vertical point of the base station is x, and the distance between the legitimate user and the BS transmitting antenna is d_u. HSR runs away from the BS according to the speed v from the nearest position to the BS. There is a static eavesdropper at the distance d_e, which attempts to obtain the information from the BS. Also, this paper makes the following assumptions: (1) all users and BS are equipped with a single antenna, and the legitimate users communicate directly with the BS; (2) the legitimate users are fixed on the HSR, and the distance between the two users follows the uniform distribution with $L_u/2$ mean; and (3) the legitimate channel is composed of Rician fading and large-scale fading [9].

2.1. Main Link. According to the NOMA scheme, the transmitted signal from BS to all of the legitimate users can be expressed as

$$x = \sum_{l=1}^{M} \sqrt{\alpha_l P_t} x_l, \tag{1}$$

where α_l is the power allocation factor with $\sum_{l=1}^{M} \alpha_l = 1$, set $\alpha_1 < \alpha_2 < \cdots < \alpha_M$, P_t is the transmitting power of the BS, and x_l is the transmission signal of the l^{th} user. Then, the signal received by the k^{th} legitimate user U_k can be expressed as

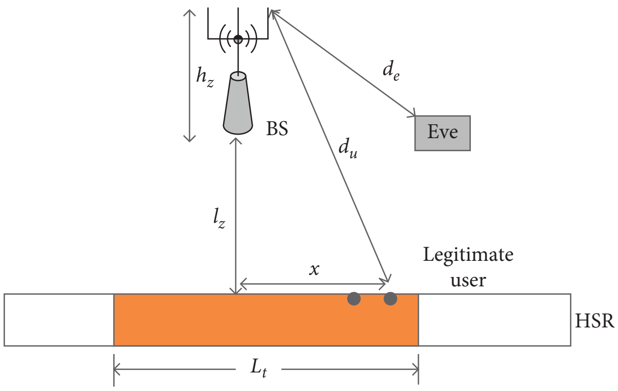


Figure 1: System model.

$$y_k = g_k \sum_{l=1}^{M} \sqrt{\alpha_l P_t} x_l + n_k, \tag{2}$$

where n_k follows the additive white Gaussian noise (AWGN) with the mean zero and variance σ_k^2. For simplicity, the variance is assumed to $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma^2$. The channel gain between the k^{th} legitimate user and the BS can be given by $g_k = h_k \beta_k^{1/2}$, where h_k and $\beta_k$represent small-scale and large-scale fading, respectively, and the expression of β_k is related to the distance d_k. Set $d_1 < d_2 < \cdots < d_M$; according to order statistics, the probability density function (PDF) of d_k from the BS to the k^{th} nearest legitimate user is given by [9]

$$f_{d_k}(x) = k \binom{M}{k} \sum_{j=0}^{M-k} (-1)^j \binom{M-k}{j} \frac{x\left(\sqrt{x^2 - h_z^2 - l_z^2}\right)^{k+j-2}}{L_t^{k+j}}, \tag{3}$$

where $\sqrt{h_z^2 + l_z^2} < x \leq \sqrt{L_t^2 + h_z^2 + l_z^2}$. According to the knowledge of the successive interference cancellation (SIC) receiver [10], for the k^{th} user, the signal-to-interference-plus-noise ratio (SINR) of the k^{th} user can be expressed as

$$\gamma_{D_k} = \frac{\rho \alpha_k |h_k|^2 \beta_k}{\rho |h_k|^2 \beta_k \sum_{l=1}^{k-1} \alpha_l + 1}, \tag{4}$$

where P_t is the transmitted power, and $\rho = P_t/\sigma^2$ is denoted as the average signal-to-noise ratio (SNR). In equation (6), since the small-scale fading channel h_k is the Rician fading channel, we define $\overline{\gamma} = s^2 + 2\lambda_k^2$. Thus, the $|h_k|^2$ is the noncentral chi-square random variable with two degree of freedom, and its PDF and cumulative distribution function (CDF) are given as [11]

$$f_{|h_k|^2}(y) = \frac{(1+K)e^{-K}}{\overline{\gamma}} \exp\left[-\frac{(1+K)y}{\overline{\gamma}}\right] I_0 \cdot \left(2\sqrt{\frac{(1+K)Ky}{\overline{\gamma}}}\right), \quad y > 0, \tag{5}$$

$$F_{|h_k|^2}(y) = 1 - Q_1\left(\sqrt{2K}, \frac{\sqrt{y}}{\lambda_k}\right),$$

where $I_0(x)$ is the modified Bessel function of the zero kind, and $Q_1(a,b)$ is the first order Marcum Q-function [12]. $K = s^2/2\lambda_k^2$ denotes the Rician factor, and similar to [9], we generally set $K = 7\,\mathrm{dB}$ on HSR scenario. Therefore, the conditional CDF of the $F_{|g_k|^2|d_k}(y|d_k)$ can be obtained as

$$F_{|g_k|^2|d_k}(y|d_k) = 1 - Q_1\left(\sqrt{2K}, \frac{\sqrt{\beta^{-1}(d_k)y}}{\lambda_k}\right), \tag{6}$$

where $\beta(d_k) = d_k^{-\chi}$, and χ is the path loss exponent. Since the small-scale fading and the large-scale fading are independent of each other. According to equations (3) and (6), the CDF of $|g_k|^2$ can be written as

$$\begin{aligned} F_{|g_k|^2}(y) &= \int_{\sqrt{h_z^2+l_z^2}}^{\sqrt{L_t^2+h_z^2+l_z^2}} F_{|g_k|^2|d_k}(y|x) f_{d_k}(x)\mathrm{d}x \\ &= \int_{\sqrt{h_z^2+l_z^2}}^{\sqrt{L_t^2+h_z^2+l_z^2}} \left(1 - Q_1\left(\sqrt{2K}, \frac{\sqrt{\beta^{-1}(x)y}}{\lambda_k}\right)\right) k \binom{M}{k} \sum_{j=0}^{M-k} (-1)^j \binom{M-k}{j} \frac{x\left(\sqrt{x^2 - h_z^2 - l_z^2}\right)^{k+j-2}}{L_t^{k+j}} \mathrm{d}x. \end{aligned} \tag{7}$$

Unfortunately, evaluating the integrals in equation (7) is very difficult. Thus, by using the Gauss–Chebyshev quadrature [13], equation (7) can be approximated as

$$\begin{aligned} F_{|g_k|^2}(y) = \sum_{i=1}^{n} \frac{\pi k}{4n} \left(1 - Q_1\left(\sqrt{2K}, \frac{\sqrt{c_1 y}}{\lambda_k}\right)\right) \binom{M}{k} \\ \cdot \sum_{j=0}^{M-k} (-1)^j \binom{M-k}{j} \left(\sqrt{\frac{t_i+1}{2}}\right)^{k+j-2} \sqrt{1-t_i^2}, \end{aligned} \tag{8}$$

where $c_1 = \beta^{-1}(((t_i+1)/2)L_t^2 + h_z^2 + l_z^2)$, $t_i = \cos((2i-1)/2n)\pi$, and n is the approximate order of Chebyshev, which can be selected according to the accuracy and complexity requirements.

Then, the CDF of γ_{D_k} can be readily formulated as

$$\begin{aligned} F_{\gamma_{D_k}}(z) &= P\left(\frac{\rho\alpha_k|g_k|^2}{\rho|g_k|^2\sum_{l=1}^{k-1}\alpha_l + 1} < z_k\right) \\ &= \sum_{i=1}^{n} \left(\frac{\pi k}{4n}\left(1 - Q_1\left(\sqrt{2K}, \frac{1}{\lambda_k}\sqrt{\frac{c_1 z_k}{\rho\left(\alpha_k - \sum_{l=1}^{k-1}\alpha_l z_k\right)}}\right)\right)\binom{M}{k}\sum_{j=0}^{M-k}(-1)^j\binom{M-k}{j}\left(\sqrt{\frac{t_i+1}{2}}\right)^{k+j-2}\sqrt{1-t_i^2}\right). \end{aligned} \tag{9}$$

2.2. Eavesdropping Link. For the eavesdropper, we assume that the eavesdropper attempts to obtain the signal coming from the direct link, and the eavesdropping link is also a hybrid channel similar to the main link. Moreover, the small-scale fading v_k of eavesdropping link experiences independent Rayleigh distribution, and the received signal of an eavesdropper can be expressed as

$$y_E = q_k \sum_{l=1}^{M} \sqrt{\alpha_l P_t} x_l + n_e, \tag{10}$$

where $q_k = v_k \beta_{ek}^{1/2}$ is the channel gain, $\beta_{ek} = d_e^{-\chi}$ represents the large-scale fading, which is related to the distance d_e between the eavesdropper and the BS, and $n_e \sim \mathscr{CN}(0, \sigma_e^2)$ is the AWGN at the eavesdropper. Assuming that the eavesdropper has a strong decoding ability to the transmitted signals, the eavesdropper can decode the mixed signal s to obtain separate signals x_l [14]. Hence, the received SNR of the eavesdropper is given by

$$\gamma_{E_k} = \frac{\alpha_k \rho_e |q_k|^2}{\rho_e |q_k|^2 \sum_{l=1}^{k-1}\alpha_l + 1}, \tag{11}$$

where $\rho_e = P_t/\sigma_e^2$ is the average SNR. Similarly, $|q_k|^2$ is an exponent distribution random variable with parameter η_k. For simplicity, set $\eta_1 = \eta_2 = \cdots = \eta_M$. Then, the PDF of γ_{E_k} can be expressed as

Table 1: Simulation parameter configuration.

Index	Variable/Unit	Value	Description
1	Lt/m	201	Train length
2	Hz/m	30	BS height
3	lz/m	100	Horizontal distance between BS and rail
4	K/dB	7	Rician channel factor
5	fc/MHz	2.5*e*3	Carrier frequency
6	v/Km/h	350	Speed
7	B/MHz	10	Bandwidth
8	Rs/KHz	4.8	Rate
9	kai/dB	3	Attenuation factor
10	SNR_SD	20	Legitimate user SNR
11	SNR_SE	10	Eavesdropping user SNR
12	*M*	2	Number of the legitimate user

$$f_{\gamma_{E_k}}(y)=\begin{cases}\dfrac{\alpha_k}{\eta_k\rho_e\beta_{ek}\left(\alpha_k-\sum_{l=1}^{k-1}\alpha_l y\right)^2}e^{-y/\left(\eta_k\beta_{ek}\left(\alpha_k\rho_e-\rho_e\sum_{l=1}^{k-1}\alpha_l y\right)\right)}, & y<\dfrac{\alpha_k}{\sum_{l=1}^{k-1}\alpha_l},\\ 0, & y\geq\dfrac{\alpha_k}{\sum_{l=1}^{k-1}\alpha_l}.\end{cases}\tag{12}$$

3. Performance Analysis

SOP is a typical performance metrics to analyse the secrecy performance for physical layer secrecy, which is defined that the instantaneous secrecy rate is less than a certain threshold. In this section, we analyse the SOP to determine the security of using NOMA in the HSR scenario. Since the main channel and eavesdropping channel are independent, the SOP of the user U_k can be expressed as [7]

$$\begin{aligned}\mathrm{SOP}&=\Pr\left\{\ln\left(1+\gamma_{D_k}\right)-\ln\left(1+\gamma_{E_k}\right)<C_{\mathrm{th}}\right\}\\&=\int_0^{\infty}F_{\gamma_{D_k}}\left(\theta\gamma_{E_k}+\theta-1\right)f_{\gamma_{E_k}}\left(\gamma_{E_k}\right)\mathrm{d}\gamma_{E_k},\end{aligned}\tag{13}$$

where C_{th} is the target secrecy rate, and $\theta=e^{C_{\mathrm{th}}}$. Then, by using the Gauss–Laguerre [13] and substituting (9) and (13) into (14), we can obtain the approximate expression of SOP as

$$\begin{aligned}\mathrm{SOP}=\sum_{j=1}^{m}\sum_{i=1}^{n}w\left(x_j\right)e^{x_j}\frac{\pi}{4n}&\left(1-Q_1\left(\sqrt{2K},\frac{\sqrt{\beta^{-1}\left(\sqrt{((t_i+1)/2)L_t^2+h_z^2+l_z^2}\right)\left(\theta x_j+\theta-1\right)}}{\lambda_u\sqrt{\rho\left(\alpha_k-\sum_{l=1}^{k-1}\alpha_l\left(\theta x_j+\theta-1\right)\right)}}\right)\right)\\&\times\frac{\sqrt{((t_i+1)/2)L_t^2+h_z^2+l_z^2}}{\sqrt{(t_i+1)/2}}\sqrt{1-t_i^2}\,\frac{\alpha_k}{\eta_k\rho_e\beta_{ek}\left(\alpha_k-\sum_{l=1}^{k-1}\alpha_l x_j\right)^2}e^{-x_j/\left(\eta_k\beta_{ek}\left(\alpha_k\rho_e-\rho_e\sum_{l=1}^{k-1}\alpha_l x_j\right)\right)},\end{aligned}\tag{14}$$

where N is the approximate order terms of Laguerre, $w_j=x_j/\left((m+1)^2\left[L_{m+1}\left(x_j\right)\right]^2\right)$, and $j<33$ is the weight of Laguerre polynomial x_j.

4. Simulation and Analysis

In this section, we present a numerical example to illustrate our analytical results, and the simulation parameters are shown in Table 1 [15].

In Figure 2, we plot the SOP versus transmitted power P_t in the presence of an eavesdropper based on the NOMA scheme on the HSR scenario. It is observed that the analytical results match well with the simulation, which verifies the theoretical derivation. Moreover, increasing the number of P_t results in decreasing SOP for two users. As expected, we note that floors appear at relatively high P_t. Therefore, the SOP of the near user is much better than the far user based on the NOMA scheme. Finally, to improve the performance of SOP, we can increase the transmitting power which is not the only scheme but needs to be optimized together with other schemes to ensure the security of communication.

In Figure 3, we present the SOP curves for the different distances between the eavesdropper and BS. As can be observed, a better secrecy performance will be obtained with the increases of d_e. More specifically, the slopes of performance curves of near users are large than the far users; while $d_e>100$, the change trend of SOP is gentle. Thus, the change

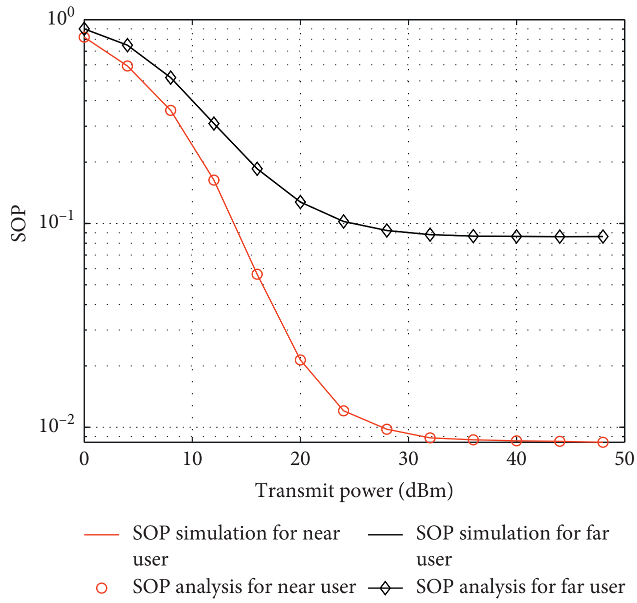


Figure 2: SOP curve of P_t for near user and far user.

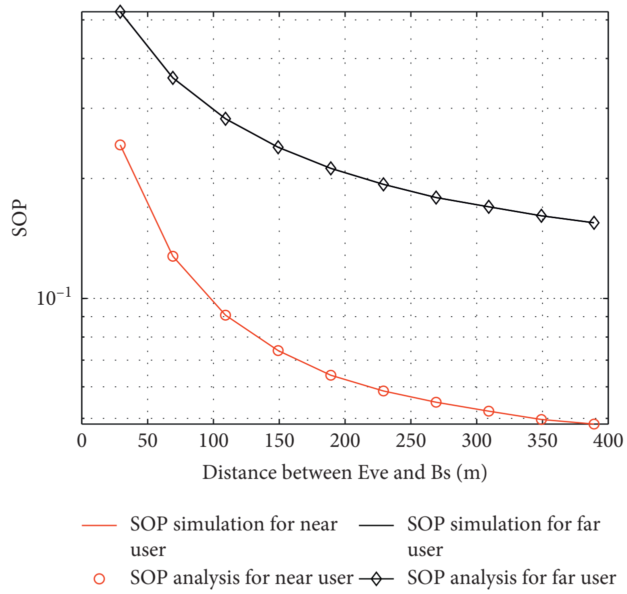


Figure 3: SOP curve of d_e for near user and far user.

of the distance between the eavesdropper and BS has a greater impact on the secrecy performance of the near users than on far users.

In Figure 4, we plot the SOP versus for a different power allocation coefficient α_1 of near user based on the NOMA scheme. It is clearly shown that increasing α_1 near users can significantly improve the near user secrecy performance, and the impact of α_1 on near users is greater than that on far users from the slope of the SOP versus. Thus, it is necessary to select an appropriate α_1 to ensure secure communication in both near user and far user.

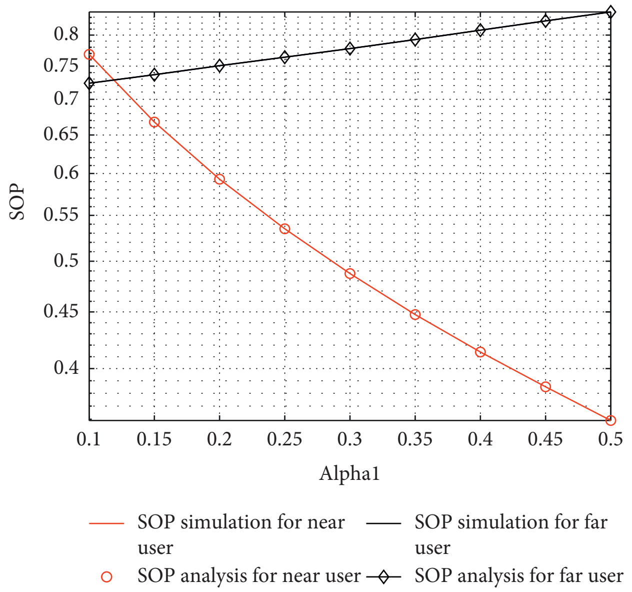


Figure 4: SOP curve of α_1 for near user and far user.

5. Conclusions

In this work, we provide secrecy performance analysis for the HSR scenario in the presence of an eavesdropper with the help of NOMA. More specially, the expression for SOP was derived and verified by simulation. Numerical results showed that the secrecy performance can be improved by choosing the appropriate power allocation coefficient α_1 based on the NOMA scheme.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Wenwu Xie and Xin Peng conceived and designed the study. Xinzhong Liu and Zhihe Yang performed the simulations. Jinxia Yang wrote the paper. All authors reviewed and edited the manuscript. All authors read and approved the final manuscript.

Acknowledgments

This work was in part supported by the National Natural Science Foundation of China (No. 61772195), the Hunan Natural Science Foundation (Nos.2019JJ40043, 2018JJ2154, and 2018JJ2156), the Key Research Foundation of Education Bureau of Hunan Province (No.18A320), the Outstanding Youth Project of Hunan Provincial Education Department (Nos.18B353 and 19K037), the Hunan Emergency Communication Engineering Technology Research Centre (No.2018TP2022), the Hunan Institute of Science and Technology for Postgraduate (No.YCK2020A38), the Hunan

Province for Postgraduate (No.CX20201139), and the National College Student Innovation and Entrepreneurship Training Program (s202010543042).

References

[1] Y. P. Cui and X. M. Fang, "A physical layer secure wireless communication scheme for high speed railway," in *Proceedings of the Sixth International Workshop on Signal Design and Its Applications in Communications*, pp. 114–117, Tokyo, Japan, November 2013.

[2] J. Yang, B. Ai, S. Salous et al., "An efficient MIMO channel model for LTE-R network in high-speed train environment," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3189–3200, 2019.

[3] T. Zhou, H. Li, Y. Wang, L. Liu, and C. Tao, "Channel modeling for future high-speed railway communication systems: a survey," *IEEE Access*, vol. 7, pp. 52818–52826, 2019.

[4] Y. Liang, X. Li, J. Zhang, and Z. Ding, "Non-orthogonal random access for 5G networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4817–4831, 2017.

[5] M. R. Usman, A. Khan, M. A. Usman, Y. S. Jang, and S. Y. Shin, "On the performance of perfect and imperfect SIC in downlink non orthogonal multiple access (NOMA)," in *Proceedings of the International Conference on Smart Green Technology in Electrical and Information Systems (ICSGTEIS)*, pp. 102–106, Bali, Indonesia, October 2016.

[6] D. Feng, "Performance comparison on NOMA schemes in high speed scenario," in *Proceedings of the 2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, pp. 112–116, Chengdu, China, May 2019.

[7] Z. Liao, L. Yang, J. Chen, H.-C. Yang, and M.-S. Alouini, "Physical layer security for dual-hop VLC/RF communication systems," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2603–2606, 2018.

[8] H. Lei, Z. Yang, K.-H. Park et al., "Secrecy outage analysis for cooperative NOMA systems with relay selection schemes," *IEEE Transactions on Communications*, vol. 67, no. 9, pp. 6282–6298, 2019.

[9] J. Fan, J. Zhang, S. Chen, J. Zheng, and B. Ai, "The application of NOMA on high-speed railway with partial CSI," in *Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. 1–5, Honolulu, HI, USA, September 2019.

[10] Y. Liu, Z. Ding, M. Elkashlan, and H. V. Poor, "Cooperative non-orthogonal multiple access with simultaneous wireless information and power transfer," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 938–953, 2016.

[11] M. K. Simon and M.-S. Alouini, *Digital Communication over Fading Channels*, Wiley-Interscience, New York, NY, USA, 2 edition, 2005.

[12] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, Academic, San Diego, CA, USA, 7 edition, 2007.

[13] F. B. Hildebrand, *Introduction to Numerical Analysis*, Courier Corporation, North Chelmsford, MA, USA, 1987.

[14] G. Brante, H. Alves, R. D. Souza, and M. Latva-aho, "Secrecy analysis of transmit antenna selection cooperative schemes with no channel state information at the transmitter," *IEEE Transactions on Communications*, vol. 63, no. 4, pp. 1330–1342, 2015.

[15] F. Hasegawa, A. Taira, G. Noh et al., "High-speed train communications standardization in 3GPP 5G NR," *IEEE Communication Standard. Magazine*, vol. 2, no. 1, pp. 44–52, 2018.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 6611463, 10 pages
https://doi.org/10.1155/2020/6611463

Research Article

A Movie Recommendation System Based on Differential Privacy Protection

Min Li,[1] **Yingming Zeng**,[2] **Yue Guo**,[1] **and Yun Guo**[1]

[1]*College of Cyber Science, Nankai University, Tianjin 300017, China*
[2]*Hangtian INC, Beijing 100140, China*

Correspondence should be addressed to Yun Guo; guoyun@nankai.edu.cn

Received 19 October 2020; Revised 19 November 2020; Accepted 3 December 2020; Published 16 December 2020

Academic Editor: Hao Peng

In the past decades, the ever-increasing popularity of the Internet has led to an explosive growth of information, which has consequently led to the emergence of recommendation systems. A series of cloud-based encryption measures have been adopted in the current recommendation systems to protect users' privacy. However, there are still many other privacy attacks on the local devices. Therefore, this paper studies the encryption interference of applying a differential privacy protection scheme on the data in the user's local devices under the assumption of an untrusted server. A dynamic privacy budget allocation method is proposed based on a localized differential privacy protection scheme while taking the specific application scene of movie recommendation into consideration. What is more, an improved user-based collaborative filtering algorithm, which adopts a matrix-based similarity calculation method instead of the traditional vector-based method when computing the user similarity, is proposed. Finally, it was proved by experimental results that the differential privacy-based movie recommendation system (DP-MRE) proposed in this paper could not only protect the privacy of users but also ensure the accuracy of recommendations.

1. Introduction

With the development of information technology, tons of data are piled up on the Internet and users have many ways to access these data. For the users, what they spend most of their time on is no longer where to get information, but to find out what they are really interested in among numerous information. Therefore, the recommendation system came into being as an inevitable product of this era of big data. However, a key factor that usually influences the performance of recommendation systems is whether the amount of user data is enough or not and that may lead to a high risk of privacy leakage. In 2013, LG Corporation was charged for illegal collection of user data via smart TVs, which reflects the increasing awareness of privacy protection among users. What is more, IoT devices such as WiFi fingerprint which are frequently used in our daily life are also facing many kinds of security attacks [1, 2]. However, most of the existing recommendation systems [3–6] are developed based on the assumption of trusted servers. In most commonly used collaborative filtering algorithms, a trusted server collects all user data and makes user behavior analysis to give out personalized recommendations.

The application of differential privacy protection scheme in recommendation systems was first proposed by McSherry et al. [7, 8]. In their scheme, the server is responsible for encrypting user data, and random noise is added to each step of aggregation in the recommendation system. In such privacy protection schemes, only the circumstances that user data were published to a third-party from a trusted server were considered. However, other circumstances, such that when user data are uploaded from the local device to the cloud, attackers may eavesdrop on the transmission channel and launch a Man-in-the-Middle (MITM) attack or the attackers may directly hack into the cloud server and get access to sensitive user data, are not taken into consideration. Therefore, we reach our research question that how to apply differential privacy protection on users' local data under the basic assumption of an untrusted server. In this paper, existing differential privacy protection schemes and commonly used recommendation algorithms are reviewed, and the application of localized differential privacy protection scheme in recommendation systems to solve the security issue in recommendation algorithms is investigated. The main contributions of this work are summarized as follows:

(i) A privacy budget allocation scheme that can dynamically allocate privacy budget is proposed based on the localized differential privacy protection. In this allocation scheme, users' behaviors such as movie watching records are allocated to the nodes in the privacy prefix tree with equal probability. After that, Laplace noise is added according to the privacy budget allocated to each node. This scheme could avoid the extreme circumstances of unevenly distributed privacy budget and added noise. In the meantime, this allocation scheme could also ensure the security of users' private data, as well as guaranteeing the accuracy of recommendation results by recording the combinatorial sequences of user behavior.

(ii) The traditional user-based collaborative filtering recommendation algorithm is improved by taking the specific application scene of movie recommendation into consideration. During the process of calculating user similarity to find out a similar group of the target users, a matrix-based method is proposed to replace the traditional vector-based method. More specifically, after the privacy prefix tree is generated, we construct a user-interest matrix E according to users' movie watching records and the characteristics of combinatorial sequences, then apply the user-based collaborative filtering recommendation algorithm with matrix E to calculate the similarity between users and find out the similar group of the target user, and finally give out the recommendation results.

2. Theoretical Basis

2.1. Differential Privacy. In 2006, Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith introduced the concept of differential privacy [8–12], which assumes that the attackers are able to access all information except the target information and makes it hard for attackers to access users' privacy via difference calculation. For the calculation result of the dataset, whether a single record is in the dataset or not has a negligible impact on the result. The basic definitions and properties of differential privacy involved in this paper are as follows.

Assume that datasets D_1 and D_2 have the same property structure, the symmetric difference between them is denoted as $D_1 \Delta D_2$, and the number of records in $D_1 \Delta D_2$ is denoted as $|D_1 \Delta D_2|$. If $|D_1 \Delta D_2| = 1$, we say that D_1 and D_2 are adjacent datasets.

Definition 1 (ε-differential privacy). Let ε be a positive real number and M be a random algorithm that takes a dataset as input. Let $M(x)$ denote the result obtained from a query of random algorithm M and R be a subset of $M(x)$. The algorithm M is said to provide ε-differential privacy if, for all adjacent dataset pairs of D_1 and D_2 that differ on a single element and all subsets R of $M(x)$, the following equation is satisfied:

$$\Pr[M(D_1) \in R] \le e^{\varepsilon} \times \Pr[M(D_2) \in R]. \quad (1)$$

Definition 2 (global sensitivity). For query function $f\colon D \longrightarrow R^d$, where D is a dataset and R^d is a d-dimensional vector of real numbers representing the query result, the global sensitivity of f over all adjacent dataset pairs of D_1 and D_2 is described by

$$\mathrm{GS}_{f(D)} = \max \| f(D_1) - f(D_2) \|. \quad (2)$$

Global sensitivity describes the maximum range of changes when a query function is performed on a pair of adjacent datasets. It has nothing to do with the dataset, but it is only determined by the query function itself. The global sensitivity of the counting query is 1.

Property 1 (Sequential composition). Assume that there are n independent algorithms $M_1, M_2, \ldots, M_N$ whose privacy guarantees are $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$, respectively. Then for the same dataset D, the composite algorithm $M(M_1(D), M_2(D), \ldots, M_n(D))$ is $(\sum_{i=1}^{n} \varepsilon_i)$-differentially private.

Definition 3 (The Laplace mechanism). The Laplace mechanism adds Laplace noise to the original query outputs to realize ε-differential privacy. The noise is from Laplace distribution Lap (σ) that can be expressed by the following probability density function with mean value 0 and scale parameter:

$$p(x) = \frac{1}{2\sigma} \exp\left(-\frac{|x|}{\sigma}\right). \quad (3)$$

2.2. Privacy Prefix Tree. The movie recommendation system based on differential privacy protection that we proposed in this paper combines users' movie watching records with the characteristic structure of prefix tree [13] to construct a privacy prefix tree (DP-tree), which can be considered as an improved prefix tree, and its structure is shown in Figure 1.

In Figure 1, Prior is a pointer point to the parent node; Value stores the value; Num is the number of times that this value shows; Depth is the depth of value; Child[i] is an array of pointers that point to the child nodes, and EndNum stores the number of times that the current node is the end of each path. The genres of all the movies that a user has watched are recorded and abstracted to a privacy prefix tree with a root node denoted as *Root*, in which each node represents a genre. In the privacy prefix tree, each branch is actually a sequence of the combination of different tags that represent different movie genres, and each sequence is started with node *Root*. The identical subsequences are merged and the

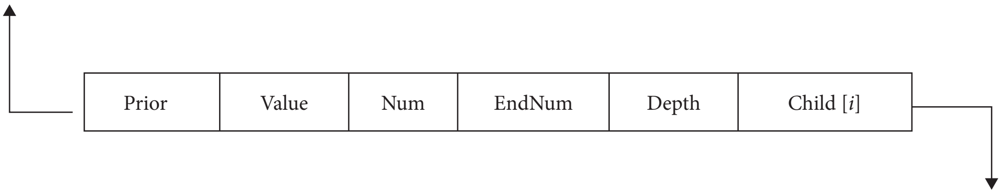


Figure 1: Data structure of a privacy prefix tree.

number of times that the subsequence shows is accumulated. Finally, the frequency that each genre of movie is watching as well as the frequency that each movie genre sequence shows is also recorded.

Based on the construction of the privacy prefix tree, the movie recommendation system proposed in this paper decomposes the record of user behavior, allocates privacy budget dynamically for the privacy prefix tree, and adds Laplace noise that satisfies the Laplace distribution. After that, a user-interest matrix E is constructed according to the appearance frequency of different movie genres and the movie genre sequences that we get from the privacy prefix tree. Finally, matrix similarity is calculated to find out the similar user group of the target user, and a user-based collaborative filtering algorithm is adopted to give out a recommendation of movies.

3. Design of the Differential Privacy Protection Scheme

3.1. Principal Steps in Differential Privacy Protection Scheme. There are two principal steps when designing a differential privacy protection scheme: firstly, select appropriate privacy budget parameters and allocate a proper privacy budget for the protected data; secondly, add some noise interference to the protected data.

For the noise addition of the counting query, the Laplace mechanism is adopted to add interference to the privacy data, and the size of noise is closely related to the result of privacy budget allocation. More precisely, the privacy budget ε is inversely proportional to the size of the added noise. Therefore, the privacy budget ε not only determines the level of differential privacy protection but also influences the addition of noise interference; that is why ε is the core parameter in differential privacy protection scheme. In this paper, we will mainly focus on how to allocate the privacy budget appropriately.

For the movie recommendation system based on differential privacy protection, firstly, a privacy prefix tree movie genre is constructed according to users' watching history. Movie genres and sequences that appear more frequently in the privacy prefix tree are more likely to arouse users' interest, and they also have a higher possibility of being attacked. In order to prevent the privacy budget from being exhausted, we usually allocate more privacy budgets for the data that are commonly used. However, the traditional privacy budget allocation method which evenly allocates the privacy budget to each node or each layer of the privacy prefix tree will lead to unreasonable addition of noise interference. What is more, limited privacy budget allocation for commonly used data may lead to quick exhaustion of the total budget, which will undermine the protection of users' privacy. Therefore, the problem of how to allocate the privacy budget reasonably is worth further investigation. In this paper, we proposed a scheme based on prefix tree allocation that can allocate the privacy budget ε dynamically and reasonably according to the frequency of data use.

3.2. Prefix Tree Privacy Budget Allocation Scheme. The film recommendation system based on differential privacy introduced in this paper is based on the tree structure for data protection and encryption. Figure 2 shows the structure of user information based on the prefix tree structure; the genres are extracted as a movie feature and a privacy prefix tree is constructed based on the prefix tree structure. Specifically speaking, the genres (types) are extracted from users' watching records and stored in sequences in the substructure of a tree, where each path represents a certain combination of movie types and then records the showing frequency of each child node as well as the frequency of them appearing as leaf nodes. In order to reasonably allocate the privacy budget, we assign the privacy budget for each node in the privacy prefix tree proportionally. In particular, the root node R is abstract and does not represent a real movie type, so it will not consume any privacy budget. All other nodes in subtrees need to be assigned a privacy budget.

Instead of storing the movie type directly in the prefix tree, the corresponding letter representation of the movie type is stored, as shown in Table 1.

Table 2 shows the data stored in each node in the privacy prefix tree structure shown in Figure 2.

As shown in Figure 3 and Table 2, the first path represents that the times (counts) of user watching movies that are tagged with a are 10, b is 5, and the end number is 2, which means that the user has watched 3 movies that are depicted by sequence <a, b>, and similarly, we can tell that he or she has also watched 2 movies that are depicted by sequence <a, b, c>.

As shown in Figure 3, assuming that the total privacy budget of the tree is ε, start with the first level of this tree; the frequencies of movie types a, $d2$, and f are 10, 6, and 4, respectively. Therefore, the total privacy budget allocation proportion of the subtree with node a as its root node should be $(10/20)\varepsilon$; thus, the dashed box shown in Figure 3 should totally be assigned 0.5ε privacy budget. Similarly, $\varepsilon_b =(0.5 * 0.5 * 0.6)/2\varepsilon$. When a movie type appears in different sequences, the privacy budget of it equals the total

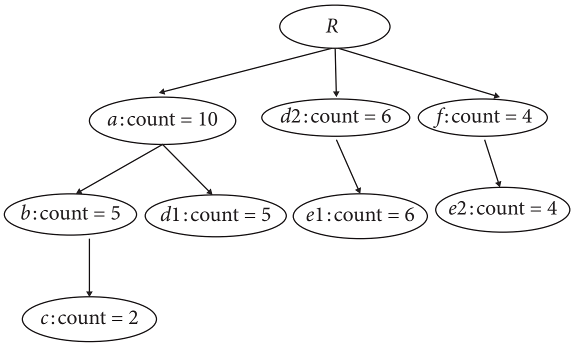


Figure 2: User information diagram based on the prefix tree structure.

Table 1: Mapping table of movie genres to tags.

Genre	Love	Suspense	Action	Comedy	Plot	Tragedy
Tag	*a*	*b*	*c*	*d*	*e*	*f*

Table 2: DP-tree data structure.

Prior	Value	Num	EndNum	Depth	Child[*i*]
—	*R*	—	—	0	[*a*, *d*2, f]
R	*a*	10	0	1	[*b*, *d*1]
A	*b*	5	3	2	[*c*]
B	*c*	2	2	3	—
A	*d*1	5	5	2	—
R	*D*2	6	0	1	[*e*1]
*d*2	*E*1	6	6	2	—
F	*E*2	4	4	2	—
R	*f*	4	0	1	[*e*2]

sum of the allocated privacy budget in each sequence. For example, the privacy budget of movie type d is $\varepsilon_d = \varepsilon_{d1} + \varepsilon_{d2} = 0.125\varepsilon + 0.15\varepsilon = 0.275\varepsilon$. According to Property 1, the sequential composition property of the differential privacy protection, it can be concluded that

$$\varepsilon = \varepsilon_a + \varepsilon_b + \cdots + \varepsilon_f. \tag{4}$$

It can be seen that, compared with other privacy budget allocation methods [14–18], the method of allocating privacy budget is based on the value of each node in the prefix tree, instead of just allocating uniformly according to the level structure. This allocation method can allocate the privacy budget reasonably and dynamically in the case that big differences exist among structures of the subtrees, and it also eliminates the requirement of artificially adjusting the value of privacy budget allocation.

3.3. Prefix Tree Privacy Budget Allocation Algorithm. The privacy budget allocation algorithm based on the prefix tree is shown as follows. *TMovie* stores the result of privacy budget allocation of movie type nodes; DP-tree movie type node v and its privacy ε_v are stored as $<v, \varepsilon_v>$ in the queue set *TQueue*; *Pv* is the statistical frequency of the current node v being watched by users; *GetTop (LinkQueue* Q, *string r*, and *float e)* represents the dequeue function of header element (Algorithm 1).

In the above algorithm, the *TMovie* and *TQueue* sets are initialized to be empty after inputting the privacy budget ε, and the prefixed prefix tree and root node R are constructed. Then, add the current node and its privacy budget to *TQueue* (when R is not the root), and compare the weight of the current node with its parent node. If their weights are equal, assign half of the current privacy budget for both of them. Otherwise, compare the current node with its brother nodes and assign half of parent nodes' privacy budget to them according to their weight ratio. Repeat this process for each child node of the current node.

4. Design of DP-MRE

4.1. Overall Framework of DP-MRE. Figure 4 is the overall frame diagram of the movie recommendation system based on differential privacy protection, where the overall system is composed of five components. Firstly, users' private data are collected on their local devices, and then a prefix tree is constructed based on the collected data to dynamically allocate the privacy budget. Next, noise interference that obeys Laplace distribution is added, and then the users' data after being interfered with as well as public data are used together as the input of recommendation system and finally give out movie recommendations. The detailed meaning of each component in Figure 4 is as follows:

Public data refer to the public information related to users' private data from internal or external resources. We chose the MovieLens 1M dataset, which contains 100 million ratings from 6,000 users on nearly 4,000 movies. This dataset will be used as an experimental dataset and test dataset for experimental verification in this paper.

User data refer to the historical data of users' behavior collected from their personal devices. In this paper, we used the historical records of movies watched by users, such as the frequency of a user watching a certain type of movie, as well as users' ratings on these movies. What is more, this part of data is not interfered with.

Privacy quantification refers to the process that constructs the privacy prefix tree according to users' behavior records and allocates privacy budget according to the appearing times and frequencies of each node in the privacy prefix tree that we proposed in this paper.

Data perturbation refers to the process that adds noise which obeys Laplace distribution to each node in the privacy prefix tree according to its privacy budget, in order to interfere with the original data to ensure the security of users' private data while preserving the effectiveness of data. In other words, the interfered data should satisfy two necessary conditions: being secure enough to protect users' privacy and being effective enough to give out accurate recommendation in the subsequent recommendation stage.

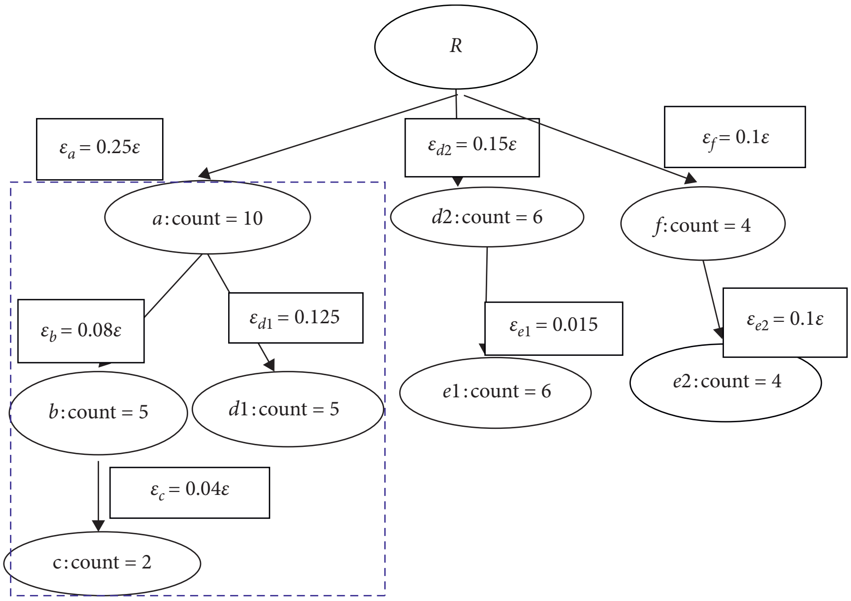

R
$\varepsilon_a = 0.25\varepsilon$
$\varepsilon_{d2} = 0.15\varepsilon$
$\varepsilon_f = 0.1\varepsilon$
a: count = 10
$d2$: count = 6
f: count = 4
$\varepsilon_b = 0.08\varepsilon$
$\varepsilon_{d1} = 0.125$
$\varepsilon_{e1} = 0.015$
$\varepsilon_{e2} = 0.1\varepsilon$
b: count = 5
$d1$: count = 5
$e1$: count = 6
$e2$: count = 4
$\varepsilon_c = 0.04\varepsilon$
c: count = 2

FIGURE 3: Privacy budget allocation scheme based on the prefix tree.

```
Input: Privacy budget $s$, prefix tree DP-Tree, root node R
Output: Privacy budget allocation results set TMovie
Initialize set TMovie and TQueue to 0
If ($R$ = = ' ')
    $\varepsilon_R = 0$
    R $\longrightarrow$ child(R)
Else
    Add the current node <$R$,$\varepsilon_R$> to TQueue
    While TQueue $\neq$ NULL Do
        GetTop(TQueue,R $\varepsilon_R$)
        IF R$\in$ TMovie Then
            $\varepsilon_R \longleftarrow$ privacy budget for node $R$ in TMovie
            TMovie $\leftarrow$ <R,$\varepsilon_R + \varepsilon_{P_R}$>
        Else
            TMovie $\leftarrow$ <R,$\varepsilon_{P_R}$>
        End If
        If ($P_R = P_{R\text{-parent}}$)
            $\varepsilon \leftarrow \varepsilon/2$
        Else
            $\varepsilon \leftarrow (\varepsilon - \varepsilon_{P_R})/2$
        For $v$ (child node of the current node)
            $P_v \leftarrow$ frequency of watching movies with tag v
            Append <$v$, $\varepsilon_{P_v}$> to TQueue
        End For
    End while
```

ALGORITHM 1: Privacy budget allocation algorithm.

Recommendation refers to the final stage of our DP-MRE system design, in which an untrusted third-party server obtains the data after perturbation, that is, after adding Laplace noise, and then uses these data to build a user-interest matrix according to user's preference on different types of movie. Next, similarity calculation based on the multidimensional matrix is performed to find out similar user groups, and a user-based collaborative filtering algorithm is adopted to give out a final recommendation for users.

4.2. User-Based Collaborative Filtering Algorithm. The user-based collaborative filtering recommendation algorithm [19–21] is usually composed of two parts: (1) to calculate the user similarity; (2) to recommend the interested contents of similar user groups to the target user.

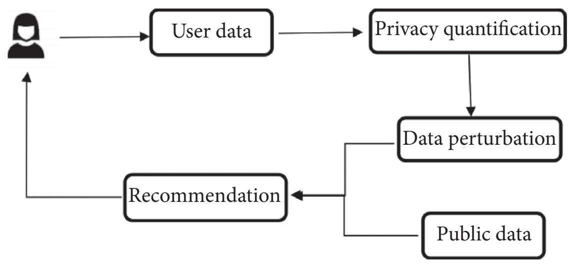


Figure 4: Frame diagram of the movie recommendation system based on differential privacy protection.

This paper extends the traditional method of computing vector-based similarity to matrix-based similarity and further combines the watching frequency of movie types as well as the combinatorial sequence of movie types. The specific method is to construct an N ∗ N user-interest matrix E with movie type as both horizontal and vertical quantities. For example, assume that $a{\sim}n$ represent movie types and the user-interest matrix E is constructed as follows:

$$E = \begin{pmatrix} p_{aa} & q_{ab} & \cdots & q_{an} \\ q_{ba} & p_{bb} & \cdots & q_{bn} \\ \vdots & \vdots & \ddots & \vdots \\ q_{na} & q_{nb} & \cdots & p_{nn} \end{pmatrix}, \qquad (5)$$

where the diagonal of the matrix, that is, the set $P = \{p_{aa}, p_{bb}, \ldots, p_{nn}\}$, represents users' rating scores on movie type $a \sim n$; other quantities q_{mn} represent users' rating score on certain movie type sequences. For example, $p_{aa} = 3, p_{bb} = 2, q_{ab} = 2$ indicate that the user has an interest score of 3 for type a movies, 2 for type b movies, and 2 for <a,b> sequence.

As shown in Figure 3, the privacy prefix tree is constructed from user A's movie watching record. According to the values of each node in the prefix tree and the sequence relationship between movie types in Figure 3, the user-interest matrix of user A can be constructed as follows:

$$E = \begin{pmatrix} 10 & 5 & 2 & 5 & - & - \\ - & 5 & 2 & - & - & - \\ - & - & 2 & - & - & - \\ - & - & - & 11 & 6 & - \\ - & - & - & - & 10 & - \\ - & - & - & - & 4 & 4 \end{pmatrix}. \qquad (6)$$

After constructing the user-interest matrix, the similarity between users can be obtained via matrix similarity calculation. In this paper, the correlation coefficient is used to evaluate the similarity of two matrices. The correlation coefficient is an indicator used to measure the statistical relationship between two variables, and it is a ratio, which can also be regarded as a special form of covariance after the standardization that eliminates the impact of the variation of amplitude. The correlation coefficient could be either positive or negative, which represents the direction of correlation between two variables but does not change the degree of similarity. In other words, the degree of similarity between two variables is reflected by the absolute value of the correlation coefficient. The correlation coefficient used in this paper is calculated as follows:

$$r = \frac{\sum_m \sum_n \left(A_{mn} - \overline{A}\right)\left(B_{mn} - \overline{B}\right)}{\sqrt{\left(\sum_m \sum_n \left(A_{mn} - \overline{A}\right)^2\right)\left(\sum_m \sum_n \left(B_{mn} - \overline{B}\right)^2\right)}}, \qquad (7)$$

where $\overline{A} = \text{mean}(A), \overline{B} = \text{mean}(B)$, matrix A and B are two matrices with the same size, $\overline{A}$ and $\overline{B}$ represent the mean value matrix of A and B, respectively, and r denotes the correlation coefficient which ranges in [1, +1]. It indicates that matrices A and B share high similarity when the absolute value of r is close to 1, and when r is close to 0, it indicates that matrices A and B are less similar.

The similarity of the rating scores on movie type *(a~c)* between *UserA* and *UserB~UserE* is calculated using the method described above in this section, and the results are as shown in Table 3.

According to the calculation results based on matrix similarity in Table 3, the similarity between *UserA* and *UserE* is the highest. However, if we change to use the Pearson correlation coefficient to evaluate the similarity between users, although the structure of the user-interest matrix of *UserA* and *UserE* shares the highest similarity, the common rating items of *UserA* and *UserC* will lead to the calculation result of the similarity between *UserA* and *UserC* being exactly 1, which is not consistent with the real situation. However, in the matrix-based similarity calculation method we proposed, the similarity between *UserA* and *UserE* is a little higher than that between *UserA* and *UserD*. Therefore, both the absolute value and the quantity structure of the matrices are taken into account in the method we proposed based on matrix similarity.

What is more, if we change to use the Euclidean distance to evaluate the similarity between users, if there are no common rating items between two users, the similarity it gives out would be relatively low even if the structure and value are highly similar to each other. For example, in Table 3, the similarity between *UserA* and *UserB*, *UserC*, and *UserE* is all relatively low. When calculating the similarity of matrices, we can easily notice that actually *UserA* and *UserE* have high similarity, and their similarities to *UserB* and *UserC* are also higher than the results given by Euclidean distance calculation.

Assuming that the number of users who need personalized recommendation is u and the similar group of the target user is K, use *S*(u,*K*) to denote the process of selecting items that user u interested in from similar group K, denote the interest rating score of user v to item j as r_{vj} and similarity between the interest of user u and user v as w_{uv}, and denote the user group who are interested in item j as *N*(*j*). Then, the interest rating score of user u to item j should be given by equation (8):

$$p(u, j) = \sum_{v \in S(u,K) \cap N(j)} w_{uv} \times r. \qquad (8)$$

Table 3: User similarity calculated based on matrix similarity.

User	User-interest matrix	Similarity to *UserA* (absolute value)
UserA	$\begin{pmatrix} 5.0 & 3.0 & 0 \\ 0 & 3.0 & 0 \\ 0 & 0 & 2.5 \end{pmatrix}$	1.000
UserB	$\begin{pmatrix} 2.0 & 0 & 0 \\ 0 & 2.5 & 0 \\ 0 & 2.5 & 5.0 \end{pmatrix}$	0.546
UserC	$\begin{pmatrix} 2.5 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	0.329
UserD	$\begin{pmatrix} 5.0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3.0 \end{pmatrix}$	0.875
UserE	$\begin{pmatrix} 4.0 & 2.0 & 0 \\ 0 & 3.0 & 0 \\ 0 & 0 & 2.0 \end{pmatrix}$	0.977

After calculating *p (u,j)*, compare the value *p (u,j)* between different users. If two users have a similar value of *p (u, j)*, it indicates that they share a similar interest in a particular item. Then, the recommendation results could be given out by sorting the values from largest to smallest and selecting the highest-ranked items.

4.3. Analysis of Privacy Security. In this section, the privacy security of the DP-MRE algorithm proposed in this paper is analyzed based on differential privacy protection. Let D1 and D2 be the adjacent dataset (i.e., d $(D_1, D_2) = 1$), $f(D_i)$ denotes the category set of users' private data, C denotes the size of the public movie set, *j* denotes the users' private data, and z(j) is the size of the Laplace noise added to movie type *j*. From the definition of differential privacy, we can know that, for arbitrary $r = (r_1, \ldots, r_c) \in \mathrm{Range}(\mathrm{DP} - \mathrm{MRE})$, if the algorithm DP - MRE satisfies

$$\Pr[\mathrm{DP} - \mathrm{MRE}(D_1) = r] \leq e^{\varepsilon} \times \Pr[\mathrm{DP} - \mathrm{MRE}(D_2) = r], \tag{9}$$

or if the algorithm DP-MRE satisfies

$$\frac{\Pr[\mathrm{DP} - \mathrm{MRE}(D_1) = r]}{\Pr[\mathrm{DP} - \mathrm{MRE}(D_2) = r]} \leq e^{\varepsilon}, \tag{10}$$

then we can conclude that the algorithm DP-MRE satisfies the *ε*-differential privacy protection.

According to the differential privacy protection proposed in this paper, the differential privacy protection is carried out on users' local private devices, so the privacy protection analysis only focuses on the steps of privacy budget allocation and noise addition, while there is no privacy leakage problem in the user similarity calculation and recommendation steps. Therefore, privacy security analysis can be performed in the privacy budget allocation and noise addition steps as follows:

$$\begin{aligned}\frac{\Pr[\mathrm{DP} - \mathrm{MRE}(D_1) = r]}{\Pr[\mathrm{DP} - \mathrm{MRE}(D_2) = r]} &= \prod_{j\in C}\frac{\Pr[\mathrm{DP} - \mathrm{MRE}(D_1)(j) = r(j)]}{\Pr[\mathrm{DP} - \mathrm{MRE}(D_2)(j) = r(j)]}\\ &\geq \exp\left(-\sum_{j\in C}\frac{1}{z(j)}\left|f_j(D_1) - f_j(D_2)\right|\right)\\ &\geq \exp\left(-\max_{d(D_1,D_2)=1}\sum_{j\in C}\frac{1}{z(j)}\left|f_j(D_1) - f_j(D_2)\right|\right) \geq e^{-\varepsilon}.\end{aligned} \tag{11}$$

In the first step, according to the sequential composition property of difference privacy, the noise is added to each category set independently; thus, the difference in privacy remains unchanged. Furthermore, the second step can be derived from the added Laplace noise and triangle inequality. Therefore, we have proved that the DP-MRE algorithm satisfies Inequality 11.

5. Experimental Results and Analysis

5.1. Privacy Budget Allocation. The key point in the application of differential privacy protection algorithm is to preserve users' privacy as well as the effectiveness of data in the meantime. On one hand, users' privacy is ensured by the differential privacy protection mechanism, which is realized by adding the noise satisfying Laplace distribution to users' personal data. On the other hand, the effectiveness means the property of data that it can still be analyzed and processed after being protected by a differential privacy scheme, and the analysis results should be relatively accurate. At the same time, to allocate the privacy budget reasonably should also be taken into consideration when designing a differential privacy protection scheme.

In order to evaluate the effectiveness of the prefix tree privacy budget allocation method proposed in this paper, the query error of each node in the tree structure is analyzed, and it is compared with the traditional allocation method which allocates the privacy budget uniformly or proportionally according to arithmetic or geometric series. Mean square error is adopted to evaluate the query error. Assume that the accurate value of a set of data is given by $(a_1, a_2, \ldots, a_n)$ and the approximate value is given by $(a_1', a_2', \ldots, a_n')$. Then, the mean square error (MSE) is given by equation (12).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(a_i' - a_i\right)^2. \tag{12}$$

MovieLens 1M dataset, which contains 6,000 user ratings on nearly 4,000 films, was used and we designed a query for the Movies dataset and repeated the query *n* times ($n = 10, 20, \ldots, 1000$) to obtain the mean square error value generated by these *n* queries. In order to get a more accurate result and to avoid the extreme circumstance that the randomness of noise may lead to, the calculation of mean square error is repeated d rounds ($d = 100$); for each round, the mean square error is denoted as MS ($i = 1, 2, \ldots, d$). Thus, we could get the average of the mean square error $\overline{\text{MSE}}$. The greater value of $\overline{\text{MSE}}$ reflects the larger noise and correspondingly infers a lower accuracy of query results. The calculation method of ($i = 1, 2, \ldots, d$) and $\overline{\text{MSE}}$ is shown in equation (11) and (12).

$$\begin{aligned} \text{MSE} &= \frac{1}{n}\sum_{j=1}^{n}\left(y_j - x_j\right)^2, \\ \overline{\text{MSE}} &= \frac{1}{d}\sum_{i=1}^{d}\text{MSE}. \end{aligned} \tag{13}$$

Denote the query defined on Movies dataset as f, x_j is the result of the j-th query on f, and y_j is the corresponding noise result.

As can be seen from Figure 5, under repeated attacks, the errors generated by all privacy budget allocation schemes are increasing. The traditional allocation method which evenly allocates privacy budget to each layer generates the largest error, which indicates that this method produces the largest error in the case of uneven distribution of tree structure. In the cases when the number of queries is relatively small, the error between privacy budget allocation schemes based on arithmetic difference and arithmetic ratio is not much different from that based on the prefix tree structure. However, with the increase of the number of queries, the noise error generated under the privacy budget allocation based on the prefix tree is lower than other methods. The results indicate that when the number of queries is relatively small, all privacy budget allocation schemes produce relatively similar errors, except the scheme that evenly allocates privacy budget based on layers. However, when the number of queries is large enough, the privacy budget allocation scheme based on the prefix tree performs better than all the other schemes.

5.2. Performance of DP-MRE. In order to reflect the impact of differential privacy on the recommendation quality of the recommendation system (DP-MRE) in this paper, we use precision and recall to evaluate the performance of the recommendation system. Precision and recall are two indicators that are commonly used to evaluate the efficiency and quality of information retrieval systems with chaotic data. Both of these two indicators range from 0 to 1. The closer their value is to 1, the higher the quality of the system is, in other words, the higher the accuracy of the results given out by the information retrieval system is. Precision is defined according to the prediction results, which indicates how many of the samples whose predictions are positive are truly positive, whereas recall is defined according to our original samples, which indicates how many positive samples are predicted correctly as positive. The definition of precision and recall in a recommendation system is shown as follows:

$$\begin{aligned} \text{precision} &= \frac{\text{\# of effective recommended sets}}{\text{\#of total recommended sets}}, \\ \text{recall} &= \frac{\text{\# of effective recommended sets}}{\text{\# of total tested sets}}. \end{aligned} \tag{14}$$

In order to objectively analyze the feasibility and effectiveness in the film recommendation system of DP-MRE algorithm based on differential privacy protection proposed in this paper, it is compared with the S-DPDP algorithm based on differential privacy protection proposed by Shen et al. We set the difference privacy parameter ε as an independent variable, took different values for the privacy budget parameter in the experiment, and controlled a single variable to compare multiple recommendation algorithms. In addition, in order to more intuitively reflect the impact of privacy protection on the overall recommendation algorithm, this paper also added the data recommendation algorithm *Baseline* without privacy protection scheme into comparison. Therefore, two algorithms with differential privacy protection scheme, S-DPDP and DP-MRE algorithm, as well as an algorithm without privacy protection are taken into comparison.

Figure 6 shows the impact of differential privacy protection on the precision of the recommendation system. From the experimental results, we could see that, for the recommendation system without privacy protection, the precision of the user-based collaborative filtering recommendation system is about 0.53, and differential privacy protection algorithms DP-MRE and S-DPDP indeed cause a certain degree of loss in recommendation precision. When the differential privacy parameter ε is close to 1, the precision of DP-MRE and S-DPDP algorithm recommended is about 0.51. With the increase of the privacy parameter ε, the precision of DP-MRE and S-DPDP algorithms gradually increases to that of *Baseline* algorithm. Compared with S-DPDP, DP-MRE has a smaller loss of precision, since DP-MRE allocates the privacy budget according to the DP-tree structure, which maintains the type combination sequence and frequency characteristics of the movies watched by users and distributes the Laplace noise reasonably, therefore reducing the loss of recommended quality caused by noise addition. However, S-DPDP adopted an iterative algorithm to add noise, which blurs the similarity between users. Therefore, from the perspective of recommendation quality loss, DP-MRE performs better than S-DPDP algorithm, whereas DP-MRE has a higher time complexity in the privacy budget allocation process, which affects the overall system efficiency.

Figure 7 shows the impact of differential privacy protection on the recall rate of the recommendation systems. From the experimental results, we could see that, for the recommendation system without privacy protection, the user-based collaborative filtering recommendation system has a recall rate of around 0.51, and differential privacy protection algorithms DP-MRE and S-DPDP also cause a certain degree of recommendation quality loss. However, with the increase of the privacy parameter ε, the recall rate gradually increases to that of *Baseline* algorithm. In the

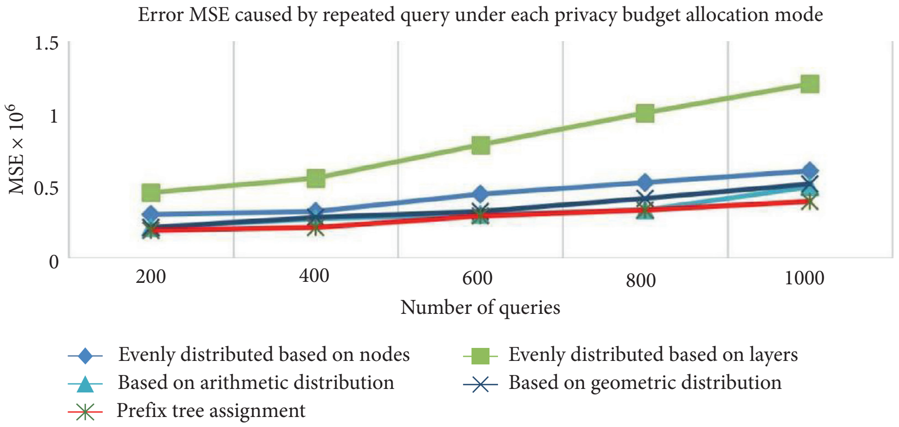


Figure 5: Error MSE caused by repeated query under each privacy budget allocation scheme.

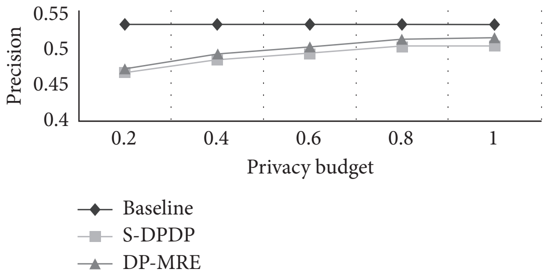


Figure 6: Impact of differential privacy protection on the precision of recommendation systems.

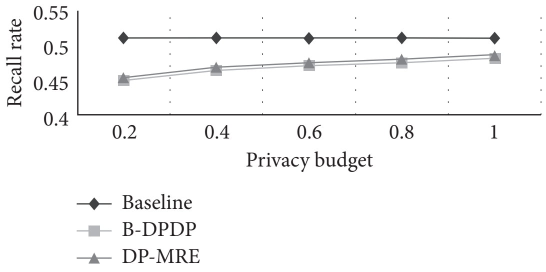


Figure 7: Impact of differential privacy protection on the recall rate of recommendation systems.

recommendation results, higher precision and recall rate indicate a higher recommendation system. According to the experimental results, the recall rate of DP-MRE is very similar to that of S-DPDP; especially when the dataset is relatively large, the recall rate of these two recommendation algorithms is basically the same, whereas the recall rate of DP-MRE is slightly higher than that of S-DPDP algorithm.

6. Conclusion

In this paper, we mainly introduced how to apply the differential privacy protection scheme in a movie recommendation system to protect users' privacy during the recommendation process, while in the meantime, ensuring the recommendation performance will not suffer too much loss. In conclusion, the scheme proposed in this paper firstly adds noise to local sensitive data in a dynamic manner to ensure users' privacy, then sends the data with added noise to the server for similarity calculation, and finally gives out movie recommendation via user-based collaborative filtering algorithm. The experimental results have shown that this scheme could achieve a considerable balance in the trade-off between preserving users' privacy and ensuring the performance of recommendation system. A meaningful attempt of combining differential privacy and recommendation algorithm has been made in our research. However, there are still a lot of open issues that are worth to be investigated in both fields of differential privacy and recommendation algorithms [22]. What is more, the application of differential privacy in recommendation algorithms other than user-based collaborative filtering algorithm will be further studied in our future research.

Data Availability

All data are owned by third parties. The dataset used in this paper is the MovieLens 1M from https://grouplens.org/datasets/movielens/1 m/.

Disclosure

An earlier version of this paper was presented at the International Symposium on Security and Privacy in Social Networks and Big Data (Social Sec 2020).

Conflicts of Interest

The authors declare that they do not have any commercial or associative interest that represents conflicts of interest in connection with the work submitted.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61672300) and the Industrial Internet Innovation and Development Project of the Ministry of Industry and Information Technology of PRC (Grant no. TC190H3WM).

References

[1] Z. Yang and K. Järvinen, "Towards modeling privacy in WiFi fingerprinting indoor localization and its application," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 10, no. 1, pp. 4–22, 2019.

[2] J. Jung, H.-J. Kim, S.-J. Cho, S. Han, and K. Suh, "Efficient android malware detection using API rank and machine learning," *Journal of Internet Services and Information Security*, vol. 9, no. 1, pp. 48–59, 2019.

[3] K. Chaudhuri, A. Sarwate, and K. Sinha, "Near-optimal differentially private principal comp-onents," in *Proceedings of the Conference on Neural Information Processing Systems*, pp. 989–997, Toronto, Canada, December 2012.

[4] K. Chaudhuri and S. A. Vinterbo, "A stability-based validation procedure for differentilly private machine learning," in *Proceedings of the Conference on Neural Information Processing Systems*, pp. 2652–2660, Lake Tahoe, ND, USA, December 2013.

[5] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: asurvey of recent developments," *ACM Computing Surveys*, vol. 42, no. 4, p. 14, 2010.

[6] A. Guha Thakurta and A. Smith, "Optimal algorithms for private online learning in full-information and bandit settings," in *Proceedings of the Conference on Neural Information Processing Systems*, pp. 2733–2741, Lake Tahoe, ND, USA, December 2013.

[7] M. Hardt, K. Ligett, and F. Mcsherry, "A simple and practical algorithm for differentially private data release," in *Proceedings of the Conference on Neural Information Processing Systems*, pp. 2339–2347, Toronto, Canada, December 2012.

[8] F. McSherry and I. Mironov, "Differentially private recommender systems: building privacy into the net," in *Proceedings of the Knowledge Discovery and Data Mining*, pp. 627–636, Bangkok, Thailand, July 2009.

[9] Q. Ye, X. Meng, M. Zhu et al., "A review of localized differential privacy," *Journal of Software*, vol. 29, no. 7, pp. 159–183, 2018.

[10] C. Dwork, K. Kenthapadi, F. Mcsherry et al., "Our data, ourselves: privacy via distributed noise generation, advances in cryptology-EUROCRYPT 2006," in *Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, St. Petersburg, Russia, June 2006.

[11] C. Dwork, "Calibrating noise to sensitivity in private data analysis," *Lecture Notes in Computer Science*, vol. 3876, no. 8, pp. 265–284, 2012.

[12] C. Dwork, F. Mcsherry, and K. Talwar, "The price of privacy and the limits of lp decoding, acm symposium on theory of computing," *Association for Computing Machinery*, vol. 23, 2007.

[13] S. Vágvölgyi, "Descendants of a recognizable tree language for prefix constrained linear monadic term rewriting with position cutting strategy," *Theoretical Computer Science*, vol. 732, pp. 60–72, 2018.

[14] M. Hay, V. Rastogi, G. Miklau et al., "Boosting the accuracy of differentially private histograms through consistency," *Proceedings of the VLDB Endowment*, vol. 29, pp. 1021–1032, 2009.

[15] R. Chen, B. C. M. Fung, and B. C. Desai, "Differentially private transit data publication:a case study on the Montreal transportation system," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 213–221, Beijing China, August 2012.

[16] T. Shang, Z. Zheng, W. Shu et al., "Algorithm of big data decision tree based on isometric privacy budget allocation," *Engineering Science and Technology*, vol. 51, no. 2, pp. 134–140, 2019.

[17] X. Wang, H. Han, Z. Zhang, Q. Yu, and X. Zheng, "Budget allocation method for tree index data differential privacy," *Computer Application*, vol. 38, no. 7, pp. 1960–1966, 2008.

[18] D. Hu and Z. Liao, "Differential privacy location privacy protection method for m- fork average tree," *Journal of Small and Micro Computer Systems*, vol. 40, no. 3, pp. 76–82, 2019.

[19] J. Paul Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 35, no. 3, 1997.

[20] J. J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Kno-wledge-Based Systems*, vol. 46, 2013.

[21] J. Chen, X. Wang, S. Zhao, F. Qian, and Y. Zhang, "Deep attention user- based collaborative filtering for recommendation," *Neurocomputing*, vol. 383, 2020.

[22] L. J. Helsloot, G. Tillem, and Z. Erkin, "BAdASS: preserving privacy in behavioural advertising with applied secret sharing," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 10, no. 1, pp. 23–41, 2019.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8841233, 11 pages
https://doi.org/10.1155/2020/8841233

Research Article

An Android Malware Detection Model Based on DT-SVM

Min Yang,[1] **Xingshu Chen**,[2] **Yonggang Luo**,[2] **and Hang Zhang**[3]

[1] *College of Cybersecurity, Sichuan University, Chengdu, China*
[2] *College of Cybersecurity and the Cybersecurity Research Institute, Sichuan University, Chengdu, China*
[3] *Technology and Engineering Group, Tencent, Shenzhen, China*

Correspondence should be addressed to Xingshu Chen; chenxsh@scu.edu.cn

Received 31 August 2020; Revised 8 October 2020; Accepted 26 November 2020; Published 15 December 2020

Academic Editor: Zhe-Li Liu

In order to improve the accuracy and efficiency of Android malware detection, an Android malware detection model based on decision tree (DT) with support vector machine (SVM) algorithm (DT-SVM) is proposed. Firstly, the original opcode, Dalvik opcode, is extracted by reversing Android software, and the eigenvector of the sample is generated by using the n-gram model. Then, a decision tree is generated via training the sample and updating decision nodes as SVM nodes from the bottom up according to the evaluation result of the test set in the decision path. The model effectively combines DT with SVM. Under the premise of maintaining a high-accuracy decision path, SVM is used to effectively reduce the overfitting problem in DT and thus improve the generalization ability, and maintain the superiority of SVM for the small sample training set. Finally, to test our approach, several simulation experiments are carried out, and the results demonstrate that the improved algorithm has better accuracy and higher speed as compared with other malware detection approaches.

1. Introduction

In recent years, mobile Internet has played a leading role in the evolvement of the Internet, and smartphones have become almost an indispensable tool in people's daily life. Smartphone penetration among adults in developed countries will reach 90 percent by the end of 2023, compared with 85 percent in 2018, and global smartphone sales will reach 1.85 billion units, 19% increase over 2018 [1]. According to [2], worldwide sales of smartphones to end users are on track to reach 1.57 billion units in 2020, an increase of 3% year over year. Although the market sales of smartphone went through a slight declination in 2019, Gartner forecasts that sales of 5G mobile phones will total 221 million units in 2020, and more than double in 2021, to 489 million units; there is no doubt that the gradual maturity of 5G technology will also push the demand of smartphones rise considerably.

Currently, the common operating systems of smartphone terminals include iOS, Android, and Windows Phone, among which Android, in particular, became the dominating operating system with the highest market share on a global scale because of its open-source nature, which gives users and developers the flexibility to customize basic functionality [3]. According to survey data released by Gartner, the share of the Android system in 2017 was as high as 85.9% [4]. However, the increasing popularity of Android is also accompanied by the proliferation of malware. In 2018, 360 Internet Security Center intercepted about 4.342 million new malicious samples on the mobile terminal, with an average of about 12,000 new ones added every day. The new malware types are mainly tariff consuming, accounting for about 63.2%, followed by privacy theft 33.7%, malicious deduction 1.6%, rogue behavior 1.2%, and remote control 0.3% [5]. The terminal application endangers the users' interests by allowing unauthorized access to privacy-sensitive information, rooting devices, monitoring their daily behaviors, etc. [6]. The amount of malware continues to grow at a faster rate each year and poses a serious security threat, antivirus vendors detect thousands of new malware samples daily, and there is still no end in sight [7]. In particular, with the gradual maturity of 5G technology, which marks the arrival of the era of intelligent networking and industrial Internet, the Internet of everything will lead to more lethal and wider harm caused by malware, and hence, malware detection has been and will be a critical topic in computer security.

In this study, we develop a DT with the SVM algorithm (DT-SVM) for improving the detection efficiency and accuracy of malware on the Android platform. The major contributions of this work can be summarized as follows:

(i) We develop an advanced machine learning algorithm, which firstly extracts the opcode of samples; then, n-gram is utilized to vectorize and train the sample to generate the decision tree; and, finally, the nodes with high error are updated from the bottom up as SVM nodes. The algorithm combines the advantages of DT and SVM; on the premise that high accuracy is maintained, the SVM node is employed to reduce the overfitting problem caused by DT. Therefore, the algorithm takes full advantage of the SVM in a small sample set and has a better classification effect than merely using DT or SVM separately.

(ii) We design an Android malware detection framework based on DT-SVM algorithm. The framework is trained based on the improved learning algorithm with the malicious and benign applications utilized, and feature vectors of these applications are generated by Android reverse engineering, feature engineering, and n-gram, which are used as the input of the proposed algorithm for malicious detection. In this way, users can employ our proposed framework to distinguish whether the application is malicious or benign before installation; thus, the Android platform security issues can be greatly improved.

(iii) We verify the effectiveness of our advanced algorithm based on real-word benign applications and malware, perform malicious detection on the same dataset of the proposed algorithm with the shallow learning algorithms DT and SVM and the deep learning algorithms CNN and LSTM, and use four evaluation metrics (Precsion, ACC, Recall, $F1$) as well as time consumption to measure the performance of the algorithm. The results demonstrate that our proposed algorithm performs better than SVM, DT, and LSTM almost in all metrics and performs better than CNN in some metrics. All the four metrics, that is, Precision, Recall rate, ACC, and F1, increase by nearly 0.01% compared with SVM, while the time consumption reduces to one-tenth, as well as increasing by nearly 0.03% separately compared to DT with time consumption not changed much. Compared with CNN, although ACC and F1 are lower, Precision and Recall are higher; furthermore, our algorithm takes less time, and the implementation process is much simpler. In terms of LSTM, our method performs better than it in all metrics.

The remainder of this paper is organized as follows. Section 2 states some current work of Android malware detection. Section 3 depicts the related methodology. Section 4 describes the proposed classification algorithm. Section 5 illustrates the Android malware detection framework and explains the specific process of applying the proposed algorithm to the detection of malicious applications. Section 6 verifies the effectiveness of the advanced algorithm based on Android applications. Section 7 concludes the paper and points out the main limitations and future directions.

2. Related Work

There have been a lot of achievements in terms of detecting malware on the Android platform, which can be divided into two analysis approaches, that is, static analysis and dynamic analysis [8]. Static analysis is the process of analyzing the code or binary without executing it. Dynamic analysis is the process of studying traces of the malware (API, system calls, permission, etc.) through running the sample in a controlled and isolated environment [9]. Traditionally, malware detectors have been built on handmade detection patterns that are not usually applicable to new instances of malware; however, the increasing number and diversity of these applications make traditional defenses largely ineffective; Android smartphones often fail to protect themselves from new malware [10]. Owing to the emergence of machine learning technology, which can potentially detect never-before-seen attacks or variants of known malware with its strong generalization and prediction ability, machine learning-based methods are increasingly applied to Android malware detection by researchers, and the improvement of classic algorithms has always been the tireless work of scholars. The shallow learning model and the deep learning model are the two main types of machine learning techniques [11]. The shallow learning model usually includes SVM, DT, and k-means as well as k-nearest neighbor (KNN) algorithms, etc. [12]. Reference [13] improved the accuracy of the classifier by using machine learning to extract features from the system call of Android malware. Due to the high feature dimension in Dalvik opcode-based detection, [14, 15] utilized two strategies of probability statistics and feature extraction to effectively reduce the dimensionality of extracted features, and the linear SVM was employed for classification, and therefore, the inspection efficiency was improved. Based on the characteristics of permission information and Intent information in AndroidManifest.xml file, a random forest improvement algorithm based on weighted voting was proposed in [16], and the inability to distinguish strong and weak classifiers was solved. Nancy and Sharma [17] compared the network traffic of malware with that of benign applications to find out the characteristics that distinguish the two types of traffic and built a DT classifier to detect normal and malicious applications from the test dataset. The results showed that the network traffic analysis method was efficient in detecting Android malware, with an accuracy rate of more than 90%. Nevertheless, most of the work mentioned above has not achieved decent performance. Recently, Android malware researchers have also been exploring deep learning classifiers for malware analysis to increase detection accuracy [18]. Cui et al. [19] took the advantage of the performance of deep learning in image recognition; the malicious detection code was converted into a grayscale image as the input of CNN under the condition of the fixed image size, which was not realistic in a real scenario. Therefore, this method suffered from fluctuating in performance when processing different sizes of images. To improve the accuracy of malware detection and reduce the training time, Wang et al. [20] proposed a hybrid model based on deep autoencoder (DAE) and convolutional neural network (CNN); the experiments demonstrated a significant improvement compared with traditional machine learning methods in Android malware detection. Wang et al. [21] ranked the permissions w.r.t. their risk to the Android system and evaluated the feasibility of using permission requests for malapp detection with different subsets of risky permissions and classification algorithms; the detection rate can achieve 94.62%. Furthermore, the author considered the issue of user privacy information leakage in literature [22] and implemented a framework called ‘Alde’ to detect the users in-app actions collected by analytics libraries; experimental results show that some apps indeed leak users personal information through analytics libraries. Lei et al. [23] adopted more advanced features than the API event behavior model as a data source, using different behavior patterns of events and the semantic relationships between events to detect malicious software. This method can effectively solve the problem of confusion deformation. However, the results of the experiment performed quite well only in the malware dataset provided in 2013. As the complexity of the malware increased, the detection ability declined.

In summary, it can be concluded that there are two ways to improve the detection accuracy and efficiency of Android malware; the first is through optimization of feature selection and detection model, and the second is to optimize classification algorithms. We mainly focus on the latter and improve the classic classification algorithm in this study. SVM is simple and can achieve high classification accuracy. However, it is merely suitable for small samples; if the sample set is large, it will consume a lot of time and have a high false positive rate. DT is easy to overfit, leading to weak generalization ability of prediction results. To overcome these limitations, our work proposes an advanced learning algorithm based on static features and combines the advantages of SVM and DT algorithm, and the experimental results are quite good. In the next section, we explain the methodology.

3. Methodology

3.1. N-Gram. N-gram model is derived from Natural Language Processing (NLP), commonly used in large-scale continuous speech recognition, which believes that the appearance of the N_{th} word must be related to the first N – 1 words, but not to other words. Hence, the probability of the entire sentence should be equal to the probability product of the occurrence of each word. N-gram can also be used in malware detection. As early as 2008, Moskovitch et al. [24] proposed the opcode n-gram scheme and achieved good detection results.

3.2. Support Vector Machine (SVM). SVM [25] is a two-class model whose basic model is a linear classifier that defines the interval maximization in the eigenspace. Meanwhile, it can also solve the nonlinear problem employing kernel trick [26]. The learning strategy of SVM is to maximize the interval, which can be formalized as a problem of solving convex quadratic programming, also called the maximum edge algorithm, whose advantage lies in strong generalization ability, which can solve the issues of nonlinear, small samples, high dimension, etc. Taking the linear separable SVM as an example, the principle of SVM is to search for a separable hyperplane in given eigenspace and then divide the sample space into two categories, one is a positive class and the other is a negative class, corresponding to two different categories of samples. The hyperplane H in the support vector machine can be represented by the equation of $w \cdot x + b = 0$, where w is the normal vector and b is the intercept, as shown in Figure 1.

When the training samples are linearly separable, there are many straight lines that can correctly classify the two types of samples, and SVM is to find the line that can correctly divide them with the largest interval. SVM also supports nonlinear problem classification, whose main character is the utilization of kernel trick, the basic idea behind which is to match the input space to an eigenspace, so that its hypersurface model in the input space corresponds to the hyperplane model in the eigenspace through a nonlinear transformation. The radial basis function (RBF) is one of the commonly used kernel functions.

Definition 1. Gaussian kernel function

$$K(x, z) = \exp\left(-\frac{\|x - z\|_2^2}{2\sigma^2}\right). \tag{1}$$

Here, $\|x - z\|_2^2$ is the square Euclidean distance of two eigenvectors, and σ is a free parameter.

3.3. Decision Tree. Decision tree [27] is a basic classification and regression method, which classifies samples into a tree structure, represents the process of classifying samples based on features in classification problems, and can also be considered as a collection of if-then rules. DT is widely used because of its intuitive feature description, high classification accuracy, and simple implementation [28]. The learning process of DT is to find a mapping relationship between the object attribute and the object value, enabling it to generalize a set of classification rules represented by tree structure from random samples. The decision path of DT has important properties: mutual exclusion and completeness; that is, each instance is covered by the one and the only one path. The learning algorithm of DT includes feature selection, decision tree generation, and pruning process. The widely used generation algorithms of DT are ID3, C4.5, and CART. The Gini index is used for optimal feature selection in CART algorithm.

Definition 2. Gini index

In the classification problem, suppose that there are K classes and the probability that the sample belongs to the k_{th} class is p_k; then, the Gini index of the probability distribution is defined as

$$\text{Gini}(p) = \sum_{k=1}^{K} p_k (1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2. \tag{2}$$

In the dichotomy problem, the Gini index of the sample set D is expressed as

$$\text{Gini}(D) = 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D|} \right)^2. \tag{3}$$

Here, $|C_k|$ represents the number of samples in category k, and $|D|$ represents the total number of samples. The Gini index indicates the uncertainty of the sample set. The larger the Gini index, the greater the uncertainty of the sample set.

4. Decision Tree with SVM Algorithm (DT-SVM)

To overcome the problem of overfitting and weak generalization ability in DT algorithm, DT-SVM is proposed. SVM is embedded into DT for node optimization, which not only ensures the high accuracy of the decision path and improves the generalization ability of DT, but also takes advantage of SVM in small sample training. DT-SVM aims to create a decision model as shown in Figure 2. The process of the algorithm is to generate a decision tree based on the sample set and then update the decision node from the bottom up.

DT is a supervised learning algorithm. The sample set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ is divided into the training set and the test set, denoted by TrainSet and TestSet.

Definition 3. Assume that the decision tree is as shown in Figure 3, where the leaf nodes are instance sets, represented by $S = \{d_1, d_2, \ldots, d_n\}$, where n is the number of leaf nodes. The nonleaf node is a feature set and is denoted by $C = \{c_1, c_2, \ldots, c_n\}$. Each leaf node corresponds to a decision path, the decision path corresponding to the leaf node j is defined as $\text{dp}_j = \{c_1, c_k, \ldots, d_j\}$, and $h = \text{len}(\text{dp}_j)$ indicates the depth of the path. The details of our suggested DT-SVM algorithm for Android malware detection are presented in Steps 1 to 8.

According to the algorithm process, assume that the initial decision tree is shown in Figure 4 and the DT-SVM tree generated by the algorithm is shown in Figure 5.

The algorithm has a good performance in the example illustrated by Figure 6, in which the sample set cannot be effectively segmented, adopting DT and SVM algorithm separately, but the DT-SVM algorithm can preserve the high precision decision path and optimize nodes with low precision as SVM nodes.

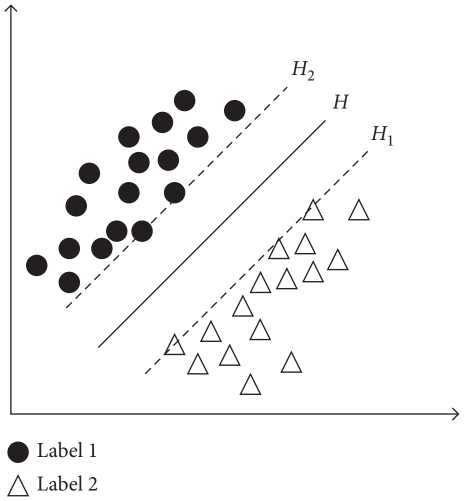


FIGURE 1: The hyperplane of SVM.

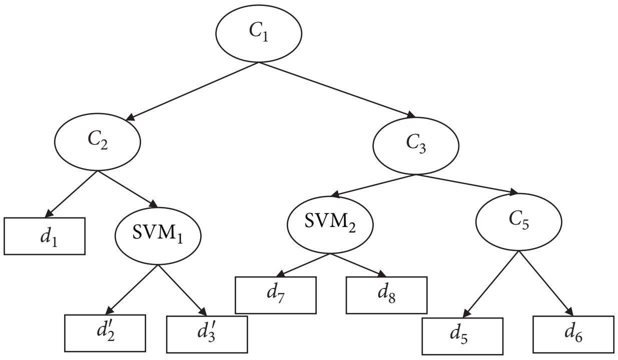


FIGURE 2: Decision tree with SVM nodes.

5. DT-SVM-Based Malware Detection Framework

5.1. Model Overview. The DT-SVM-based malware detection framework is shown in Figure 7. The framework consists of four modules, that is, instruction extraction, feature engineering, classifier training, and result evaluation.

5.2. Sample Instruction Extraction. Firstly, those samples are labeled as two categories, positive and negative. Then, opcode extraction is performed for each apk. After apk decompression, the core classes.dex file of the app will be obtained. The classes.dex file is the executable file of the Android system, which contains all operation instructions and data required by the runtime. The dex file can be parsed by 010 Editor, and its structure is shown in Figure 8. The Methods structure contains all the methods of the app, represented by the DexMethod structure.

```
struct DexMethod{
  /* Index pointing to the list of DexMethodId*/
  u4 methodIdx;
  u4 accessFlags;
  /* offset to the DexCode structure  */}
```

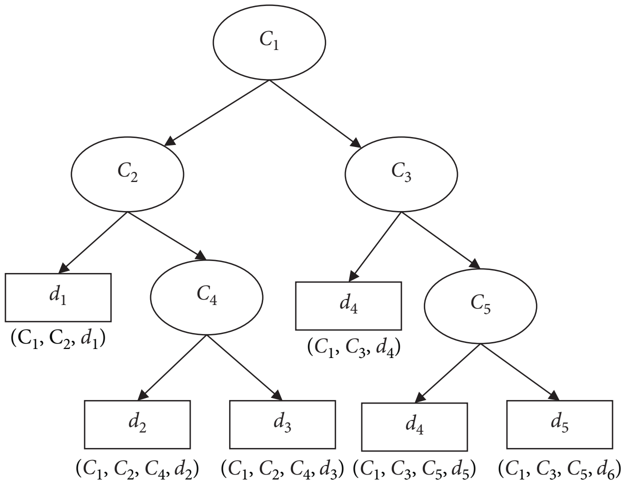


Figure 3: Traditional decision tree labeled with decision path.

Step 1. According to the training set *TrainSet*, the Gini index is used for feature selection and prepruning, and the decision tree T is constructed.

Step 2. Use the test set *TestSet* to evaluate the decision tree and calculate the *Precision* of each decision path p_i, then constitute the decision object do = $(\mathrm{dp}_i, p_i, h_i)$, and set the decision path accuracy threshold *Th*.

Step 3. Initialize the queue Q = { }, sort the decision objects generated in step 2 in descending order according to the path depth h of the decision path dp, and sequentially add them to the queue Q.

Step 4. Determine if the queue is empty. If it is, the algorithm ends. Otherwise, go to step 5.

Step 5. Fetch the element $q = (\mathrm{dp}, p, h)$ from the queue, and compare the decision path Precision rate p with the preset threshold *Th*. If it is less than the threshold, go to step 6; otherwise, retain the decision path and go to step 4.

Step 6. Determine whether the sibling node of q is a leaf node. If it is, go to step 7; otherwise, go to step 8.

Step 7. Determine whether the Precision of the path of q's sibling nodes is lower than the threshold *Th*. If it is, all the samples passing through the two decision paths (both path of q and q's siblings) are taken as a training set, which is trained with the SVM model and then merged and updated as SVM nodes; thereafter, the process proceeds to step 4.

Step 8. Take out all the training sets of the path of p, train them with the SVM model, and update them to SVM nodes. Then, go to step 4 and continue to traverse so as to update nodes.

Algorithm 1: The detailed procedure of DT-SVM.

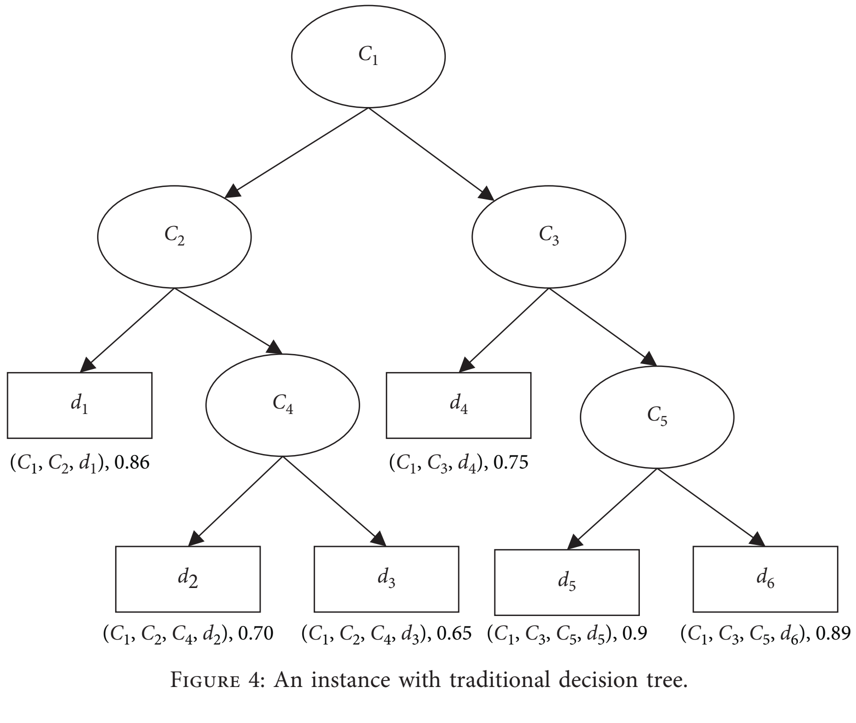

C_1
C_2
C_3
d_1
(C_1, C_2, d_1), 0.86
C_4
d_4
(C_1, C_3, d_4), 0.75
C_5
d_2
d_3
d_5
d_6
(C_1, C_2, C_4, d_2), 0.70
(C_1, C_2, C_4, d_3), 0.65
(C_1, C_3, C_5, d_5), 0.9
(C_1, C_3, C_5, d_6), 0.89

Figure 4: An instance with traditional decision tree.

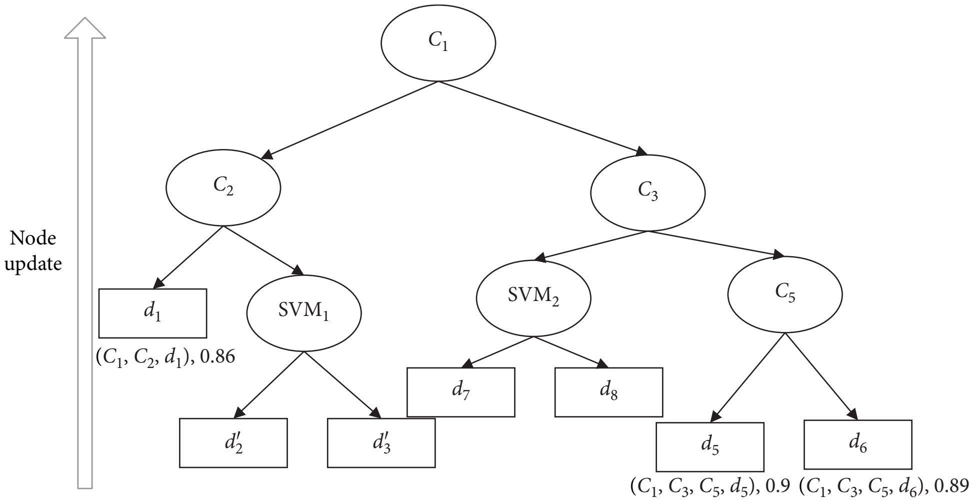


Figure 5: Updating decision tree with SVM nodes from the bottom up.

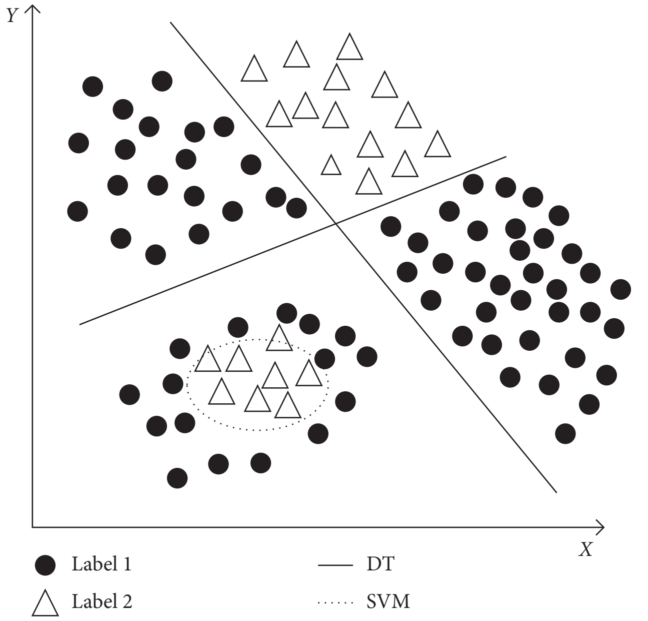


Figure 6: Classification of samples with DT-SVM algorithm.

```
  u4 codeOff;
  }
struct DexCode{
  /* the number of used registers */
  u2 registersSize;
  /* the number of parameters */
  u2 insSize;
  /* the number of used registers when calling other
 methods */
  u2 outSize;
 /* the number of try and catch */
  u2 triesSize;
  /* offset to debug information */
  u4 debugInfoOff;
  /* the number of Instruction Set */
  u4 insnsSize;
  /* Instruction Set */
  u2 insns [1];
  }
```

In this structure, the last field insns[1] contains all the instruction sets of the method, namely, the corresponding Dalvik opcode. By going through all the methods, all opcode instructions can be fetched according to the Dalvik opcode instruction list in Table 1.

5.3. Feature Engineering. Since there are more than 200 Dalvik instructions, if all of them are directly input into the n-gram model, the feature dimension will be too high. In this paper, first of all, the Dalvik instruction sets are simply

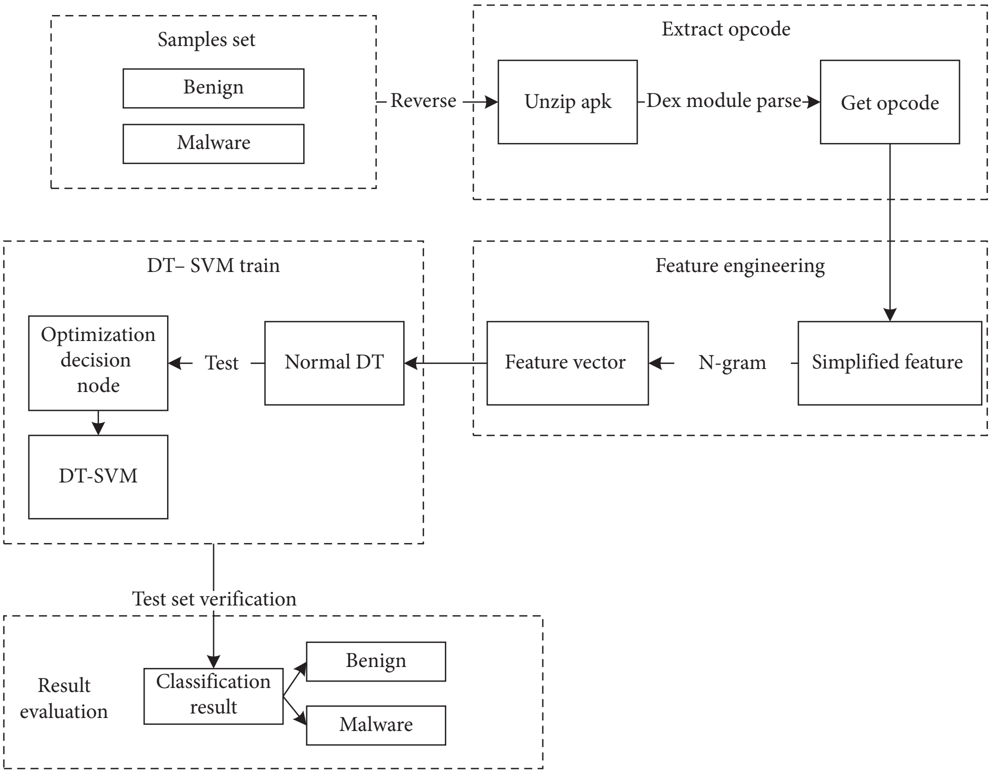


Figure 7: Android malware detection model based on DT-SVM algorithm.

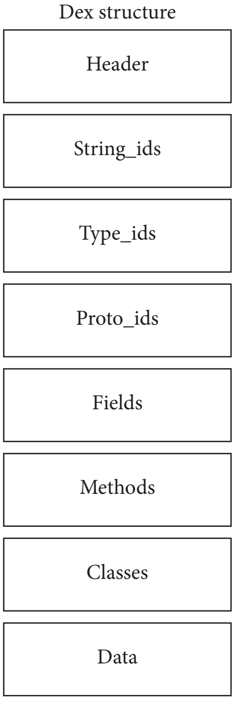


Figure 8: Android dex file structure.

classified; then, irrelevant instructions are removed; and, finally, only eight types are left. The opcode and its corresponding identifier are shown in Table 2.

Table 1: Dalvik opcode.

Opcode (hex)	Opcode name	Length
00	nop	2
01	move vx, vy	2
02	move/from16 vx, vy	4
03	move/16	6
04	move-wide	2
05	move-wide/from16 vx, vy	4
06	move-wide/16	6
07	move-object vx, vy	2
08	move-object/from16 vx, vy	4
09	move-object/16	6
0A	move-result vx	2
0B	move-result-wide vx	2
0C	move-result-object vx	2
0E	return-void	2
0F	return vx	2
10	return-wide vx	2
...	...	...

After simplifying the Dalvik instruction sets, all of them can be input to the n-gram model to generate sample eigenspace. The extracted opcode for each sample in Section 5.2 is mapped to the identifier, and the n-gram vector is constructed. Assuming that the Dalvik instruction is $\{G, P, V, I, J, R, M, C\}$, when $N = 3$, the 3-gram vector is $[\{GPV\}, \{PVI\}, \{VIJ\}, \{JRM\}, \{RMC\}]$.

After the n-gram model of samples is obtained, the n-gram types are counted. If a feature appears in the sample,

Table 2: Feature simplification mapping.

Identifier	Description	Opcode
G	Fetching data	aget—iget—sget—aget-wide—aget-object—aget-boolean—aget-byte—aget-char
P	Storing data	aput—iput—sput—aput-wide—aput-object—aput-boolean—aput-byte—aput-char
V	Method call	invoke-virtual—invoke-super—invoke-direct—invoke-static
I	Judgement	if-eq—if-ne—if-lt—if-ge—if-gt—if-le—if-eqz—if-nez—if-ltz—if-ltz—if-gez—if-gtz—if-lez
J	Goto	goto—goto/16—goto/32
R	Return	return—return-void—return-wide—return-object
M	Move	move—move-wide—move-object—move-result—move-exception
C	Compare	cmpl-float— cmpg-float— cmpl-double— cmpg-double—cmp-long

the value of the feature is set to 1; otherwise, it is set to 0; the feature vector of the sample is finally obtained.

5.4. Evaluation Metrics. Four metrics are employed to verify the performance of our proposed algorithm, namely, Precision, Recall, classification accuracy ACC, and F1 value, which are broadly used in machine learning. The Precision can be denoted as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4}$$

where TP (true positive) indicates the number of Android malware samples which are correctly detected and FP (false positive) indicates the number of benign applications that are wrongly detected as Android malware [29]. In this study, the Precision refers to the ratio of the identified malicious samples to the real malicious samples. The Recall can be formulated as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{5}$$

where FN (false negative) indicates the number of Android malware samples that are not detected (predicted as benign applications) [29]. In this study, Recall reflects the proportion of malicious samples identified in the real malicious sample. The ACC can be formulated as

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{6}$$

where TN (true negative) represents the number of benign applications that are correctly classified and ACC is an overall evaluation of the classifier, representing the proportion of the total number of the applications that are correctly classified whether as benign or malicious. The higher the ACC is, the better the performance will be. F1 is the harmonic mean between the Precision and Recall; it can be denoted as

$$F1 = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{7}$$

6. Experimental Simulation

6.1. Datasets. In the experimental simulation environment, the malicious sample set was obtained from the malicious sample database in the Drebin project of the University of Gottingen, Germany [30], in which the malware samples are 5560 in total, and the time range was from August 2010 to October 2012. An overview of the top 20 malware families in the dataset is provided in Table 3, including several families that are currently actively distributed in application markets. There are 4414 benign samples, and the benign samples were randomly selected from the applications, which were downloaded from the Google Play app store in order of ranking through the crawler module. The tools used in the experiment include unzip, dexParser, scikit-learn, etc. Scikit-learn is an excellent Python programming machine learning library, which has a variety of classification, regression, and clustering algorithms, including support vector machine, random forest, and gradient enhancement.

Table 3: Top malware families in our dataset.

ID	Family	Number
A	FakeInstaller	925
B	DroidKungFu	667
C	Plankton	625
D	OpFake	613
E	GingerMaster	339
F	BaseBridge	330
G	Iconosys	152
H	$K_{\min}$	147
I	FakeDoc	132
J	Geinimi	92
K	Adrd	91
L	DroidDream	81
M	Linux/Lotoor	70
N	GoldDream	69
O	MobileTx	69
P	FakeRun	61
Q	SendPay	59
R	Gappusin	58
S	Imlog	43
T	SMSreg	41

6.2. Experimental Procedure. The sample set was divided into a training set, a pseudo test set, and a test set in the ratio of 6 : 2:2. The training set feature vector was input into the DT-SVM model for training. The pseudo test set was used to update the decision node and obtain the DT-

SVM tree. Finally, the test set was employed to evaluate the performance of the classifier.

The experiment used 1638 malicious samples and 1324 normal samples. 60% of the training sets and 20% of the pseudo test sets were used to generate the DT-SVM model, and then the remaining 20% were used to evaluate the classifier performance. 3-gram was used for feature selection. Since different sampling will affect the classification results, the experiment will perform 10-fold cross-validation.

In order to ensure that the decision leaf node has sufficient sample capacity for SVM training, the decision tree needs to be prepruned. In the experiment, the minimum sample number of the leaf node min_samples_leaf is 40, the maximum depth of the decision tree max_depth is 5, and the Precision threshold is set to 0.9. The decision tree path below the threshold is shown in Table 4, where the field of 'Path matrix' is the binary representation of decision path. The encoding process is to sort all nodes of a decision tree from left to right and from top to bottom, and then map them to a multidimensional vector. The position of this multidimensional vector represents the sort of decision tree node, and the value represents whether the decision path contains this node. If it is 1, the node is included; otherwise, it is not included.

For these decision paths with higher error, the samples under each path are taken out for SVM training to generate SVM nodes. The Gaussian kernel function is used to process the feature space during training. At this time, there are two essential parameters that need to be adjusted, namely, the C (Penalty factor) and gamma (RBF kernel width). In general, a larger C leads to higher tolerance, but fewer errors, so as to eliminate overfitting. Otherwise, it is easy to result in underfitting. Gamma is a parameter of the Gaussian kernel function. The larger the gamma is, the less the support vector is, and the simpler the model is.

After training, the parameters of each SVM node are shown in Table 5.

6.3. Experimental Results

6.3.1. Scenario I: The Impact of Different N-Gram Types on the Classifier. DT and SVM classifiers were trained separately applying different n-gram models, and the predictive Precision results are shown in Table 6.

The results show that DT and SVM can get good evaluation results on the basis of 3-gram and 4-gram, demonstrating the feasibility of the modeling method. When $n > 3$, the Precision of DT only increases by 0.7%; SVM increases by 2%, but it consumes a lot of time. SVM takes 1002.23 seconds under 4-gram and 113.65 seconds under 3-gram, so $n = 3$ gives the best performance for sample vectorization.

6.3.2. Scenario II: Results Comparison with Shallow Learning Algorithm. The sample was vectorized based on 3-gram, and Table 7 demonstrates a comparison of the proposed algorithm with SVM and DT for Android malware detection.

The results show that the Precision, ACC, Recall, and $F1$ of the DT-SVM algorithm are apparently higher than traditional DT and slightly higher than SVM. In terms of efficiency, SVM takes the longest time, while DT-SVM is trained by DT first, and then the small sample is trained by the SVM node. Hence, the time dramatically reduces compared with SVM, albeit a little longer than DT.

6.3.3. Scenario III: Results Comparison with Deep Learning Algorithm. We also compared the CNN [31] and LSTM [32] using the same sample set for training. The results show that ACC and $F1$ of CNN are relatively high, but other metrics are lower than our proposed model, which means that there would be a lot of false positives of CNN. In addition, CNN is time consuming and requires high machine configuration. The performance of the LSTM model for malicious detection of Android is not so good as that of DT-SVM algorithm, and the time consumption is 117s higher than that of our algorithm. The results are detailed in Table 8.

6.3.4. Scenario IV: Comparison of DT-SVM Results with Different Sample Sizes. We randomly select 507 samples from the 2962-sample set for experiment. The effects of different sample sizes on DT-SVM classifier are shown in Table 9 .

The experimental results show that the sample size has a certain influence on the detection effect. The number of samples increases, and Precison, ACC, Recall, and $F1$ increase by 0.03. Hence, we can conclude that the larger the sample size is, the better the overall performance will be.

6.4. Analysis. Decision tree is a prediction model, which represents a mapping relationship between object attributes and object values. Its branches classify objects of this type based on attributes. It is a decision tool using a decision model, which can help determine a strategy most likely to achieve the goal. DT is easy to understand and implement, the advantage of which lies in its ability to make accurate and feasible predictions for large data sources in a short time. The basic principle of DT-SVM is to first extract some high-accuracy decisions through DT model and quickly find the strong correlation between the results and the attributes, and then the kernel technique of SVM is used to solve nonlinear prediction for some weakly correlated samples and at the same time give full play to the advantages of SVM in small sample prediction. Hence, the prediction accuracy of the samples is largely improved through the combination of DT and SVM.

The time complexity of DT is $O(n \log n)$, and SVM is $O(n^3)$. However, DT-SVM first generates a decision tree, selects the optimal path, and then uses SVM for training for the remaining samples, so the time complexity is $O(n \log n) + O(m^3)$, where n is the total number of samples and m is the number of samples that cannot be distinguished with high accuracy after training the sample using the decision tree; $m \ll n$; thus, the value falls in the interval $(O(n \log n), O(n^3))$. In this experiment, decision tree was

Table 4: Decision tree path with Precision.

Path ID	Decision path	Path matrix	Precision
1	$(C_{296}, C_9, C_{313}, C_{304}, C_{308}, d_7)$	1100000110001010000000000	0.737
2	$(C_{296}, C_9, C_{120}, d_1)$	1111000000000000000000000	0.571
3	$(C_{296}, C_9, C_{313}, d_8)$	1100000100000001000000000	0.590
4	$(C_{296}, C_{307}, C_{223}, d_{10})$	1000000000000000101100000	0.685
5	(C_{296}, C_{307}, d_9)	1000000000000000110000000	0.850

Table 5: SVM node parameters.

SVM node ID	C	gamma
1	7	0.03
2	7	0.003
3	1	0.04
4	5	0.04
5	5	0.04

Table 6: The results of scenario I.

N-gram	DT	SVM
2-gram	0.79	0.76
3-gram	0.92	0.95
4-gram	0.94	0.97

Table 7: The results of scenarios II.

Classifier	Precision	ACC	Recall	F1	Time consumption
DT	0.92	0.93	0.93	0.93	8.01s
SVM	0.96	0.96	0.94	0.95	105.79s
DT-SVM	0.96	0.96	0.96	0.96	18.9s

Table 8: The results of scenario III.

Classifier	Precision	ACC	Recall	F1	Time consumption
LSTM	0.893	0.938	0.556	-	117.8s
CNN	0.944	1	0.944	0.971	357.24s
DT-SVM	0.96	0.96	0.96	0.96	18.9s

Table 9: The results of scenario IV.

Samples size	Precision	ACC	Recall	F1
507	0.93	0.93	0.94	0.93
2962	0.96	0.96	0.96	0.96

first used to train samples, and it can be found from Table 4 that the Precision of paths 1, 2, 3, 4, 5 is low, indicating that DT cannot accurately separate positive and negative samples. Taking path 2 $(C_{296}, C_9, C_{120}, d_1)$ as an example, by mapping and restoring, the opcode sequence corresponding to path 2 is JRG, GPP, PCG, where it is observed that JRG is a jump return to obtain data sequence, GPP is a data acquisition and storage sequence, and PCG is a data dump sequence. These sequences are often used for both positive and negative samples; therefore, merely using DT cannot distinguish them effectively (the accuracy is only 57.1%). Based on this, this paper trains these undifferentiated samples using SVM, and the Precision reaches as high as 96%. In summary, the proposed algorithm improves detection accuracy, while the time consumption is relatively low.

7. Conclusion and Future Work

Taking the sample Dalvik opcode as the research object, the n-gram model is utilized to generate the sample eigenvector, and DT-SVM is proposed. Based on the original DT, the proposed algorithm uses SVM to update the decision nodes from the bottom up. The advantages of DT and SVM can be combined through DT-SVM, and the disadvantages of overfitting of DT and low accuracy of SVM for large samples are overcome. Finally, the superiority of the algorithm is demonstrated by simulation experiments, and good results are obtained in Android malicious apps detection.

However, in addition to the above advantages, there are some limitations to our study. This paper only performs static analysis on the sample; if the sample is hardened or confused, the unzip file will no longer be the sample's classes.dex, but the hardened executable file. The Dalvik opcode will be virtualized, and all instructions will be executed by a hardened virtual machine. At this time, opcode will no longer correspond to the Dalvik instruction list, and only the dynamic behavior analysis method can be used for malicious code detection. In addition, the proposed DT-SVM algorithm can still be improved by, for example, using the random forest to further improve the classification ability of DT-SVM and extending DT-SVM algorithm to the multiclassification decision model.

Data Availability

The data in this paper are divided into benign samples and malicious samples. The malicious sample data that support the findings of this study are available but restrictions apply to the availability of these data, which were used under license for the current study, and so they are not publicly available. These data are however available from the corresponding author upon reasonable request and with permission of the Drebin project of the University of Gottingen, Germany. The benign sample data generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] D. Insights, “Tech Trends 2019 beyond the Digital Frontier,” Tech. Rep., Deloitte, London, UK, 2019, https://www.innovation4.cn/library/r37176.

[2] U. K. Egham, “Gartner says worldwide smartphone sales willgrow 3Gartner, UK,” Stamford, CT, USA,2020, Tech. Rep., Gartner, https://www.gartner.com.

[3] R. Mente and A. Bagadi, “Android application security,” *Advances in Computational Sciences and Technology*, vol. 10, pp. 1207–1210, 05 2017.

[4] U. K. Egham, “Gartner says worldwide sales of smartphones recorded first ever decline during the fourth quarter of 2017,” Stamford, CT, USA, Tech. Rep. Gartner, https://www.gartner.com/newsroom/id/3876865.

[5] I. S. Center, “Android malware special report in 2018,” 360 Fenghuo Laboratory, Beijing, China, Tech. Rep., 2019, https://zt.360.cn/1101061855.php?dtidŁ1101061451&didŁ610100815.

[6] S. Y. Yerima and S. Sezer, “Droidfusion: A novel multilevel classifier fusion approach for android malware detection,” *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–14, 2018.

[7] A. Demontis, M. Melis, B. Biggio et al., “Yes, machine learning can be more secure! a case study on android malware detection,” *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 4, pp. 711–724, 2017.

[8] N. Milosevic, A. Dehghantanha, and K.-K. R. Choo, “Machine learning aided android malware classification,” *Computers & Electrical Engineering*, vol. 61, pp. 266–274, 2017.

[9] A. Afianian, S. Niksefat, B. Sadeghiyan, and D. Baptiste, “Malware dynamic analysis evasion techniques: a survey,” 2018, http://arxiv.org/abs/1811.01190.

[10] B. Biggio and F. Roli, “Wild patterns: ten years after the rise of adversarial machine learning,” *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[11] W. Zhong and F. Gu, “A multi-level deep learning system for malware detection,” *Expert Systems With Applications*, vol. 133, pp. 151–162, 2019.

[12] Z.-U. Rehman, S. N. Khan, K. Muhammad et al., “Machine learning-assisted signature and heuristic-based detection of malwares in android devices,” *Computers & Electrical Engineering*, vol. 69, pp. 828–841, 2018.

[13] P. Vinod, A. Zemmari, and M. Conti, “A machine learning based approach to detect malicious android apps using discriminant system calls,” *Future Generation Computer Systems*, vol. 94, pp. 333–350, 2019.

[14] Y. Zhang and C. Yin, “Android malware detection based on svm,” *Journal of Shandong University (Engineering Science)*, vol. 47, no. 1, pp. 42–47, 2017.

[15] X. Liu, J. Weng, Y. Zhang, B. Feng, and J. Weng, “Android malware detection based on apk signature information feedback,” *Journal on Communications*, vol. 38, no. 5, pp. 190–198, 2017.

[16] H. Yang and J. Xu, “Android malware detection based on improved random forest,” *Journal on Communications*, vol. 38, no. 4, pp. 8–16, 2017.

[17] D. Nancy and D. Sharma, “Android malware detection using decision trees and network traffic,” *International Journal of Computer Science and Information Technologies*, vol. 7, no. 4, pp. 1970–1974, 2016.

[18] A. Mohammed K, Y. Suleiman, and S. Sezer, “Dl-droid: deep learning based android malware detection using real devices,” *Computers & Security*, vol. 89, 2019.

[19] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-g. Wang, and J. Chen, “Detection of malicious code variants based on deep learning,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.

[20] W. Wang, M. Zhao, and J. Wang, “Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3035–3043, 2018.

[21] W. Wang, X. Wang, D. Feng, J. Liu, Z. Han, and X. Zhang, “Exploring permission-induced risk in android applications for malicious application detection,” *IEEE Transactions on Information Forensics & Security*, vol. 9, no. 11, pp. 1869–1882, 2017.

[22] X. Liu, J. Liu, S. Zhu, W. Wang, and X. Zhang, “Privacy risk analysis and mitigation of analytics libraries in the android ecosystem,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1184–1199, 2020.

[23] T. Lei, Z. Qin, Z. Wang, Q. Li, and D. Ye, “Evedroid: event-aware android malware detection against model degrading for iot devices,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6668–6680, 2019.

[24] R. Moskovitch, C. Feher, N. Tzachar et al., “Unknown malcode detection using opcode representation,” in *Proceedings of the European Conference on Intelligence and Security Informatics*, pp. 204–215, Esbjerg, Denmark, December 2008.

[25] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[26] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, Pittsburgh, PA, USA, July 1992.

[27] J. R. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[28] “Application of machine learning in cyberspace security research,” *Chinese Journal of Computers*, vol. 41, no. 9, 2018.

[29] A. Mahindru and P. Singh, “Dynamic permissions based android malware detection using machine learning techniques,” in *Proceedings of the 10th Innovations in Software Engineering Conference*, pp. 202–210, Jaipur, India, February 2017.

[30] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, “Drebin: effective and explainable detection of android malware in your pocket,” *In Ndss*, vol. 14, pp. 23–26, 2014.

[31] N. Mclaughlin, A. Doupé, G. J. Ahn, J. M. D. Rincon, and Z. Zhao, “Deep android malware detection,” in *Proceedings of the Acm on Conference on Data & Application Security & Privacy*, Richardson, TX, USA, March 2017.

[32] R. Vinayakumar, K. P. Soman, P. Poornachandran et al., “Detecting android malware using long short-term memory (lstm),” *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1277–1288, 2018.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8872853, 10 pages
https://doi.org/10.1155/2020/8872853

Research Article

Privacy-Preserving Blockchain-Based Nonlinear SVM Classifier Training for Social Networks

Nan Jia,[1] **Shaojing Fu**,[1,2] **and Ming Xu**[1]

[1] *College of Computer, National University of Defense Technology, Changsha, China*
[2] *State Key Laboratory of Cryptology, Beijing, China*

Correspondence should be addressed to Shaojing Fu; shaojing1984@163.com

Received 10 September 2020; Revised 3 November 2020; Accepted 25 November 2020; Published 14 December 2020

Academic Editor: Ilsun You

With the development of social networks, there are more and more social data produced, which usually contain valuable knowledge that can be utilized in many fields, such as commodity recommendation and sentimental analysis. The SVM classifier, as one of the most prevailing machine learning techniques for classification, is a crucial tool for social data analysis. Since training a high-quality SVM classifier usually requires a huge amount of data, it is a better choice for individuals and small enterprises to conduct collaborative training with multiple parties. Nevertheless, it causes privacy risks when sharing sensitive data with untrusted people and enterprises. Existing solutions mainly adopt the computation-intensive cryptographic methods which are not efficient for practical applications. Therefore, it is an urgent and challenging task to realize efficient SVM classifier training while protecting privacy. In this paper, we propose a novel privacy-preserving nonlinear SVM classifier training scheme based on blockchain. We first design a series of secure computation protocols which can achieve secure nonlinear SVM classifier training with minimal computation overheads. Then, leveraging these building blocks, we propose a blockchain-based secure nonlinear SVM classifier training scheme that realizes collaborative training while protecting privacy. We conduct a thorough analysis of the security properties of our scheme. Experiments over a real dataset show that our scheme achieves high accuracy and practical efficiency.

1. Introduction

Nowadays, social networks have been playing a significant role in reflecting and influencing human living styles. It makes people be able to keep in touch with each other and share information anytime and anywhere. The development of social networks has resulted in more and more social-related data being produced, which consist of various raw insights and information. With the evolution of artificial intelligence and machine learning, many individuals and companies try to train learning models by utilizing social data to learn valuable information for personal or commercial purposes. Support vector machine (SVM), as one of the most essential and important machine learning techniques for classification, can be applied to many fields, such as medical diagnosis [1], image recognition [2], and recommendation system [3]. It is also crucial for social data analysis and processing. For instance, many e-commerce companies may collect social information about potential customers and train effective SVM classifiers to conduct commodity recommendation. In addition, individuals may want to train SVM models by utilizing photos from social media to support automated image annotation, which can make it more convenient to find photos from electronic albums. To obtain an SVM classifier with high accuracy, the training process usually requires a huge amount of social data. However, it is difficult for a single party (e.g., an individual or a company) to collect plentiful and diversified data. Therefore, there has been a surge in demand for collaborative training by a group of parties. The merged dataset from multiple sources on social networks has obvious advantages on data volume and variety. Therefore, it is a growing trend for companies and individuals to share information and collaboratively train a high-quality classifier. However, it causes nonnegligible privacy risks when sharing private data with untrusted parties [4]. For instance, the individuals own some private photos in social networks which only themself and their friends can access. They want to train a classifier to support automated image annotation. However, they hesitate to share the photos with other entities that they do not trust because it may cause the leakage of personal privacy. Therefore, it is a crucial problem to perform collaborative training while protecting privacy.

To address this issue, many privacy-preserving classifier training schemes have been proposed. The most common cryptographic method for protecting data is homomorphic encryption [5, 6]. However, the homomorphic encryption technique is usually involved with computationally expensive cryptographic primitives, which result in heavy computation cost. Differential privacy is another method to guarantee the security of data [7, 8]. Nevertheless, the differential privacy technique cannot achieve high accuracy because it protects data privacy by adding immeasurable noises to the parameters of the model. Besides, in the above privacy-preserving training schemes, the data owners completely lose control of their data when outsourcing the data to the untrusted parties. The data ownership has not been well guaranteed, which is also a potential threat of data confidentiality. Once the data are shared to the untrusted servers for training, there are potential risks that the data might be modified or replicated by the unauthorized servers. Recently, several privacy-preserving schemes are proposed to achieve privacy-preserving training by utilizing the blockchain technique due to its decentralized digital ledger property. Shen et al. [9] proposed a privacy-preserving SVM training scheme over blockchain-based encrypted IoT data. However, their scheme just fits linear data but cannot deal with classification tasks for nonlinear datasets, which are more common in practice. Moreover, they adopted the computing-intensive Paillier homomorphic encryption technique, which causes huge computation overheads.

In this paper, we propose a new privacy-preserving nonlinear SVM classifier training scheme based on blockchain. Specifically, we establish a blockchain-based data sharing and computation outsourcing mechanism which allows participants to collaboratively train the model while protecting privacy. We utilize the blockchain technique to establish a distributed data sharing environment. In our scheme, the communication of multiple parties is recorded on the blockchain. It ensures transparent delivery of training tasks and the guarantee of data ownership. The blockchain also supports fair incentives to avoid deceptive behavior. We adopt the additive secret sharing technique based on two-party computation to design the computation protocols, which can achieve secure SVM training with minimal computation overheads.

The main contribution of this paper can be summarized as follows:

(1) To train a high-quality nonlinear SVM classifier while protecting privacy, we propose a privacy-preserving training scheme based on blockchain. We utilize the blockchain technique to design a decentralized scheme for collaborative training while ensuring the invariance and ownership of training data. The incentive mechanism of blockchain can also help to guarantee the fairness of the participants and prevent destructive behaviors of the computing parties meanwhile.

(2) We adopt the additive secret sharing techniques and design a series of arithmetic primitives such as multiplication, comparison, and natural exponential computation to realize efficient collaborative training while protecting the privacy of both the data and the model. The protocols in our scheme contain no computation-intensive cryptographic primitives and greatly reduce the computation overheads.

(3) We thoroughly analyze the security of our scheme and conduct experiments over real-world datasets. The security analysis shows that our scheme can well protect the privacy of the data and the users. We conduct comprehensive experiments, and the results demonstrate that our scheme can achieve high accuracy and obtain nonlinear SVM classifiers with practical training efficiency.

The rest of this paper is organized as follows: Section 2 introduces the formulation of the problem, our design goals, and the preliminaries of our scheme. In Section 3, we present the secure computation protocols based on secret sharing. Section 4 gives the details of our privacy-preserving training scheme. Security analysis and performance evaluation are presented in Section 5 and Section 6, respectively. We conclude the paper in Section 8.

2. Problem Statement

2.1. System Model. In this paper, we focus on designing a scheme for privacy-preserving and efficient nonlinear SVM classifier training. There are three entities in our framework: the blockchain, the data providers, and the servers, as shown in Figure 1. The role of each entity is described as follows:

Blockchain. The blockchain in our scheme serves as a distributed and immutable ledger. Each block of the blockchain stores a group of transactions of the training requests, the delivery of training data, and so on. The blockchain also provides an incentive mechanism to guarantee fairness.

Data Providers. The data providers can be institutions, enterprises, or individuals who own some training data and participate in the collaborative training of the SVM classifier. They take charge of encrypting the original datasets before sending them to the servers and generating random values in the secure computation process.

Servers. The servers are two computing parties which are selected from the parties in the decentralized network. They are incented to conduct the training tasks, like the miners in Bitcoin.

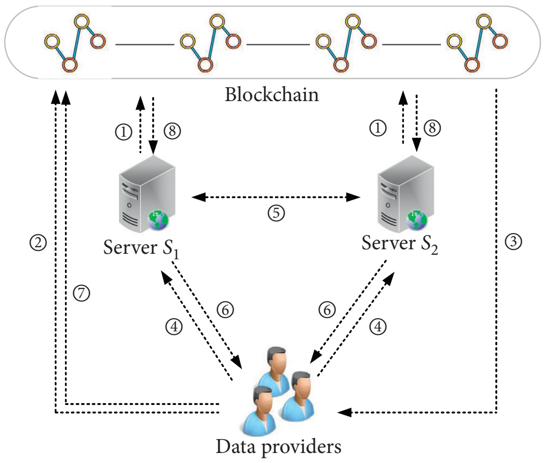


Figure 1: System model of the proposed scheme: ① register and \$deposit; ② request and \$payment; ③ *addr* of servers; ④ encrypted data and random values; ⑤ secure computing protocols; ⑥ encrypted model; ⑦ acknowledgement messages; ⑧ rewards or penalties.

In the blockchain-based decentralized system, the servers should first register on the blockchain and join the network. Meanwhile, each party is required to commit a deposit, which is frozen in the account. The goal of freezing a deposit is to prevent dishonest behavior of the servers. It will be withheld as forfeit if the servers return deceitful results. When the data providers want to conduct collaborative training by utilizing the computing resources in the network, they should generate a transaction that contains the requests and pays. After receiving the transaction, the blockchain selects two servers as computing parties and sends the addresses and the public encryption keys of the servers to the data providers. Then, each provider encrypts the training data with the respective public keys and sends the encrypted data to the servers. On receiving the encrypted data from all the participants, the servers perform the secure computation protocols and conduct the SVM classifier training. The secure computation involves some random values, which can be generated offline by the data providers. After finishing the training process, the servers send the results back to the data providers. Then, each provider reconstructs the model and sends to the blockchain an acknowledgement message, with which the blockchain determines to give the servers rewards or penalties.

2.2. Threat Model. In this paper, we assume that the servers are noncolluding and there are secure channels among the nodes in the decentralized network. We consider the servers to be honest but curious. It means that the servers would execute the designed task honestly, but they are curious to infer sensitive information from the encrypted data and the interactive messages. The privacy threats mainly come from two aspects: (1) the encrypted original datasets. The original datasets may contain lots of sensitive information about the data providers. The adversaries can still infer valuable information about the local data through the training process. (2) The training results. The results of training, i.e., the SVM model, can be used for some commercial benefit. Thus, the results could be embezzled by the computing parties or adversaries.

Specifically, to better evaluate the security of our scheme, we consider two levels of threat as follows:

Level 1. The computing parties can learn nothing about the original datasets. They are assumed to only know the split shares of the datasets and the immediate results of each step in the training protocol.

Level 2. Apart from the information in Level 1, the computing parties can also obtain part of original datasets from certain data providers among the multiple participants of the collaborative training.

2.3. Design Goals. To implement privacy-preserving and efficient nonlinear SVM classifier training, our scheme should meet the following goals:

Security. Our scheme is supposed to protect the information and resist the potential adversities introduced in the threat model. Besides, even if some data providers collude with the servers, the privacy of datasets of other data providers stored on the servers is still preserved.

Correctness. Our training algorithm should be designed correctly to obtain a high-quality SVM classifier model. The privacy-preserving classifier ought to be almost equally effective as those obtained in the plaintext domain, and the accuracy should be high enough for practical use.

Efficiency. High efficiency is the premise for a training algorithm to be used in practical applications. For this purpose, the encryption technique we adopt should not contain computation-intensive cryptographic primitives, and our scheme ought to achieve minimal computing overhead.

2.4. Preliminaries

2.4.1. SVM Classifier. The support vector machine (SVM) is a widely used learning method based on the structural risk minimization [10]. The objective of SVM is to find an optimal separating hyperplane with the maximum margin that distinctly classifies the data points. Suppose that $(\overrightarrow{x}_i, y_i)$ is a pair of instance, where $\overrightarrow{x}_i$ is a vector containing attributes of the i^{th} instance and y_i is the class label which satisfies $y_i \in \{-1, +1\}$. The decision function to classify an instance $\overrightarrow{x}$ is $y = w^T \overrightarrow{x} + b$. The optimization problem of the SVM classifier is described as

$$\begin{aligned} &\arg\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i, \\ &\text{subject to} \quad y_i\left(w \cdot \overrightarrow{x}_i + b\right) \geq 1 - \xi_i, \xi_i \geq 0, \quad \text{for } i = 1, \ldots, m. \end{aligned} \tag{1}$$

In practice, the sample data are not linear separable in some classification tasks. That is, there exists no hyperplane which separates the data points. The nonlinear SVM can map the training data from the low-dimensional space into a higher dimensional space through a kernel function. The corresponding decision function is $y = \sum_{i=1}^{m} \alpha_i^* y_i K(\vec{x}_i, \vec{x}) + b$. The commonly used kernel functions include linear kernel, polynomial kernel, and Gaussian kernel. In this paper, we choose the Gaussian kernel which function is

$$K(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\left\|\vec{x}_i - \vec{x}_j\right\|^2}{2\sigma^2}\right). \tag{2}$$

2.4.2. Blockchain. Blockchain is a kind of distributed ledger technology (DLT) on a peer-to-peer (P2P) network which makes the history of any digital transactions unalterable and transparent through the use of decentralization and cryptographic hashing [11, 12]. The transactions are stored in blocks which are linked as a chain. Blockchain has been originally proposed for constructing a public-distributed ledger for transactions in Bitcoin [13], which is a worldwide electronic payment system.

Blockchain is well known for its decentralization, transparency, and tamper-proof property. It is built with no need for a trusted third party or a central administrator. There is no single-point-of-failure since the blockchain is not maintained by a single party but kept by all parties. The transactions that occur in the network would be recorded on the digital ledger, and no one can modify it. Blockchain usually adopts consensus protocols to manage the right of creating new blocks. There are three kinds of common consensus protocols, i.e., Proof-of-Work (PoW), Proof-of-Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT). Recently, there are lots of attempts of applying blockchain into different scenarios, such as solving trust crisis, simplifying various procedures, and verifying digital identities.

2.4.3. Secure Multiparty Computation. Secure multiparty computation (SMC) originates from the secure two-party computation (2PC) in 1982 (for the so-called Millionaires' Problem) introduced by Yao [14]. SMC is a central cryptographic task that allows multiple parties to jointly compute some function over their inputs while protecting the privacy. Specifically, there are multiple participants $P_1, P_2, \ldots P_n$. Each participant P_i holds its input x_i, and the participants agree on some function f that takes the n inputs. Their goal is to compute $y = f(x_1, x_2, \ldots, x_n)$ while satisfying the following two conditions:

(i) Correctness: the value of y is correctly computed

(ii) Privacy: the participants cannot learn anything about the inputs of others

Most of the SMC architectures are based on some cryptographic tools such as homomorphic encryption (HE), garbled circuits (GC), and oblivious transfer (OT). However, these frameworks are computationally expensive and difficult to be deployed for processing large-scale data sets. By contrast, SMC frameworks based on secret sharing do not involve any cryptographic primitives. Therefore, they could obtain a better performance.

3. Secret Sharing-Based Secure Computing Protocols

In this section, we mainly introduce the secure computation protocols used in our scheme which are based on additive secret sharing. Additive secret sharing is a kind of cryptography technique that means all intermediate values are additively shared between the chosen worker servers. Given two inputs a and b, they will be randomly split into two shares, i.e., $a = a_1 + a_2$ and $b = b_1 + b_2$, which are outsourced to two servers $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Here, we denote $\langle a \rangle_1$ and $\langle a \rangle_2$ as the two shares stored on $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. To cooperatively work out $f(a, b)$, $\mathcal{S}_1$ outputs f_1 and $\mathcal{S}_2$ outputs f_2. f_1 and f_2 satisfy that $f_1 + f_2 = f$. During the computation, $\mathcal{S}_1$ and $\mathcal{S}_2$ learn no information about the value of a, b, and the result f.

3.1. Secure Addition/Subtraction Protocol. Given two values a and b, the addition/subtraction protocol is to jointly compute $a \pm b$. It is obvious that the computation can be executed by $\mathcal{S}_1$ and $\mathcal{S}_2$ independently since $(\langle a \rangle_1 \pm \langle b \rangle_1) + (\langle a \rangle_2 \pm \langle b \rangle_2) = (\langle a \rangle_1 + \langle a \rangle_2) \pm (\langle b \rangle_1 + \langle b \rangle_2) = a \pm b$. Note that there is no interaction between the two servers during the computation.

3.2. Secure Multiplication Protocol. The multiplication protocol is to calculate the product of two given values a and b. We adopt the *Beaver's precomputed multiplication triplets* [15] technique to realize multiplication protocol. The steps of our secure multiplication protocol $\mathrm{SecMul}(\cdot)$ are given as follows.

To obtain $c = a \times b$, the algorithm utilizes a pregenerated triplet (u, v, w), where u and v are randomly generated and $w = u \times v$. The shares of u, v, and w are u_i, v_i, and w_i $(i = 1, 2)$, which are stored in $\mathcal{S}_i$, respectively. The servers $\mathcal{S}_i$ then calculate $\langle e \rangle_i = \langle a \rangle_i - \langle u \rangle_i$ and $\langle f \rangle_i = \langle b \rangle_i - \langle v \rangle_i$ locally. After that, they send $\langle e \rangle_i$ and $\langle f \rangle_i$ to each other and reconstruct e and f. Finally, $\mathcal{S}_i$ computes and outputs the shared results as $\langle c \rangle_i = f \cdot \langle a \rangle_i + e \cdot \langle b \rangle_i + \langle w \rangle_i + (i - 1) \cdot e \cdot f$.

Thus, the product c can be reconstructed simply by adding the respective results of $\mathcal{S}_1$ and $\mathcal{S}_2$ as $c = \langle c \rangle_1 + \langle c \rangle_2$.

3.3. Secure Comparison Protocol. Given a value a and b, the secure comparison protocol $\mathrm{SecComp}(\cdot)$ is used to judge whether $a < b$. Specifically, the function outputs 1 if and only if $a < b$ and outputs 0 otherwise. We adopt the bit-decomposition method in [16] and follow the comparison protocol proposed by Huang et al. [17] which is based on additively secret sharing.

We first transform the real-number shares into integers. Specifically, we multiply the numbers by 10^p and truncate the remaining decimal parts. Then, we utilize the two's complement representation, where the most significant bit (MSB) of a number indicates whether it is a positive or negative. For an l-bit signed number c, its binary complement form can be denoted as $c^{(l-1)}, c^{(l-2)}, \ldots, c^{(0)}$. Correspondingly, c can be reconstructed as

$$c = -c^{l-1} \cdot 2^{l-1} + \sum_{j=0}^{l-2} c^{(j)} \cdot 2^{j}. \tag{3}$$

Suppose that the l-bit shares of c are c_1 and c_2, and the protocol performs bitwise operations over c_1 and c_2 to compute the sign of c. Thus, the protocol can compare a and b by computing the sign of $a - b$.

3.4. Secure Natural Exponential Protocol. Given a value a, the secure natural exponential protocol $\mathrm{SecExp}(\cdot)$ is used to calculate e^a. The algorithm utilizes $\mathrm{SecMul}(\cdot)$ and the property of additive secret sharing. The shares of a are $\langle a\rangle_1$ and $\langle a\rangle_2$. $\mathcal{S}_1$ and $\mathcal{S}_2$ calculate $e^{\langle a\rangle_1}$ and $e^{\langle a\rangle_2}$ on the local servers, respectively. Note that $\langle a\rangle_1 + \langle a\rangle_2 = a$, and we can calculate e^a as

$$e^{\langle a\rangle_1} \cdot e^{\langle a\rangle_2} = e^{\langle a\rangle_1 + \langle a\rangle_2} = e^{a}. \tag{4}$$

Thus, $\mathcal{S}_1$ randomly splits $e^{\langle a\rangle_1}$ into two parts, i.e., $\langle e^{\langle a\rangle_1}\rangle_1$ and $\langle e^{\langle a\rangle_1}\rangle_2$, and sends $\langle e^{\langle a\rangle_1}\rangle_2$ to $\mathcal{S}_2$. Correspondingly, $\mathcal{S}_2$ randomly splits $e^{\langle a\rangle_2}$ into $\langle e^{\langle a\rangle_2}\rangle_1$ and $\langle e^{\langle a\rangle_2}\rangle_2$ and sends $\langle e^{\langle a\rangle_2}\rangle_1$ to $\mathcal{S}_1$. Finally, $\mathcal{S}_1$ and $\mathcal{S}_2$ conduct $\mathrm{SecMul}(\cdot)$ to calculate $e^{\langle a\rangle_1} \cdot e^{\langle a\rangle_2}$.

4. The Proposed Scheme

In this section, we present the framework of our privacy-preserving nonlinear SVM classifier training scheme. Our scheme contains two main parts: the blockchain design and the privacy-preserving SVM classifier training. The details are as follows.

4.1. Blockchain Design

4.1.1. Registering. The parties that want to join the decentralized network and become computing parties should first create a registering transaction in the blockchain. Each party would send a register request to the blockchain which is supposed to own at least \$ deposit in the account, which is to be frozen by the contract when registering. It is used to avoid malicious behaviors during the computing period. Specifically, the servers may reduce the quantity of training data to save computation resources. They may also forge inaccurate results and return them to the data providers.

4.1.2. Consensus Protocol. In the decentralized network, a consensus protocol is necessary to make all the nodes in the network reach a consensus. We adopt the Proof-of-Work (PoW) protocol as the consensus protocol in our scheme. The computing nodes increment a nonce in the block and compute the hash value of the block header. The first and the second nodes that find such a value that meets the predefined requirement by the contract would broadcast their results in the blockchain. After the other nodes verify the correctness of the results, the two nodes are accepted as the computing parties, and the first node has the right to create a new block.

4.1.3. Computing Request. The data provider who wants to outsource computation tasks to the computing nodes should first generate a transaction as $\mathrm{Tran}_{\mathrm{req}} =$ (protocols, \$ payment), in which protocols are the computing functions for the servers to execute and \$ pay is the payment. Then, the data owner publishes the request transaction on the blockchain.

After receiving the request from the data provider, the blockchain selects two servers for computing from the nodes in the network by PoW protocol and publishes the addresses $\mathrm{addr}_i\,(i = 1, 2)$ and the public key $\mathrm{pk}_i\,(i = 1, 2)$ of the two computing nodes.

4.1.4. Payment. If there are multiple data providers who own a set of training data and want to perform collaborative training, they should first reach a consensus about their payments of the training. In a practical scenario, each data provider usually owns different amounts of training data. The payments ought to be decided based on the amounts of data that the providers contribute. Specifically, the data provider who contributes a larger amount of training data ought to give less payment for the collaborative training. Once the data providers receive the results and believe the results can meet their requirements, they would send an acknowledgement message to the blockchain. If the blockchain receives more than two-thirds of the acknowledgement messages from data providers, it splits the total payment and distributes the payments to the two computing servers, respectively. Otherwise, the blockchain would return the payments to the data providers and deduct the fine from the deposits of the computing servers. The criterion of rewards and punishments can be adjusted in the consensus protocol before training.

4.2. Privacy-Preserving SVM Classifier Training

4.2.1. System Initialization. Suppose that there are n data providers $\mathrm{DP}_j\,(1 \le j \le n)$ who own some training data, respectively, and want to collaboratively train an SVM classifier with a kernel function. The data providers first reach a consensus on the training protocols, parameters, and payments. Then, they send the transactions of request to the blockchain and thereafter receive the addresses and public keys of the computing servers.

After that, the data providers first encrypt the training data by randomly splitting each element into two shares. Then, they encrypt the shares with the corresponding public key pk_i of the two computing servers and obtain the encrypted training datasets $\langle D_j\rangle_i$. Finally, the data providers send the encrypted datasets to the two servers, respectively, and then publish the transactions on the blockchain.

4.2.2. Training. After receiving all the encrypted datasets from the data providers, the servers decrypt the shares by utilizing their corresponding private key sk_i and perform the training protocol. In our SVM model, we choose the Gaussian kernel function, which is depicted in equation (2), to achieve nonlinear separation. The function can be interactively calculated by the two servers. The steps of calculating the Gaussian kernel are shown in Algorithm 1.

To train the SVM classifier, we adopt the gradient descent (GD) as the optimization method, which is utilized in [9]. Compared with another optimization algorithm that is also frequently used in plaintext tasks, i.e., the sequential minimal optimization (SMO), GD contains less complex computation. Thus, it is regarded to be more suitable for the training in the encrypted domain. By introducing a hinge loss, the optimization problem of the SVM is converted to

$$\min \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{m} \max\left(0, 1 - y_i\left(\sum_{j=1}^{m} \alpha_j K\left(\overrightarrow{x}_i, \overrightarrow{x}_j\right) + b\right)\right). \tag{5}$$

The protocol firsts executes SecComp$(\cdot)$ to compare $y_i(\sum_{i=1}^{m} \alpha_i K(\overrightarrow{x}_i, \overrightarrow{x}) + b)$ and 1. If $y_i(\sum_{i=1}^{m} \alpha_i K(\overrightarrow{x}_i, \overrightarrow{x}) + b) < 1$, the servers update α and b by calculating the derivatives of the margin and the hinge loss. The steps of privacy-preserving training are shown in Algorithm 2. The dataset for training and the SVM model is well protected during the training process. The servers and other adversaries cannot infer any information except the respective shares obtained in each step.

5. Security Analysis

In this section, we present the security strength of our proposed scheme under the two-level threat models. First, we analyze the security of our scheme under the Level 1 threat model based on the universal composability (UC) framework [18], which is regarded to guarantee strong security properties. To prove the security of our scheme, we first give some definitions as follows.

Definition 1. A protocol is secure if there exists a probabilistic polynomial-time simulator $\mathcal{S}$ that can generate a view for the adversary $\mathcal{A}$ in the real world which is computationally indistinguishable from its real view.

Due to the secret sharing-based protocols, the addition and subtraction operations which are computed locally on the servers can be easily simulated. We prove the security of other computing protocols in our scheme.

Theorem 1. *The protocol SecMul$(\cdot)$ is secure under the honest but curious model.*

Proof. The view of $\mathcal{S}_1$ is view$_1$ = $(a_1, b_1, u_1, v_1, w_1, e_2, f_2)$. It is obvious that a_1 and b_1 are randomly split from a and b and u_1, v_1, and w_1 are uniformly random values. e_2 and f_2 are also random values because they are generated as $e_2 = a_2 - u_2$ and $f_2 = b_2 - v_2$. The output of $\mathcal{S}_1$ is view$_1$ = $f \cdot a_1 + e \cdot b_1 + w_1$, which is also uniformly random. Note that both the input and output of $\mathcal{S}_1$ are random values, so they can be perfectly simulated by $\mathcal{S}$. The view of adversary $\mathcal{A}$ and its real view are computationally indistinguishable. Similarly, the input and output of $\mathcal{S}_2$ can also be perfectly simulated.

Theorem 2. *The protocol SecComp$(\cdot)$ is secure under the honest but curious model.*

Proof. For the comparison protocol SecComp$(\cdot)$, the view$_1$ and view$_2$ of $\mathcal{S}_1$ and $\mathcal{S}_2$ are view$_1$ = (a_1, u_1, v_2) and view$_2$ = $(a_2, u_2, v_1, v_2^{(l-1)}, v_2^{(l-2)}, \ldots, v_2^{(0)})$. The values are random and simulatable. The bitwise addition can be deployed by secure addition and secure multiplication, which has been proved to be simulatable. Therefore, it can be proved that the comparison protocol can be simulated by a simulator $\mathcal{S}$.

Theorem 3. *The protocol SecExp$(\cdot)$ is secure under the honest but curious model.*

Proof. The SecExp$(\cdot)$ protocol in our scheme only involves natural exponential computation, subtraction, and secure multiplication. The first two operations are implemented locally on the servers. The secure multiplication is proved that it can be simulated. Thus, the secure natural exponential computation is simulatable. A view can be generated for the adversary $\mathcal{A}$, and the view is computationally indistinguishable with its real view.

The privacy-preserving training protocol is composed of the above computing protocols, which are proved to be secure. Thus, our privacy-preserving nonlinear SVM classifier training scheme is secure against the Level 1 threat. In the Level 2 threat model, the servers can obtain some original training data from certain data providers. However, the shares of other training data are still randomly generated. Thus, for the datasets from other data providers, the simulator $\mathcal{S}$ can still generate the view that is computationally indistinguishable from its real view. Therefore, the security of the other training data and the protocols can still be guaranteed under the Level 2 threat model.

6. Performance Evaluation

In this section, we evaluate the performance of our scheme by conducting experiments over real-word datasets. We use a real-world dataset about social network advertising collected from a trusted website. The dataset contains 400 instances. Each instance is with 4 features and labeled as purchased or not purchased. We also use the Breast Cancer Wisconsin Database (BCWD) from the UCI machine learning repository. The dataset contains 699 instances, and each instance contains nine features. The instances in the dataset are labeled as benign or malignant. We implement the experiments on a PC with a 32-core Intel i7 CPU @ 1.80 GHz and 16 GB RAM. The algorithms are programmed with Python 2.7. Specifically, we investigate the performance through accuracy and efficiency.

```
Input:$\mathcal{S}_1$: the shared vectors $\langle \vec{x}_a \rangle_1$, $\langle \vec{x}_b \rangle_1$, and $\sigma$; $\mathcal{S}_2$: the shared vectors $\langle \vec{x}_a \rangle_2$, $\langle \vec{x}_b \rangle_2$, and $\sigma$.
Output:$\mathcal{S}_1$: the shared Gaussian kernel result $\langle r \rangle_1$; $\mathcal{S}_2$: the shared Gaussian kernel result $\langle r \rangle_2$.
$\mathcal{S}_i$ initialize $\langle s \rangle_i = 0$.
for $k$ from 1 to $len(\langle \vec{x}_a \rangle_i)$do
    $\mathcal{S}_i$ locally compute $\langle z \rangle_i \longleftarrow \langle x_a[k] \rangle_i - \langle x_b[k] \rangle_i$.
    $\mathcal{S}_i$ compute $\langle g \rangle_i \longleftarrow \mathrm{SecMul}(\langle z \rangle_i, \langle z \rangle_i)$.
    $\mathcal{S}_i$ locally compute $\langle s \rangle_i \longleftarrow \langle s \rangle_i + \langle g \rangle_i$.
end for
$\mathcal{S}_i$ locally compute $\langle f \rangle_i \longleftarrow -(1/2\sigma^2) \cdot \langle s \rangle_i$.
$\mathcal{S}_i$ compute $\langle r \rangle_i \leftarrow \mathrm{SecExp}(\langle f \rangle_i)$.
```

Algorithm 1: Secure Gaussian kernel function.

```
Input:$\mathcal{S}_i$: the split dataset $\langle D \rangle_i = \{(\langle \vec{x}_1 \rangle_i, \langle y_1 \rangle_i), (\langle \vec{x}_2 \rangle_i, \langle y_2 \rangle_i), \ldots, (\langle \vec{x}_m \rangle_i, \langle y_m \rangle_i)\}$ learning rate $\lambda$, max iterations $T$, and
    precision $\varepsilon$.
Output:$\mathcal{S}_i$: $\langle \alpha^* \rangle_i$, $\langle b^* \rangle_i$.
$\mathcal{S}_i$ initialize $\langle \alpha^1 \rangle_i, \langle b^1 \rangle_i, \langle \mathrm{loss} \rangle_i$.
for $p$ from 1 to $m$ do
    for $q$ from 1 to $m$ do
        $\mathcal{S}_i$ compute $\langle K[p][q] \rangle_i \longleftarrow \mathrm{SecKer}(\langle \vec{x}_p \rangle_i, \langle \vec{x}_q \rangle_i)$.
    end for
end for
While loss $> \varepsilon$ or $t < T$ do
    $\mathcal{S}_i$ initialize $\Delta_\alpha, \Delta_b$.
    $\mathcal{S}_i$ compute $\langle \mathrm{loss} \rangle_i \longleftarrow \mathrm{SecMul}(\mathrm{SecMul}(\langle \alpha^t \rangle_i, \langle K \rangle_i), \langle \alpha^t \rangle_i^T)$.
    for $p$ from 1 to $m$ do
        for $q$ from 1 to $m$ do
            $\mathcal{S}_i$ compute $\langle g^p \rangle_i \longleftarrow \mathrm{SecMul}(\langle \alpha^t[q] \rangle_i, \langle K[p][q] \rangle_i)$.
            $\mathcal{S}_i$ locally compute $\langle s^p \rangle_i \longleftarrow \langle s^p \rangle_i + \langle g^p \rangle_i$.
        end for
        $\mathcal{S}_i$ locally compute $\langle f^p \rangle_i \longleftarrow \langle s^p \rangle_i + \langle b^p \rangle_i$.
        $\mathcal{S}_i$ compute $\langle f^p \rangle_i \longleftarrow \mathrm{SecMul}(\langle y_p \rangle_i, \langle f^p \rangle_i)$.
        $\mathcal{S}_i$ compute $\mathrm{SecComp}(\langle f^p \rangle_i, 1)$.
        if $\langle f^p \rangle_i < 1$then
            $\mathcal{S}_i$ compute $\langle \Delta_\alpha \rangle_i \longleftarrow \langle \Delta_\alpha \rangle_i - C \cdot \mathrm{SecMul}(\langle y_p \rangle_i \cdot K[p])$.
            $\mathcal{S}_i$ compute $\langle \Delta_b \rangle_i \longleftarrow \langle \Delta_b \rangle_i - C \cdot \langle y_p \rangle_i$.
        end if
    end for
    $\mathcal{S}_i$ update $\langle \alpha^t \rangle_i \longleftarrow \langle \alpha^t \rangle_i - \lambda \cdot \langle \Delta_\alpha \rangle_i$.
    $\mathcal{S}_i$ update $\langle b^t \rangle_i \longleftarrow \langle b^t \rangle_i - \lambda \cdot \langle \Delta_b \rangle_i$.
    $\mathcal{S}_i$ compute $\mathrm{SecComp}(\langle \mathrm{loss} \rangle_i, \varepsilon)$.
    $t = t + 1$.
end while
$\mathcal{S}_i$ return $\langle \alpha^* \rangle_i, \langle b^* \rangle_i$.
```

Algorithm 2: Secure nonlinear SVM classifier training.

6.1. Accuracy. The precision rate and the recall rate are two key parameters to evaluate the accuracy of a classifier. We calculate the two parameters by utilizing both our proposed privacy-preserving SVM classifier and the traditional SVM classifier over plaintext. The results are shown in Table 1. We can see that our scheme can achieve nearly the same accuracy with the SVM classifier over plaintext. It demonstrates that the cryptographic methods in our scheme do not influence the classification functionality. Our scheme can maintain high accuracy while protecting privacy.

Table 1: Accuracy performance.

	Our scheme (%)	SVM over plaintext (%)
Precision	85.8	86.7
Recall	92.8	91.3

6.2. Efficiency. In this section, we evaluate the efficiency of our scheme. Specifically, we investigate the time consumption both on the data providers and the servers by utilizing cross-validation. We evaluate the time consumption with different percentage of instances for training and

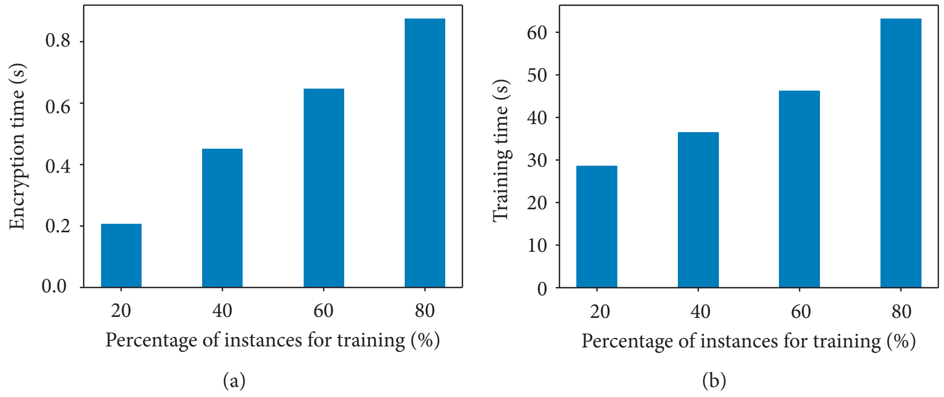


Figure 2: Efficiency performance with different percentage of instances.

testing, as shown in Figure 2. Specifically, we take several percentages of instances for training and the others for testing. We can see that the time consumption of training data encryption is positively correlated with the percentage of instances for training because more instances mean a larger number of computation both on the data provider side and the server side. Overall, the efficiency of our scheme is acceptable for practical use.

6.3. Comparison. We also compare our scheme with the privacy-preserving SVM classifier training scheme proposed by Shen et al. We use the BCWD dataset and conduct training both on our scheme and Shen et al.'s scheme. The results are shown in Table 2. We can see that our scheme can achieve higher accuracy than Shen's scheme. It is because our scheme adopts the nonlinear kernel, which makes the scheme more adaptive to the datasets that contain nonlinear data. As for the efficiency, it is shown in Table 3 that our scheme achieves much better efficiency performance. It is because our scheme does not involve any computationally expensive cryptography techniques. We can see that the time consumption on the data provider side in our scheme is just 1.4 s, while in Shen et al.'s scheme, it takes the data provider more than 2000 s. The time of computation on the server side is also much less than Shen et al.'s scheme. The experiment results show that our scheme is much more efficient than Shen et al.'s scheme and achieves better overall performance for practical utilization.

Table 2: Comparison of the accuracy.

	Our scheme	Shen's scheme (%)
Precision	93.3%	90.35
Recall	1	96.19

Table 3: Comparison of the efficiency.

	Our scheme (s)	Shen's scheme (s)
Server side	146	953
Data provider side	2.6	2233

7. Related Work

Classification is a fundamental task of machine learning and applied to many fields, such as face recognition, speech recognition [19], and financial prediction. To protect the privacy of individuals and enterprises, there have been many privacy-preserving classifier training schemes proposed. These schemes focus on training classifiers and performing classification tasks over encrypted data while protecting user privacy. Bos et al. [20] proposed a scheme to privately conduct medical predictive analysis tasks on encrypted data. However, their scheme cannot protect the model from being exposed to the users. In addition, the scheme leaks much information of the patients. Raphael Bost et al. [5] proposed privacy-preserving protocols for three common classifiers, i.e., hyperplane decision, Naïve Bayes, and decision trees. González-Serrano et al. [21] proposed a privacy-preserving semiparametric SVM scheme by utilizing a partial homomorphic cryptosystem. Recently, Shen et al. [9] proposed a privacy-preserving SVM training scheme based on the blockchain for secure IoT data sharing. They utilized the Paillier encryption technique to design secure building blocks and achieve secure SVM training. However, these schemes all adopt the homomorphic encryption and contain computationally expensive cryptographic primitives, which make the schemes inefficient.

Some existing schemes are based on a combination of homomorphic encryption and multiparty computation. Barni et al. [22] proposed a privacy-preserving neural network classification scheme. In their algorithm, the neural networks contain a series of scalar products which are encrypted by utilizing the homomorphic encryption. The activation functions are calculated based on the secure multiparty computation. Subsequently, Orlandi et al. [23] enhanced the scheme of Barni. They masked the scalar product results and protected the intermediate results from revealing to the client. However, these schemes also suffered heavy computation overheads.

There are also a number of privacy-preserving training schemes that adopt differential privacy. Pathak and Raj [24] presented a scheme to learn a discriminatively trained multiclass Gaussian mixture model-based classifier that preserves differential privacy using a large margin loss function. Zhang et al. [25] designed a privacy-preserving decision tree classification construction model based on a differential privacy-protection mechanism. Nevertheless, in these schemes, there have been conflicts between privacy and accuracy. The differential privacy-protection technology by adding immeasurable noises influences the accuracy of the models.

8. Conclusion

In this paper, we proposed a new privacy-preserving nonlinear SVM classifier training scheme for social networks. We utilize the blockchain technique to design a decentralized framework for data sharing and training while ensuring the invariance of datasets. We adopt additive secret sharing based on secure two-party computation and design a suite of secure computing protocols to conduct the training process with no information leakage. Our training scheme is proved to be secure through comprehensive analysis. Experiments over real datasets demonstrate that our scheme can achieve high accuracy and efficiency for practical applications.

Data Availability

The data used to support the findings of this study are available at https://github.com/JIANAN17/privacy-preserving-SVM.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the National Natural Science Foundation of China under Grant nos. 61672195 and 61872372, the Open Foundation of State Key Laboratory of Cryptology (no. MMKFKT201617), and the National University of Defense Technology (Grant no. ZK19-38).

References

[1] J. Hua, H. Zhu, F. Wang et al., "CINEMA: efficient and privacy-preserving online medical primary diagnosis with skyline query," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1450–1461, 2018.

[2] Z. Ma, Y. Liu, X. Liu, J. Ma, and K. Ren, "Lightweight privacy-preserving ensemble classification for face recognition," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5778–5790, 2019.

[3] M. Ghazanfar and A. Prugel-Bennett, "An improved switching hybrid recommender system using naive bayes classifier and collaborative filtering," in *Proceedings of the International Multiconference of Engineers and Computer Scientists 2010*, Hong Kong, China, March 2010.

[4] X. Yang, Z. Liu, and Li Jin, *Security and Privacy in Social Networks and Big Data*, Springer, Berlin, Germany, 2020.

[5] R. Bost, R. Ada Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proceedings of the NDSS Symposium 2015*, p. 4325, San Diego, CA, USA, February 2015.

[6] H. Zhu, X. Liu, R. Lu, and H. Li, "Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 838–850, 2016.

[7] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research: JMLR*, vol. 12, pp. 1069–1109, 2011.

[8] M. Abadi, A. Chu, I. Goodfellow et al., "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, Vienna, Austria, October 2016.

[9] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted iot data in smart cities," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7702–7712, 2019.

[10] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[11] T. Aste, P. Tasca, and T. Di Matteo, "Blockchain technologies: the foreseeable impact on society and industry," *Computer*, vol. 50, no. 9, pp. 18–28, 2017.

[12] H. Huang, X. Chen, Q. Wu, X. Huang, and J. Shen, "Bitcoin-based fair payments for outsourcing computations of fog devices," *Future Generation Computer Systems*, vol. 78, pp. 850–858, 2018.

[13] M. Crosby, Nachiappan, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: beyond bitcoin," *Applied Innovation Review*, vol. 2, no. 6–10, p. 71, 2016.

[14] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pp. 160–164, Chicago, IL , USA, November 1982.

[15] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Proceedings of the Annual International Cryptology Conference*, pp. 420–432, Santa Barbara, CA, USA, August 1991.

[16] I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, "Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation," in *Proceedings of the Theory of Cryptography Conference TCC 2006*, pp. 285–304, New York, NY, USA, March 2006.

[17] K. Huang, X. Liu, S. Fu, D. Guo, and M. Xu, "A lightweight privacy-preserving CNN feature extraction framework for mobile sensing," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[18] C. Ran, A. Cohen, and Y. Lindell, "A simpler variant of universally composable security for standard multiparty computation," in *Proceedings of the Annual Cryptology Conference*, pp. 3–22, Santa Barbara, CA, USA, August 2015.

[19] Z. Liu, Z. Wu, T. Li, J. Li, and C. Shen, "GMM and CNN hybrid method for short utterance speaker recognition," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3244–3252, 2018.

[20] J. W. Bos, K. Lauter, and M. Naehrig, "Private predictive analysis on encrypted medical data," *Journal of Biomedical Informatics*, vol. 50, pp. 234–243, 2014.

[21] F.-J. González-Serrano, Á. Navia-Vázquez, and A. Amor-Martín, "Training support vector machines with privacy-protected data," *Pattern Recognition*, vol. 72, pp. 93–107, 2017.

[22] M. Barni, C. Orlandi, and A. Piva, "A privacy-preserving protocol for neural-network-based computation," in *Proceedings of the 8th Workshop on Multimedia and Security, MM&Sec 2006*, pp. 146–151, Geneva, Switzerland, September 2006.

[23] C. Orlandi, A. Piva, and M. Barni, "Oblivious neural network computing via homomorphic encryption," *EURASIP Journal on Information Security*, vol. 2007, no. 1, pp. 1–11, 2007.

[24] M. A. Pathak and B. Raj, “Large margin Gaussian mixture models with differential privacy,” *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 4, pp. 463–469, 2012.

[25] L. Zhang, Y. Liu, Y. Liu, R. Wang, X. Fu, and Q. Lin, “Efficient privacy-preserving classification construction model with differential privacy technology,” *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 170–178, 2017.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8850472, 21 pages
https://doi.org/10.1155/2020/8850472

Research Article

Hidden Service Website Response Fingerprinting Attacks Based on Response Time Feature

Yitong Meng and Jinlong Fei

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

Correspondence should be addressed to Jinlong Fei; feijinlong@126.com

Received 20 August 2020; Revised 8 October 2020; Accepted 10 November 2020; Published 1 December 2020

Academic Editor: Zhe-Li Liu

It has been shown that website fingerprinting attacks are capable of destroying the anonymity of the communicator at the traffic level. This enables local attackers to infer the website contents of the encrypted traffic by using packet statistics. Previous researches on hidden service attacks tend to focus on active attacks; therefore, the reliability of attack conditions and validity of test results cannot be fully verified. Hence, it is necessary to reexamine hidden service attacks from the perspective of fingerprinting attacks. In this paper, we propose a novel Website Response Fingerprinting (WRFP) Attack based on response time feature and extremely randomized tree algorithm to analyze the hidden information of the response fingerprint. The objective is to monitor hidden service website pages, service types, and mounted servers. WRFP relies on the hidden service response fingerprinting dataset. In addition to simulated website mirroring, two different mounting modes are taken into account, the same-source server and multisource server. A total of 300,000 page instances within 30,000 domain sites are collected, and we comprehensively evaluate the classification performance of the proposed WRFP. Our results show that the TPR of webpages and server classification remain greater than 93% in the small-scale closed-world performance test, and it is capable of tolerating up to 10% fluctuations in response time. WRFP also provides a higher accuracy and computational efficiency than traditional website fingerprinting classifiers in the challenging open-world performance test. This also indicates the importance of response time feature. Our results also suggest that monitoring website types improves the judgment effect of the classifier on subpages.

1. Introduction

In the face of diversified attacks, anonymous communication platforms such as Tor [1], I2P [2], and Zeronet, take advantage of the protocol features to protect the data security and identity privacy for both parties involved in digital communications over Internet. For instance, Tor provides low-latency interaction to meet users' needs for anonymous access. On a daily basis, Tor embeds hidden service to protect anonymity of 150,000 websites. In order to monitor the hidden services that usually provide illegal services in violation of the user policy, regulators put forward attack technologies based on different levels of traffic and protocol as countermeasures. Instances of common legacy attack technologies include active association attack [3], information disclosure attack [4], and node attack [5]. Biryukov et al. [4] proposes to control the key nodes to send 50 padding packets and detect server's packet drop feedback. He did not however make any actual test and evaluation. Matic [6] uses service certificate vulnerabilities to detect the certificate chain of hidden services for tracking purpose; however, the achieved success rate is only 79%.

In order to avoid the problems associated with elevated conditions and low real-world test accuracy in conventional hidden service attacks, here we use mature website fingerprinting attacks [7–9]. This enables us to reexamine the website response and further achieve effective hidden service attacks based on analyzing the implicit information of hidden service response traffic. Website fingerprinting attacks are known for being capable of destroying the anonymity of transmission at the traffic level. The main goal of such an attack is to match the website contents (i.e., website pages and its addresses) generated by the Tor users. Attackers are local observers, including local network administrators, Internet service providers (ISP), autonomous system administrators (AS), and other network backbones. The attacker's attack capability is also amplified. If the observer is near the monitoring server, the attacker can passively analyze the response encrypted traffic generated by the website and then record the two-way transmission features of data packet. These features are then matched with the self-built response traffic template library to restore the monitoring website content and track the website server.

In this paper, a novel hidden service attack technique, hidden service website response fingerprinting (WRFP), is proposed based on response time feature. We show, for the first time, that in a real communication environment, the WRFP attack is in fact capable of threatening hidden services. The attack scenario is shown in Figure 1. WRFP utilizes response fingerprinting dataset of hidden service, then designs website mirroring, and builds different mounting modes of the same-source or multisource servers, thus accurately replicating the website response traffic status. WRFP further analyzes 87 combined features including response time feature with an extremely randomized tree algorithm. It is shown that the proposed method successfully categorizes the index pages and subpages, websites, and service types of hidden services. The contributions of this paper are listed in the following:

(i) We propose a novel hidden service WRFP attack which is different from the traditional website fingerprinting attack. WRFP monitors the website pages, its types, and website's servers simultaneously. Furthermore, the higher the target, the better the classification effect will be.

(ii) We collect 300,000 pages of hidden service WRFP datasets to achieve the same benefit of data traffic of the real hidden service website. The datasets include the construction and mounting of two service scenarios of the same-source and multisource servers.

(iii) We propose a combination feature based on response time to select and optimize the traffic feature. We then show rationality and effectiveness of the response time feature using test evaluations.

(iv) We compare the recognition performance and computational efficiency of WRFP with other website fingerprinting classifiers and show that its classification performance is higher than that of k-NN [7], CUMUL [8], and k-FP [9]. We also test the advanced fingerprinting defense model and show that the lightweight website fingerprinting defense model is unable to effectively resist response fingerprinting attacks.

The rest of this paper is structured as follows. Section 2 introduces the deployment form of website fingerprinting attack and hidden service server and explains the attack hypothesis and targets. Section 3 summarizes the methods and effects of traditional website fingerprinting attacks and analyzes the results and deficiencies of various attacks on hidden services. Section 4 is to design website mirroring and website mounting methods and to collect the response fingerprinting dataset of hidden service websites. Section 5 is concerned with the response time measurement method and proposes the WRFP classifier based on the response time feature. Section 6 cross-evaluates the WRFP in the closed and open worlds and further conducts tests for the network-type subpage classification, TBB version changes, and defense models. Section 7 summarizes the paper and proposes the future work.

2. Background

2.1. Website Fingerprinting Attack. Evidences show that the success of the website fingerprinting attack technique is mainly due to the fact that attackers can capture and analyze the statistical feature of data packets exposed at the traffic level. This compromises the reliability of anonymous networks. This is shown in Figure 2(a) as the attacker analyzes the original encrypted communication data, extracts the transmission data at the cell, TLS, or TCP level [8], and then obtains features such as packet length, direction, and sequence. These features are then matched with the feature sequence of the monitoring website pages. In Figure 2(b), the data traffic deformed by fingerprinting defense is shown. The essence of website fingerprinting defense [10–13] against attacks lies in the real-time adjustment of the exposed features of the traffic, the quantitative changes in the statistical feature that attacks rely on, and obscuring the transmission trajectory of real websites. These are adjustments made to deceive the classifier. Conventional methods include analyzing the diversified traffic feature attack classifiers, improving the resistance of website fingerprinting defense models, reducing the accuracy of attack classifiers, and enhancing the security and anonymity of network transmission.

By evaluating the classifier performance of a website fingerprinting attack, the attacker tries to choose a closed world with a single target or a complicated open world based on the number of monitors [9]. In the closed world, restricting the monitoring quantity and accessing range of websites have the advantages of small training data, complete presupposition, and low monitoring cost. Therefore, it is the first choice for the attackers to evaluate the basic performance of the classifiers. However, in the open world, the real communication scenario is simulated and the number of unmonitored website pages in the big background is far larger than those of the monitored pages. This means that the classifier needs to comprehensively judge the feature information of unmonitored websites, to be able to pose a viable challenge to the performance of the classifier.

2.2. Website Server. We usually suppose that the service type of a website represents its service goal. To examine the content of monitoring hidden services, we classify the websites into eight categories, including store, search, e-mail, news, forum, social, porn, and others. We further

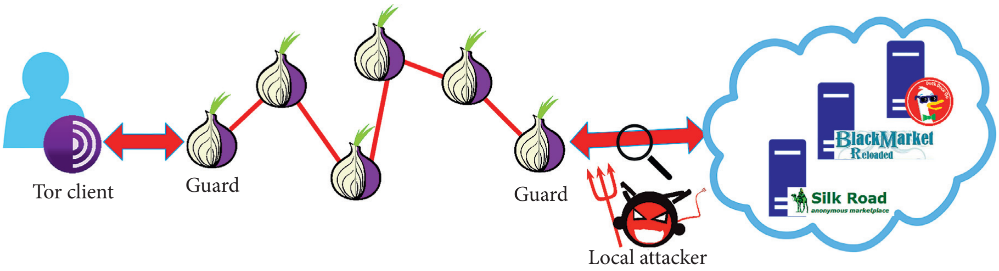


Figure 1: Hidden services response fingerprinting attack scenario.

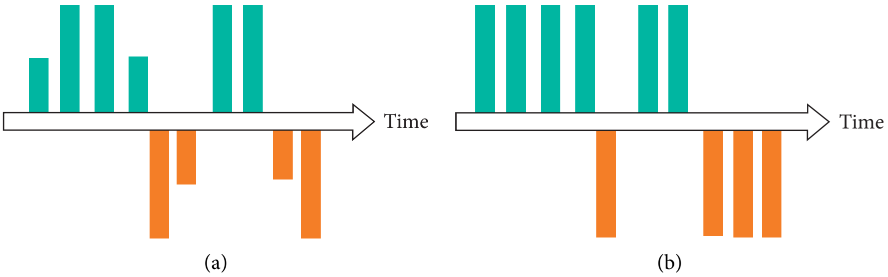


Figure 2: Website fingerprinting attack and defense data traffic. (a) Original encrypted communication data traffic. (b) Fingerprinting defense deformed data traffic.

note that a website relies on the carrying capacity of the server and responds to web data according to the internal scale of the server. Therefore, we consider two scales for the server configuration: same-source server and multisource server (Figure 3).

The same-source server is the basic configuration, where single or multiple website domain names are deployed in the same server space. To avoid issues such as insufficient server performance or data lost caused by multiuser responses, the two-way traffic of each website is concurrently processed on the service port. For instance, port 443 in Figure 3 is an external service port. By expanding the scale of the website or increasing the number of visitors, the website resources should be divided and deployed in a cluster server. In a cluster server, distributed servers, SQL servers, load balancing servers, and other functions cooperate with each other to provide the response to the needs of the website. Such a setting is collectively referred to as the multisource server. In a multisource server, the internal response traffic of the servers is then collected and provided through the service port for unified forwarding. This is different from that of the same-source servers in terms of traffic profile. In this paper, we focus on the response fingerprinting analysis of servers with two scales.

2.3. Attack Hypotheses. Juarez [14] pointed out that attacks on traffic fingerprints requires reliable attack hypotheses, and the success of website response attacks further depends on the hypotheses in the mirror mounting stage, fingerprint capturing, and response time measurement. In the mirror mounting stage of the monitoring website, it is assumed that the attacker has the ability to obtain the software and hardware configuration of the monitoring website server and copy and reconstruct the service content of the website page. It is further assumed that the attacker is capable of forwarding the response traffic according to the same-sized same-source or multisource server configured by software and hardware. This hypothesis is preliminarily verified for suitability in Section 4.2.

In the response fingerprint collection stage, it is assumed that the attacker, when it is monitoring locally, is able to judge the start and end of a single page request and response traffic and further filter out the remaining noise traffic. This enables the attacker to effectively identify and isolate the server responding to multiple response traffic of the same domain name. Wang's [15] research shows that the page parsing assumption is in fact a difficult task. In the stage of extracting the features of the response time, it was necessary to assume that the attacker is less disturbed by the environmental factors before and after the traffic is captured, and the overall time volatility is within the receiving range. Our experiments reported in Section 6.3 show that, by sliding the response time within 10% of the prediction range, the classifier keeps a normal judgment.

2.4. Attack Goal. Our goal is to attack the server mounted by the hidden service, that is, to monitor the server that provides the specified website at the traffic fingerprint level. The test in Section 6.3 shows that the recognition effect of the classifier can be improved by focusing the target on the server. The accuracy is also improved by focusing the target on the type of website service. At the same time, the classifier supports classification of the same-source and multisource servers and realizes the corresponding recognition effect in the face of different monitoring targets. In general, the response fingerprinting attack is different from the traditional

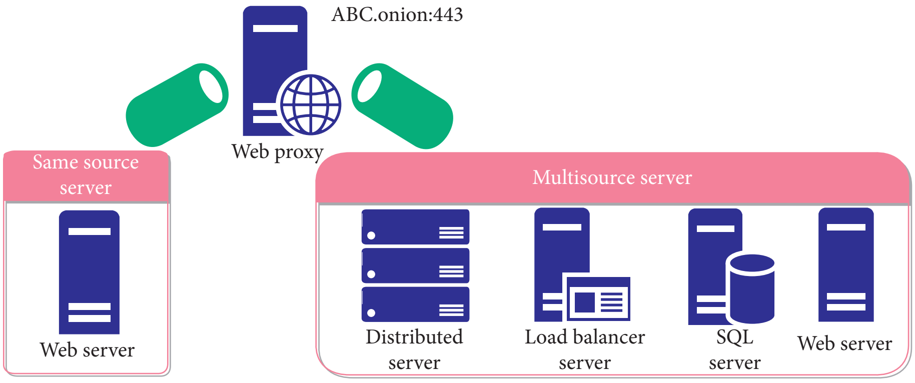


Figure 3: A schematic of the web server deployment.

fingerprinting attack, which is only a single target to restore the trajectory of user's behavior.

3. Related Works

3.1. Traditional Fingerprinting Attack. Website fingerprinting attacks are based on the passive traffic analysis in the initial stage where the tunnel traffic with data encryption techniques such as VPN, HTTPS, and SSH is identified. For the first time, Herrmann [16] uses the traditional data packet length feature to evaluate the protection performance of Tor website and obtained a modest webpage recognition accuracy rate of 2.95%. In fact, Herrmann's work turns the research goal in the field of website fingerprinting to Tor and tries to build a website fingerprinting model that is different from tunnel traffic recognition. Cai et al. [17] then optimizes Herrmann's recognition algorithm by using string-based edit distance algorithm and SVM model to characterize the communication data packet sequences. Their method achieves 86% recognition accuracy for 100 websites in the closed-world scenario. LashKari et al. [18] focus on time based traffic features, introduce 32 features, and show that the time features can be used to classify the type of communication type. The accuracy of the proposed method is not however comparable with that of the previous works.

Using deep learning techniques further increases the recognition accuracy of the classifiers. In 2016, Kota and Goto [19], instead of manual selection of the feature, use a SDAE autoencoder and achieve an accuracy of over 86% with simple input data. Later, Rimmer et al. [20] explore the automated recognition results provided by a variety of deep learning algorithms; however, to obtain a reasonable accuracy, their approach requires access to a rather large dataset including up to 2500 instances per page. In 2018, Sirinam et al. [21] propose a recognition model of deep fingerprinting (DF) which is trained and optimized based on CNN. This is the first classifier that is capable of producing effective attacks on WTF-PAD fingerprinting defense. Bhat et al. [22] further optimize the DF, to obtain an accuracy of 97.8%. They use a semiautomatic characteristic extraction to improve Var-CNN classifier understanding of the nature of the input data based on a small training dataset and short training time.

Although fingerprinting classifiers based on deep learning are in the mainstream of improving the recognition accuracy, they often lack characteristic interpretability and thus unable to express the differences between fingerprints compared with the traditional machine learning algorithms. In other words, deep learning weakens the statistical feature exposed by the encrypted traffic and prioritizes the accuracy of the classifier [23]. Therefore, in order to explore the degree of characteristic influence of response fingerprints, the following three representative machine learning classifiers are chosen for testing and comparison with the algorithm proposed in this paper.

3.1.1. k-NN Classifier. Wang et al. [7] propose a website fingerprinting classifier based on k-nearest neighbors (k-NN) and extract 3736 combined features such as the quantity, order, and single packet length of the load. The Euclidean-like distance function suitable for the recognition is then used to measure the distance similarity between features. Then, the weighted value of characteristic distance is adjusted to measure the efficiency of the classification. They show that k-NN is able to reach 85% TPR by monitoring 100 pages in the open world scenario.

3.1.2. CUMUL Classifier. Panchenko et al. [8] propose a CUMUL classifier based on the SVM algorithm which achieved 92% recognition accuracy for 100 pages in the closed-world scenario. CUMUL expresses the tracked cumulative sum of packets as a length cumulative vector to minimize the differences in bandwidth, congestion, and webpage loading time. A total of 300,000 real page fingerprint sets are collected to ensure the rationality of the classification results.

3.1.3. k-FP Classifier. Hayes and Danezis [9] propose a more advanced k-FP classifier in terms of web page recognition accuracy, and its recognition accuracy to Alexa pages and hidden service pages is both above 90%. k-FP is a set of webpage classifiers based on random forest and k-NN. It creatively uses random forest leaves to represent Hamming distance to achieve webpage classification and systematically analyzes the information benefits of 175 features. The test result shows that using the first 40 combined features results in the highest classifier accuracy.

3.2. Hidden Service Attack. In 2008, Zander and Murdoch [24] propose an attack model based on hidden service request clock drift, and the results indicate that it had a certain effect, but the attack condition depended on the number and time of arrival of hidden service requests. Elices and Perez-Gonzalez [3] improved the request arrival time prediction, taking each request as an interval as a reference, thus improving the positioning effect of hidden services. Biryukov et al. [4] found that the server would make abnormal feedback after sending the padding packet to the key position of the joint rendezvous node and guard entry node. Ling et al. [5] manipulated the transmission load by using the similarity principle and confirmed the hidden service IP address by analyzing the protocol-level feature of the load at multiple intersection points. In 2015, Matic et al. [6] designed the detection tool CARONTE to locate the actual server IP address by detecting the unique string and certificate chain leaked in the content of the hidden service. Tan et al. [25] used the Eclipse attack to destroy the DHT structure of the hidden service directory server, which could cover any hidden service under a small amount of IP resources.

In terms of the research objectives, compared with the previous active attack methods, we tended to realize attacks based on passive traffic analysis theory. As early as 2015, when Kwon et al. [26] realized the classification of Hidden Service link fingerprints, they also initially explored the threat of website fingerprints to hidden services. Although the experiment obtained 88% TPR, for the lack of consideration of some objective factors such as actual server software and hardware configuration, environment, and internal relations, this research is just a reapplication of the website fingerprinting attack scenario, and it is not helpful to the reliability and effectiveness of response fingerprinting data analysis.

4. Dataset

During the hidden service response period, the confusing traffic between the server and the client is an unequal traffic; therefore, the previously disclosed website fingerprinting dataset cannot be used. This is because of objective reasons such as network disturbance, load encapsulation, and imbalance between the demand and transmission link capacity. Here, we collect a set of universal response fingerprinting dataset of hidden service websites combined with the actual server size, server mirroring, and mount mode. The dataset provides a reliable data basis for the classifier training.

4.1. Hidden Service Addresses. The addresses of hidden services are opaque and need to be collected through public crawling, network detection, and memory extraction. The public collection methods include summarizing addresses through Hidden Wiki, Real-World Onion Sites, Daniel's Hosting, and other means. The nonpublic collection methods consist of network detection and memory extraction. Network detection is based on Elasticsearch engine, combined with large search sites for large-scale hidden service link scanning. Memory extraction [27] is much more hidden and uses a self-built server that temporarily applies for and obtains the HSDir tags to extract memory from the short-term upload addresses.

Considering the rapid offline of some home pages and Tor privacy protection of Tor, only memory addresses within 96 hours are extracted. It should be also noted that servers with versions higher than Tor 0.3.5 give priority to v3 addresses. If the hidden service provides both v2 and v3 addresses, only v3 addresses are stored. During the collection, according to the HTTP status code and curl error code of the websites, the invalid, duplicated, and censored addresses are also deleted, and a total of 34,890 active website addresses are kept (Table 1).

4.2. Website Mirroring. As mentioned in Section 2, accurate simulation of the communication behavior in the real environment is one of the key steps in fingerprint collection. Taking the response traffic of DuckDuckGo as an example, when the response fingerprinting sets of hidden service are explored, the issue with the accuracy of image matching needs to be solved. In the present study, a self-controlled exit node is set to visit the normal site (http://duckduckgogo.com) and the mirror site (http://3g2upl4pq6kufc4m.onion). The response traffic (T and RT) is then recorded. The response packets are also arranged in equal proportions in time (Figure 4). The mean value of 50 subpages is also calculated. The traffic of mirror image (RT_1) is displayed on the top of Figure 4, and the combined traffic of mirror image and multisource server mount (RT_2) is shown in the bottom:

$$1 > \text{Array}\left(\sum \frac{RT_2}{T}\right) > \text{Array}\left(\sum \frac{RT_1}{T}\right). \quad (1)$$

A closer value of the cumulative sum of RT/T in mirror matching to 1 indicates a higher sensitivity of the packet can be maintained to time. Therefore, after evaluating the relationship between the site and server image, it is determined to build a combination mode of website image and website mounting to provide a real website response traffic.

The content of mirroring is stipulated. Public information such as website templates, service content, and addresses is included. Furthermore, server's architecture technology stack, file layout, storage, and other private information are detected with the modified OnionScan supporting tools. Public information refers to the main presentation form of the website service content. First, a website crawler tool is developed based on requests and selenium to obtain complete HTML, CSS, JavaScript, and other structural data. The internal and external links of JSON, JavaScript, and CSS language are then automatically rewritten, such that the style of the website pages is not lost and even the address links of dynamically generated images cannot be mirrored correctly. Secondly, we use Flask lightweight framework to build the site and Cchardet module to assist in confirming the coding method of the

Table 1: The number of collected hidden service website addresses.

Collection method	Addresses collected	Active addresses	
		v2	v3
Public crawling	24,591	13,130	8461
Network detection	16,345	10,328	4017
Memory extraction	53,452	36,234	16,218
Total		34,890	

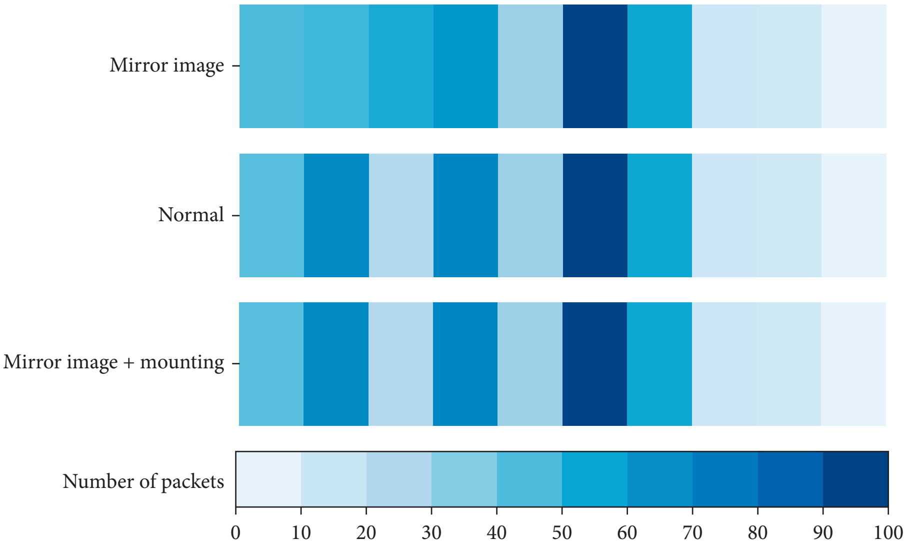


Figure 4: Comparison of response traffic of mirror image of DuckDuckGo.

website to ensure uniform mirror content. Finally, to ensure the balance between each type of website and the number of accepted subpages, the collection and processing of subsequent subpages of each website are determined according to the website type. The specific crawling depth of the website type is shown in Table 2. Websites of e-mail type need logging in and cannot be effectively crawled; therefore, only the login page is recorded as the index page.

Privacy information represents the response performance of the server; fortunately, information collection can be done without controlling the server itself. It is also found through scanning OnionScan that about 55% of hidden service websites are equipped with the classical technology stack combination of Nginx + PHP, and 38% of websites have a common file layout structure, such as /style, /images, /css, and other major directories. To optimize website hosting, the key exchange in the handshake phase of HTTPS is also ignored, and all HTTPS sites are simplified to HTTP sites. Different from the TLS protocol, in Tor transmission tunnel, the SSL certificate is only provided through simple handshake by HTTPS exchange protocol. Therefore, this cannot affect the subsequent page element transmissions. Optimization of the network bandwidth, server requirements, and internal links is conducted within the acceptable range. For example, timely deletion of access.log could effectively solve the problem of slow response.

The website relies on the server for page response. Further, its relationship with the server is usually one-to-one, i.e., one server mounts one domain name website. To meet the practical tactical needs, the relationship between website and server is also expanded to one-to-many and many-to-one cases. In this study, considering that the server configuration determines the quality of the site's response, based on the potential impact of the server size of the hidden service mount, the mounting scenarios of websites are classified into the three following categories:

4.2.1. Same Source-Single Website (SS-SW). In this scenario, the server only provides basic web services to the website. The response of the website can be optimized by using the performance of the server, and the server is very inclusive to the performance of the website. This scenario is mostly selected for temporary small websites.

4.2.2. Same Source-Multiple Websites (SS-MW). Many addresses in hidden services are attached to the reliable large server providers such as Daniel's Hosting, Ablative Hosting, and Kowloon Hosting. Each server of the server providers mounts multiple independent domain name websites. Here, the configuration mode of Daniel's Hosting is selected and copied in constructing scenarios in this study. To ensure the effective quarantine of agents between the websites, independent domain names are used for website addresses. The configuration of the hidden service hosting server is the basic configuration of Nginx + PHP + NTP, equipped with independent website registration function. Moreover, the server in the group passed the load performance stress test of the

Table 2: Selection of crawling depth for different types of websites.

Website type	Store, search, e-mail	News, porn	Others	Forum, social
Page properties	Index page	Index page + subpage	Index page + subpage	Index page + subpage
Crawling depth	Level 1	Level 2	Level 3	Level 4

Locust framework. It is found that if more than 235 domain name websites are mounted, the response failure rate will be gradually increased and the response time fluctuated frequently. Considering the bandwidth difference of the websites, 150 response ports are opened in the server to meet the load balance of multisite responses.

4.2.3. Multisource-Single Website (MS-SW). To improve the availability threshold of large-size websites, the load balancing server, web server, SQL server, and distributed server are combined into a server cluster. Among them, the distributed server provides several types of response elements, the SQL server mounts the background data, and the load balancing server balances the load of the website response traffic. By receiving large-scale request traffic, the server cluster automatically retransmits the traffic to the server members according to the response rules and responds to the specified resources.

4.3. Website Mounting. The mirror image of the website is automatically mounted according to the server size. The inconsistency of mirror response may occur between websites as well as servers. Therefore, in feasible mounting scenarios, the mounting is performed with reasonable rules according to the website configuration and data space. As shown in Algorithm 1, to avoid the intersection of website data, each domain name is only mounted in one server scenario.

Steps 5–11 of the algorithm: the website is placed in the SS-MW scenario if the server configuration matches with the configuration of classical technology stack. If the data space exceeds the maximum allotted space by the server, the MS-SW scenario is then imported.

Steps 12–15: the website is placed in the SS-SW scenario, if the underlying configuration of the current site is not in records. However, the response failure is likely, where the website is placed into the matching scenario. To exclude this situation, the HTTP access status code is detected and the abnormal website is manually debugged. In addition, the number of automatic control servers is much smaller than the total number of websites. The website mounting status is presented in Table 3, where the rotational mounting mechanism of the website is considered. The number of groups is the number of rotational mounting of the website:

4.4. Collection of the Response Traffic. To ensure the controllability of the response traffic and to prevent hidden service from being accessed by outsiders, we build a small Tor network based on the website image reserve. The collection of response traffic from resource deployment to traffic data processing is divided into the following five steps:

Step 1. Resource deployment: we build a small autocontrol network composed of 40 relay nodes. It is equipped with necessary components such as hidden service directory server, authoritative directory server, and client agent. The release and update time of hidden service address are also revised to 15 seconds. Eight Amazon EC2 servers are mounted with the website content of different scenarios. Among them, one server was set as a large hidden service hosting server and another 4 formed a server cluster.

Step 2. Synchronization of client with server: inside the server is a rotational mounting mechanism. NFS is built to allow the client to share the address list and website resources with the server. Moreover, the client is permitted to modify the list. When the website is visited successfully, the header address is deleted, and the server automatically reads the most recent address for mirror image mounting.

Step 3. Traffic collection the request-response cooperation between the client and the server realizes the collection of traffic. The client traverses the list of hidden services in order and starts the tor-browser-crawler [14] self-running program. The interval between the visiting website and the internal page is set to 20 seconds, to have a sufficient traffic response and website replacement time. We then use Munin to monitor the traffic at the backend of website server. When the bandwidth of the service port is suddenly increased, it cooperates with tcpdump to collect the traffic. After completion, server sends destroy instruction packet to the link through the stem controller, and the client restarts the Tor program synchronously. This eliminates the impact of data retention and continuation while keeping the complete response packet. Finally, the server implements permuted access to the list of websites. Therefore, when the website response traffic is successfully collected, the current website is immediately gone offline and the mirror agent of the presorted website is turned on. Compared with the multiple visits to the website in a single round, the advantage is that the response of the website will not be affected by the internal environment of the server and will not miss the opportunity to get a fresh and correct response packet. The data are collected in a total of 50 rounds, and the same number of instances of page traffic is obtained from each website.

Step 4. Traffic processing: although the collection process reduced the possibility of traffic anomalies, there are potential interference packets. Packet filtering and packet retransmission operations are in place to ensure that the response traffic within each instance is similar to others. There were five types of packets at the TCP level which are involved in packet filtering including: missing, repeated response, hierarchical transmission, window

```
Input: Mounted website collection (Webm), website public information (Webopen), website privacy information (Webprivacy),
website configuration (CF), and website data space (DS)
Output: Server scenario (Sce)
Steps:
$(\mathrm{Web}_m, \mathrm{Web}_{\mathrm{open}}, \mathrm{Web}_{\mathrm{privacy}}) \longleftarrow \mathrm{GetImformation}(\mathrm{Web}_m)$
Records $\longleftarrow$ WebsiteConfig (Records)
for each Website $w \in \mathrm{Web}_m$ do
$w$.CF $\longleftarrow$ GetConfig ($\mathrm{Web}_{\mathrm{privacy}}$), $w$.DS $\longleftarrow$ GetConfig ($\mathrm{Web}_{\mathrm{open}}$)
if $w$.CF $\subseteq$ Classical technology stack then
    if $w$.DS $\leq$ Maximum allotted space then
        return Sce $\longleftarrow$ $w$.SS − MW
        else
            return Sce $\longleftarrow$ $w$.MS − SW
        end if
    end if
if $w$.CF not in Records then
        return Sce $\longleftarrow$ $w$.SS − SW
    Records $\longleftarrow$ $w$.CF
end if
end for
```

Algorithm 1: Algorithm of website mounting.

Table 3: Same-source and multisource mounting mechanism.

Server size	Scenario	Website × server/group	Rounds
Same source	SS-SW	1 × 1	6457
	SS-MW	1 × 4	3579
Multisource	MS-SW	150 × 1	122

update, and ACK loss packets. The transmission packets with zero data length are also removed to directly filter packets that interfere with packet classification without providing reliable information. Packet retransmission improves the property value gap between the website instances. Winsorization [28], an effective method for routine detection of abnormal values in sample data, is also adopted to detect the site response duration property. The duration larger than 95 quantiles in the property is defined as an outlier, and revisiting is required to complete the instance. About 17% of the packets on the website are processed, and 6% of the page instances need to be retransmitted.

Step 5. Data conversion: to facilitate the data input of the instance which is interpreted by the classifier, the demand data are converted into a combination of symbols and values, including direction (d), time (t), and packet length (l) that are input in the form of $d = \{-, +\}$. The specific conversion formula is as follows:

$$\mathrm{Dump}(\mathrm{TCP}) = \left(d_1(t_1, l_1), d_2(t_2, l_2), \ldots, d_n(t_n, l_n)\right). \tag{2}$$

5. WRFP Attack

5.1. Response Time Features. Here, we investigate the effective function of the response packet of the website. The combination of response time features is then proposed to highlight the implicit information hidden in the response time and response time frequency features. The measurement of hidden service response time is illustrated in Figure 5. The attacker monitors encrypted TCP packets locally, records the standard time of each incoming or outgoing packet, and then calculates the time difference of the two-way packet group, that is, the difference between the start time of a set of outgoing server packets and the end time of the incoming packets, i.e., ΔRT = IT + OT.

For the response time measurement specified in Algorithm 2, the attacker records the specific TCP packet time and calculates the response time frequency as well as the binning information. Specifically, steps 2–8 indicate that a set of continuous incoming packet time (IT) and continuous outgoing packet time (OT) is regarded as a response time unit, and the difference between the start time (T.start) and the end time (T.end) of the response is considered as the response time (ΔRT). The preset number of bins (RT.Bins) is included in step 9. Steps 10–16 indicate that the response time is looped into the response frequency bins according to the specified range (Range), and the output is the response time frequency array (ΔRT.F).

Figure 6 is a visual display of the response time frequency array. The internal response time of the two popular websites is calculated, and the global time is inserted into 20 bins. The upper layer of Figure 6 summarizes the response time of the forum website http://kbhpodhnfxl3clb4.onion. There is one bin with response time of 20 ms, and the response time

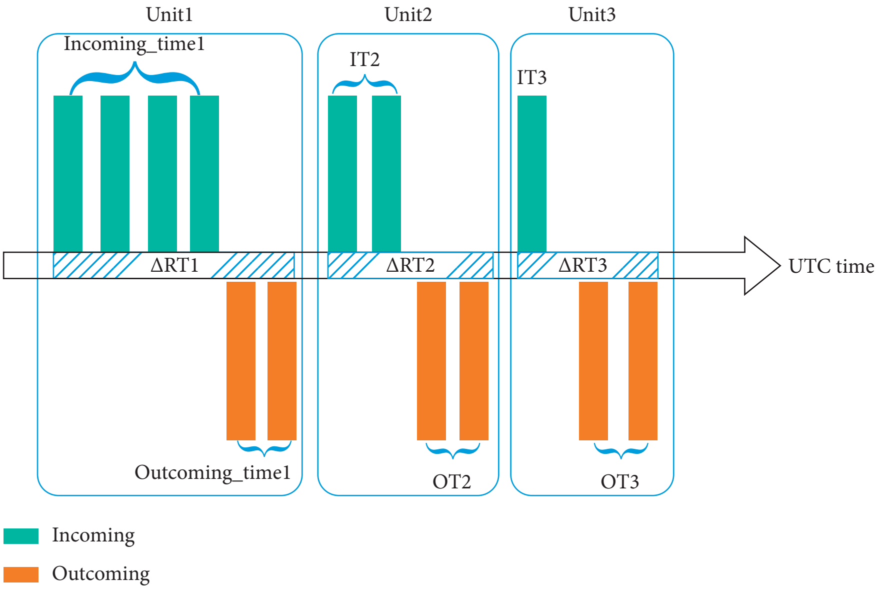


Figure 5: Measurement of the response time.

```
    Input: the continuous incoming and outgoing TCP packet collection ($T$), response time frequency bins (RT_Bins)
    Output: Response time (ΔRT), response time frequency array (ΔRT.F)
    Steps:
        $T$.start ⟵ $T_1$.time, Range ⟵ 0, RT.bins ⟵ bins
      While $i \leq$ Length ($T$) do
        if $T[i]$.in is outgoing packet and $T[i+1]$.out is incoming packet then
          $T$.end ⟵ $T[i]$.time
          return Add $T$ (ΔRT) ⟵ $T$.end − $T$.start
          $T$.start ⟵ $T[i+1]$.time
        end if
      end
      Timerange ⟵ (Max (ΔRT) − Min (ΔRT))/RT.bins
    for $j \leq$ RT.bins do
      if Range ≤ ΔRT ≤ Timerange + Range then
        ΔRT.F[$j$] + +
      end if
      Range ⟵ Timerange + Range
      return ΔRT.$F$
    end for
```

Algorithm 2: Response time measurement algorithm.

220–240 ms appears three times. The lower layer of Figure 6 displays the response time of the news website http://3g2upl4pq6kufc4m.onion, for which the minimum response time is 53 ms, the maximum response time is 669 ms, and the frequency of the 3 bins is 4 times. Compared with what is in the upper layer of Figure 6, the response time density is more concentrated in the lower layer. This suggests that the response frequency of the website represents the difference between the response of the server and the load behavior of the website. It is therefore inferred that this kind of feature can be used to extract the response classification information of different websites.

5.2. Feature Selection. It is pointed out in a number of studies [7–9] that the accuracy and robustness of website fingerprinting attacks are strongly correlated with the demand features. It is also noted that with the development of website fingerprinting attack technology, statistical features such as incoming, outgoing, burst, and total number of packets are capable of improving the performance of the attack model. It is however suggested that time characteristics are fragile and no consensus has been reached about time [29]. Hence, it is an immediate need to verify the importance and inevitability of response time features in website response fingerprinting, according to

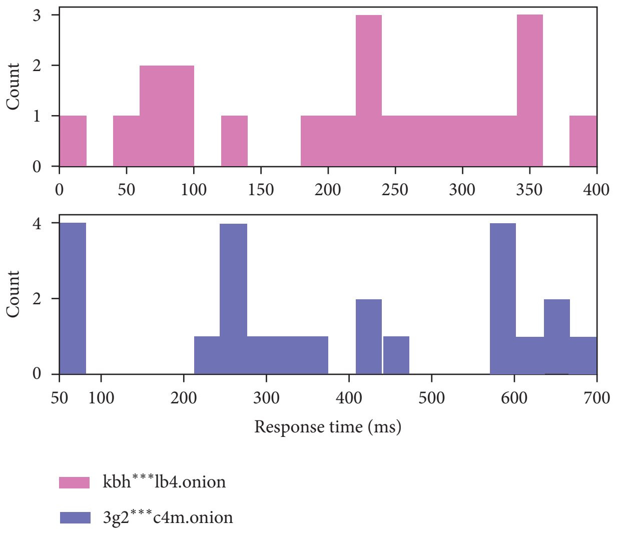


Figure 6: Visualization of the website response time frequency.

the ranking and recognition results of many kinds of features.

For feature weight ranking, we use an extremely randomized tree regression classifier on the response traffic label dataset. To obtain the best feature combination, the overall feature ranking is carried out according to the degree of influence of each preselected feature on the model prediction result.

The traditional feature classification standard is based on Gini or information entropy coefficient [30] which may ignore property overlap of some of the features. In a group, if a feature variable is set as a strong predictive variable, the importance of its associated feature variable is decreased. Note that most of fingerprinting traffic features are, in fact, combined features and tend to include multicategory variables. Therefore, in the feature ranking, the importance of excellent features might be reduced. To address this issue, we select R-square mean as the classification standard to ensure the importance of a single feature and the fair ranking of the excellent features. R-square mean measures the basic accuracy of the feature reduction model, which is regarded as the remeasurement of the model accuracy by inverting a single feature value. For instance, a feature score of 0.02 means that, after replacing the feature data, the accuracy of the attack model is reduced by 2%. As far as the results are concerned, for low-importance feature variables, an arbitrary change of the value has a slight impact on the accuracy of the model, while a disturbance on an important feature leads to a significant decrease in the model accuracy. In this study, the features are divided into main features and effective features according to their score, and the measurement range is the mean score as follows:

$$\text{main} \geq \left\lceil \frac{\sum_{i=1}^{j} \text{scores}\,[i]}{j} \times 100\% \right\rceil > \text{effective} \geq 0. \qquad (3)$$

The main features have a range of score greater than 0.02, and the score of the effective features is in the range 0 and 0.02. For features with a score smaller than 0, the prediction of the attack model is biased towards the worse result.

Figure 7 shows the main feature ranking corresponding to the response size of multisource and same-source servers after 50 raking rounds of 87 features, with 25 main features for multisource server and 23 for same-source server. The importance of each feature is not exactly the same between the two sizes. The results show that in the two response sizes, the number of packets and response time play a leading role in judging the attack, and the number of packets, sequence information, and response time information is the primary judgment basis of the attack. Among the load key information of the total number of incoming and outgoing packets, the total number of outgoing packets ranked first, with mean scores of 0.18 and 0.16. The proportion of incoming packets ranked in the top 7 in both sizes, with scores greater than 0.1. Meanwhile, 15 burst packet features have the largest changes, which are ranked the 10th with a score of 0.067 in the same-source server size, while they are ranked the 19th in the multisource server size, with a score of 0.034.

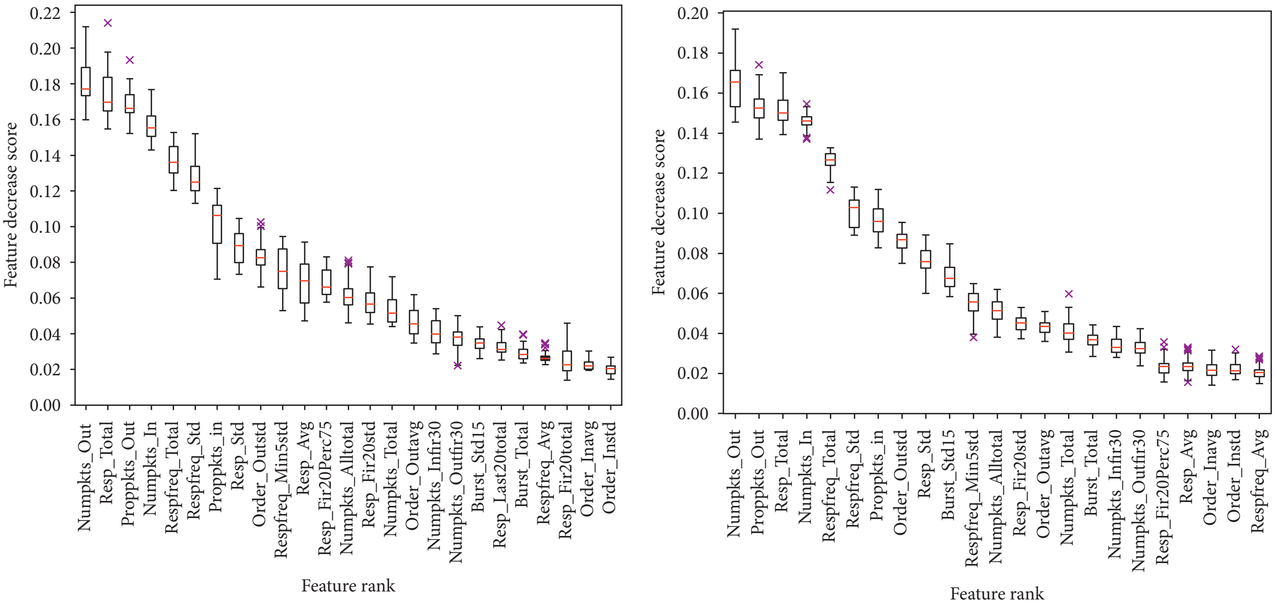
(i) Response time features: in the multisource server size, the mean score of the total response time is 0.17, which is only lower than that of the total number of outgoing packets (rank 8) and the standard deviation (rank 11). The response time of the first 20 packets ranked between 12 and 19, and the total response time of the last 20 packets ranked 20 with a score of 0.031.

(ii) Response time frequency features: in the same-source server size, the total number of response time frequency is ranked 5 with a score of 0.13, in line with the expectation. The standard deviation of response time frequency is ranked 6 with a score of 0.1. The mean response frequency is ranked 23, with a score of only 0.02. The standard deviation feature of the 5 bins with minimum response time frequency is ranked 11 with a score of 0.058.

The first 32 valid features of the same-source server are shown in Figure 8. Burst packets and transmission time provide auxiliary support to attack classification. The total transmission time of outgoing packets is ranked 28, with a score of 0.017, which was 1/10 of that of the total response time. The main reason is that the website response time has a higher stability and reliability than the distance transmission time.

Figure 9 summarizes the data transmission process. The transmission intervals corresponding to response time ΔT_3 and ΔT_4 are short, and the processing is simpler than that of the distance transmission response time ΔT_1 and ΔT_2. This is consistent with the view of Hayes [9] that the statistical features of packet interarrival times slightly improve the attack accuracy, and the information disclosure ability of the response time is much higher than that of the packet interarrival times.

The number of selected features is an important factor. Hayes [9] shows that the first 30 of the 150 features ensure 90% accuracy of the classifier. Panchenko [8] optimizes the



(a)

(b)

FIGURE 7: Rankings of main features of the multisource and same-source servers response sizes. (a) Main features of multisource server. (b) Main features of same-source server.

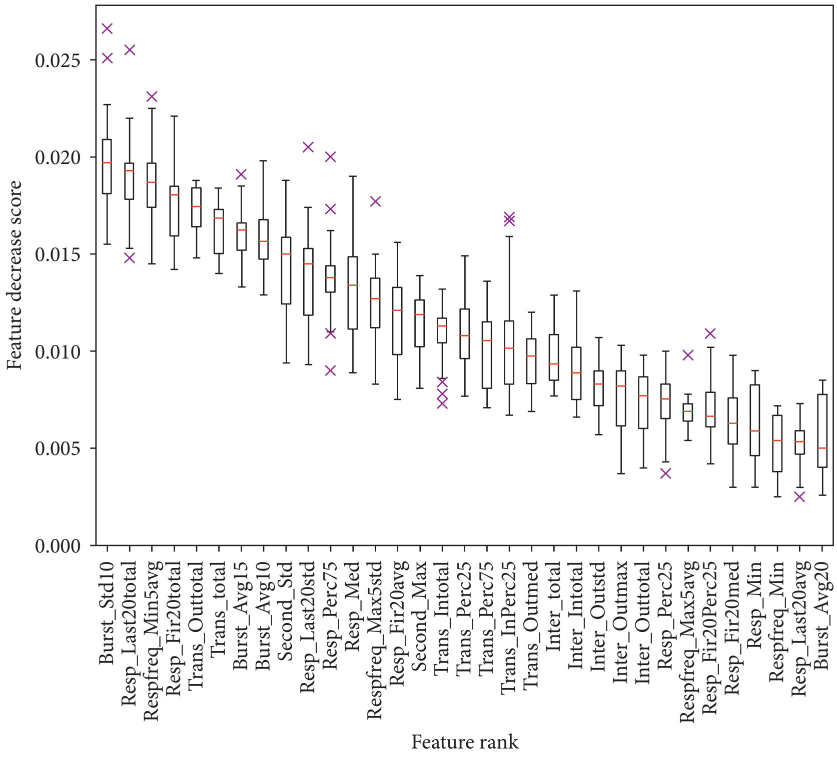


FIGURE 8: Ranking of the first 32 valid features of the same-source server (score >0.012).

number of cumulative features to 100 to improve the efficiency of CUMUL. These show that an excessive number of features are inversely proportional to the classification return, and further, the optimization of the number of features is able to maximize the accuracy of the model within the demand range. To determine the optimal number of features for the performance of the classifier, the changes in the number of features (started from 10) and the model accuracy variations of the two sizes are compared. The changes in and the relationship between the number of

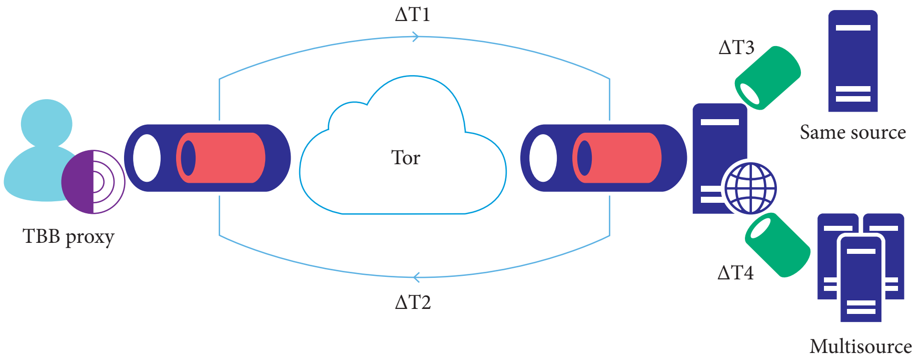


Figure 9: Data communication and transmission.

features and the accuracy are displayed in Figure 10. The accuracy of the first 24 features is increased rapidly, which is inline with the corresponding increase in the ranking degree of the main features. However, the accuracy is oscillated for the number of features larger than 70 in both same-source and multisource server sizes. This suggests that a reasonable choice for the number of the features is 73 or 82. Accuracy greater than 94% also meets the needs of the attack, and the optimized numbers prevailed in the subsequent tests.

5.3. Process of Response Attack. Here, we propose a set of standard WRFP attack processes which follow the basic construction phases of fingerprinting classifier including response attack training phase and response attack matching phase. The response attack training phase is divided into the training and WRFP classifier performance testing. The training process is shown in Figure 11 and includes the following steps. Step 1: the response traffic of the website mount image is collected. Each mirror site repeatedly responds for 50 rounds to obtain the original TCP fingerprinting dataset. Step 2: features are extracted from the dataset, labeled, and bound to the website page or server to form a feature tag library that is used in the subsequent training steps, as well as the attack matching phase. Step 3: The extremely randomized tree classifier with a training to test ratio of 8 : 2 is then applied to the website response fingerprinting dataset to build a hidden service website response fingerprinting classifier. Step 4: Cross-validation of the test set is carried out. Step 3 is repeated until the verification classification result is at its highest value. The attack model is considered as the final WRFP classifier.

It is stipulated that the trained WRFP classifier is capable of conducting response fingerprinting attacks. This because the attacker has the ability to monitor the locally encrypted TCP traffic. The attack matching process is shown in Figure 12. Firstly, the attacker marks the locally encrypted TCP traffic. Secondly, the statistical information of the load packet is extracted and converted into the response eigenvalues. Finally, after accumulating the packets observing and monitoring the numbers, the extremely randomized tree classifier generates the matching results of the website server, including the website tags matched under the same-source and multisource server sizes, as well as the website server that provides mounting.

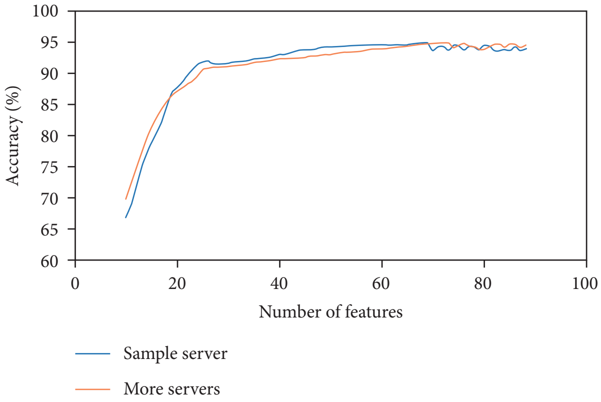


Figure 10: Changes in, and the relationship between the number of features and the accuracy.

Here, the extremely randomized tree [31] is used which is different from the random forest classifier in the way of selecting the division points in a single decision tree. The random feature splitting method is added to randomly obtain bifurcation values. This random feature selection greatly reduces the amount of training computation. The extremely randomized tree randomly selects the samples when constructing the data subset and further randomly extracts the features of the samples. In other words, when building the model, part of the features is used for training. Comparison of the classification performance between the extremely randomized tree and random forest is shown in the first two rows of Table 4. As it is seen, the test speed of extremely randomized tree is improved by 102%.

6. Experimental Results and Analysis

6.1. Evaluation of Dataset. To realize the rationality of evaluation, we use a hidden service response fingerprinting dataset collected in real network environment. The sizes and properties of the datasets are presented in Table 5, and the website size is expressed as the number of website pages multiplied by the number of fingerprint instances. In the results of dataset classification, the training effectiveness of the classifier is based on same-source and multisource server response fingerprints. The response fingerprints of 90,000 independent pages are collected, with 50 instances for each

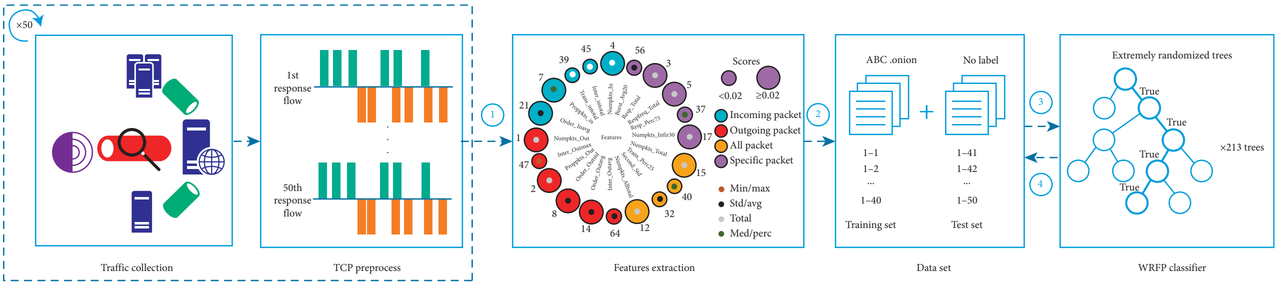


Figure 11: Training process of the WRFP classifier.

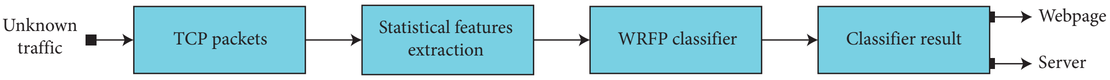


Figure 12: WRFP attack matching process.

Table 4: Performance comparison of different algorithms in the classifiers.

Classifier	Algorithm	Accuracy (%)	Test time* (hour)	Parameters	Features	Memory** (GB)	Page
WRFP	Extremely randomized tree	93.63	0.43	$n = 213$	87	8	10,000
	Random forest	91.82	0.87	$n = 487$	87	8	10,000
K-FP	Random forest + Nearest neighbors	92.35	1.14	$K = 5$, $n = 379$	186	65	10,000
K-NN	K-nearest neighbors	88.14	1.65	$K = 5$	1225	10	10,000
		88.06	4.88	$K = 5$	3736	16	10,000
CUMUL	SVM-RBF	90.78	0.55	$C = 4096$	150	30	10,000

*Test time refers to the instance feature extraction time + page classification time. **Memory denotes the minimum required memory size.

Table 5: Classification of datasets needed for evaluation.

Dataset	Data description	Data size	Website size	TBB version
Same-source server	Response fingerprinting of same-source server	40,000 × 50	11,500	8.5
Multisource server	Response fingerprinting of multisource server	50,000 × 50	15,000	8.5
Background	Response fingerprinting of big background	200,000 × 10	25,000	8.5
TBB version	Response fingerprinting of multiple TBB versions	4000 × 50	100	6.5/7.5
Defense	Fingerprinting of multiple types of defenses	8000 × 50	100	8.5

page. Besides, a big background response fingerprinting is constructed to facilitate the understanding the recognition effect of the classifier in open world. Response fingerprints of a total of 200,000 pages are collected, covering 25,000 websites. To evaluate the impact of TBB version and website fingerprinting defense on the classifier, we collect 200,000 response fingerprints of version 6.5 and 7.5, with 2000 (pages) × 50 (instances) for each defense model.

6.2. Evaluation Indexes. The test evaluation indexes proposed in this study are aimed at the evaluation of the response fingerprinting classifier in multiple scenarios. In addition to the conventional accuracy of classification, other indexes provide practical insights on performance of the classifier.

6.2.1. Precision and Recall. Precision refers to the probability of recognizing correctly classified monitored pages. Recall refers to the probability that the monitored pages are classified as correctly monitored pages. Precision and recall affect each other. In tests and evaluation, a high precision is preferred to reflect the correct classification effect of the classifier under a big background.

6.2.2. True Positive Rate (TPR) and False Positive Rate (FPR). TPR is equivalent to recall, while FPR refers to the probability that unmonitored pages are misclassified as monitored pages. The combination of the two indexes shows the performance of the classifier; however, if they are affected by the fallacy of the basic rate, these indexes become bad references.

6.2.3. Bayesian Detection Rate (BDR). BDR refers to the probability that a given classifier can correctly judge a monitored page when it is recognized as a monitored page. This index includes the ratio of the monitored pages to the total pages, to a certain extent eliminating the classification influence caused by the basic rate fallacy in open world, and highlighting the correct classification results in the big background. Juarez [14] and Hayes [9] apply BDR in their study for the first time and show that this index is feasible:

$$
\begin{aligned}
\mathrm{BDR} &= \frac{\mathrm{TPR} \times \Pr(M)}{\mathrm{TPR} \times \Pr(M) + \mathrm{FPR} \times \Pr(U)}, \\
\Pr(M) &= \frac{|\text{monitored}|}{|\text{total pages}|}, \\
\Pr(U) &= 1 - \frac{|\text{monitored}|}{|\text{total pages}|}.
\end{aligned}
\tag{4}
$$

6.3. Closed-World Test. In this section, the performance of WRFP classifier is tested in closed world. The classification results of monitoring a small number of hidden service websites is simulated, and the response fingerprinting of same-source or multisource servers is judged. A total of 300, 600, and 900 website pages were monitored, with 40 instances for each page as the foreground training set and the other 10 instances as the test set. The background pages are fixed with 10,000 × 10 random pages, and ten-fold cross-validation is performed to ensure the rational use of the dataset.

The test results are shown in Table 6. The accuracy of monitoring 300 pages is 96.7%, with a TPR higher than 94%. By increasing the number of monitored pages to 600, the accuracy declined by 2%, indicating that if the attacker wants to get the best recognition results, the attacker needs to control the number of monitored pages. In monitoring 600 multisource pages, it is promising that the FPR reaches 0.4%. By comparison, it is seen that the accuracy of recognizing same-source pages is 2% higher than that of recognizing multisource pages. This however does not mean that the classifier has a weaker ability to classify multisource servers. Instead, it may be attributed to data jitter in multisource servers which may cause deviation in the correct classification of multisource pages.

The distribution of response time frequency is determined by the bin value, and the setting of bin value affects the granularity that divides time frequency. Therefore, the selection of bin value affects the accuracy of the classifier. In this study, the bin value is set as {5, 10, 15, 20, 30}. The results of the ROC test with 600 monitored pages are given in Figure 13. It is seen that for bin = 5, the accumulation of response frequency simplified the distinction between websites and the FPR is 43%. For bin = 30, the website response time is finely differentiated, and the frequency span is excessively lengthened such that the website classification features are mixed, resulting in a TPR of only 80%. The area under curve (AUC) for the case with bin = 10 and bin = 20 is 0.9505 and 0.949, respectively. This is a rough indication of similar performance of the two classifiers. Through specific measurement, it is also seen that, for bin >15, the AUC of the increased bin value shrinks and the classifier has a diminishing return. Based on the results of several candidate values, the bin value is finally set as 15.

In the present study, websites servers that provide the specified hidden services are monitored. In other words, the website information provided by the servers is monitored, which could alleviate the confusion from the website internal pages. Under the background of 10,000 websites, the number of websites to be recognized increased from 50 to 1000. As it is seen in Figure 14, the FPR of classifying 1000 websites is only 1.5%. For a website size of greater than 4%, the TPR is maintained above 93%. In the face of the growing number of monitored websites, the recognition performance of the classifier is excellent as it does not decline slightly as in the case of page recognition. The index of precision emphasizes more on the overall classification of monitored websites. Monitoring 100 websites, the precision is 69.4%, with 30 monitored websites unrecognized. Therefore, the effect of WRFP classifier for monitoring small-size websites is not satisfactory. Increasing the size of monitored websites by 10%, the precision is increased from 69.4% to 86.1%. The overall classification results manifest that when the attack target is turned to the server, the classification becomes easier and the effect of monitoring large-size website servers is excellent.

A total of 40 page instances are used as the training set in the above test. According to the previous experiences, expansion of the training set can easily improve the performance of the classifier. Whether or not a small number of training sets can keep the performance of WRFP classifier within an acceptable range is investigated in this study. The background is 10,000 pages. The instances in each page is split into training sets and test sets as shown in Table 7, and ten-fold cross-validation is performed. For 10 training instances, the TPR is reduced to the lowest (90%), while the FPR is 2.5%. The recognition accuracy gaps of 20 and 30 instances with 40 instances is also acceptable (less than 1%). As expected, the classification result is affected by reducing the size of the training sets. However, WRFP classifier allows effective page recognition even for small training sets. Of course, 40 page instances should still be used for training to obtain the best WRFP classifier performance.

We also investigate robustness of WRFP classifier against time fluctuations. The response time is expanded proportionally, and the number of monitored pages to 600 and 20 rounds of tests are conducted. As it is shown in Figure 15, time error rates smaller than 5% have no real effect on the accuracy. As the error rate increased to 9%, the accuracy is gradually declined. For a time error rate of 24.5%, the recognition accuracy is dropped below 83.3%, which is the accuracy when the response time feature is removed. Such a decrease seriously affects the attack judgment. Hence, to ensure the balance between WRFP classifier accuracy and time error, time error rate needs to be kept below, and the accuracy rate is steadily higher than 90%, that is, the fluctuation of 2 bins within the response time frequency is acceptable.

6.4. Comparison in Open World. In this section, comparisons are made between WRFP classifier and the traditional website fingerprinting attack classifiers. Although the monitor target of the traditional classifiers and that of WRFP classifier are different, they are all based on fingerprinting

Table 6: Recognition effect of different pages in closed world.

Server information	Page	Accuracy	TPR	FPR
Same source	300	0.967 ± 0.023	0.947 ± 0.008	0.013 ± 0.005
	600	0.964 ± 0.041	0.944 ± 0.013	0.009 ± 0.007
	900	0.952 ± 0.025	0.936 ± 0.016	0.015 ± 0.013
Multisource	300	0.962 ± 0.011	0.936 ± 0.011	0.009 ± 0.009
	600	0.953 ± 0.016	0.935 ± 0.016	0.004 ± 0.008
	900	0.945 ± 0.021	0.930 ± 0.014	0.017 ± 0.012

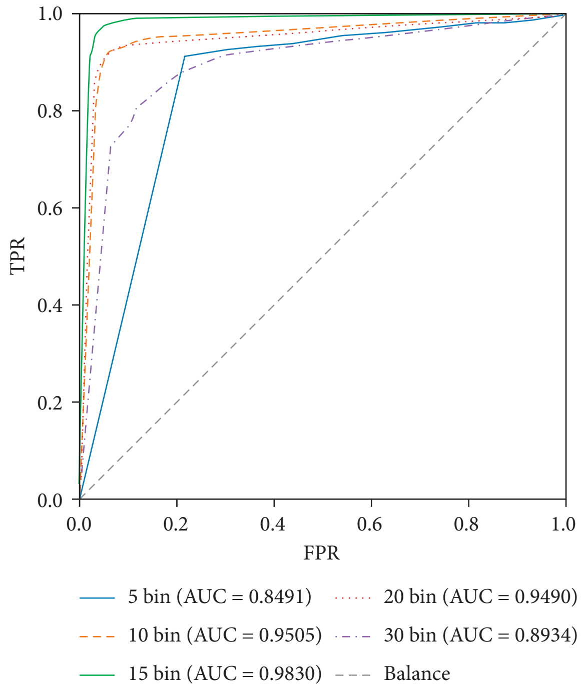


Figure 13: The effect of bin value on response time frequency.

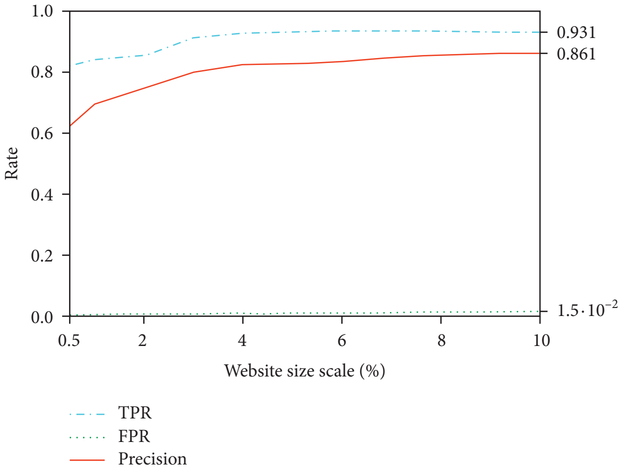


Figure 14: Classification results of monitoring website servers.

Table 7: Recognition result of the reduced training sets.

No. of training instances	No. of test instances	Accuracy	TPR	FPR
10	40	0.923 ± 0.023	0.921 ± 0.021	0.025 ± 0.015
20	30	0.957 ± 0.022	0.931 ± 0.013	0.016 ± 0.013
30	20	0.961 ± 0.035	0.934 ± 0.009	0.015 ± 0.009
40	10	0.961 ± 0.017	0.935 ± 0.009	0.008 ± 0.009

feature recognition at the traffic level and the results are comparable. The traditional models involved in the comparison are CUMUL [8], k-NN [7], and k-FP [9], and the response datasets are uniformly used for classification tests. To understand the actual effect of the classifiers, test data with unbalanced foreground and background are used to increase the challenge of classification. The background size is divided into [2000, 5000, 10,000, 50,000, 100,000, 200,000], the foreground is set as 1000 (pages) × 40 (instances), and 5 rounds of tests are carried out. In addition, the basic traffic is a response packet close to the server, and the data themselves are biased towards the response classifier. To judge the dependence between the result and the data themselves, the traditional website fingerprinting attack classifiers are divided into two states:

(i) Response time features unloaded: the basic classification function of classifiers is discussed based on the maintained original attention features of the classifier.

(ii) Response time features loaded: while keeping the original features, the response time and response time frequency features are added to test the degree of classifier judgment to page classification after loading response time.

The performance of the classifiers in open world is measured with reference to indexes of precision and recall. The classification results of original classifiers are shown in Figure 16. The performance of the classifiers is weakened by increasing background size. However, by increasing the background size from 2000 pages to 200,000 pages, the precision of WRFP classifier is kept at 86% and its recall is declined from 95.2% to 84.1%. This suggests that the WRFP classifier remains effective in recognizing monitored pages under a big background. Under a background of the same size, the precision and recall of k-NN classifier are 58.4% and 53.5%, respectively, and there are about 400 pages that are not correctly classified as monitored pages. The k-FP classifier performs well under a background of 50,000 pages, with precision of 83.8% and recall of 81.1%. However, for a background of 200,000 pages, the recall is 71.2%, and that of CUMUL is 66.4%, both of which are unable to guarantee

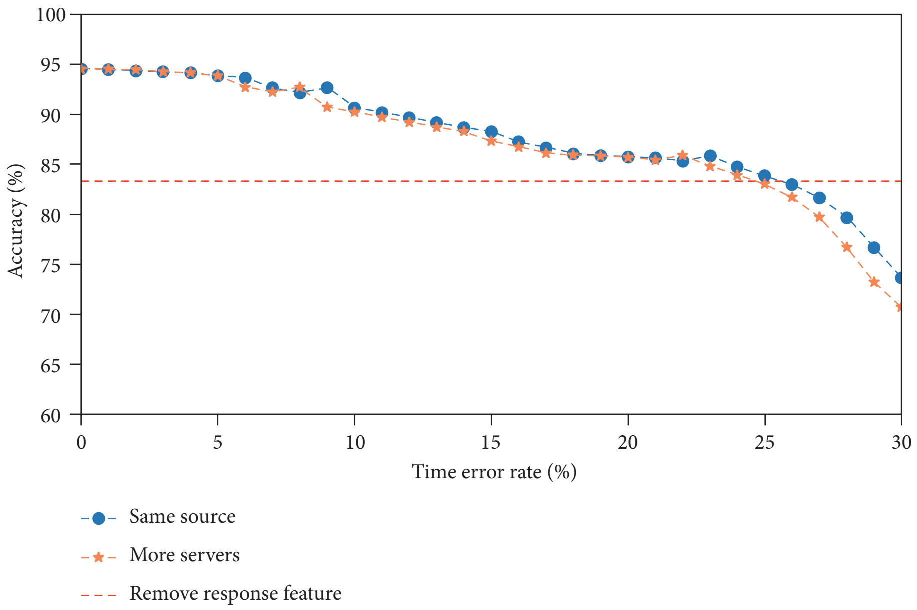


Figure 15: Response time error affected accuracy.

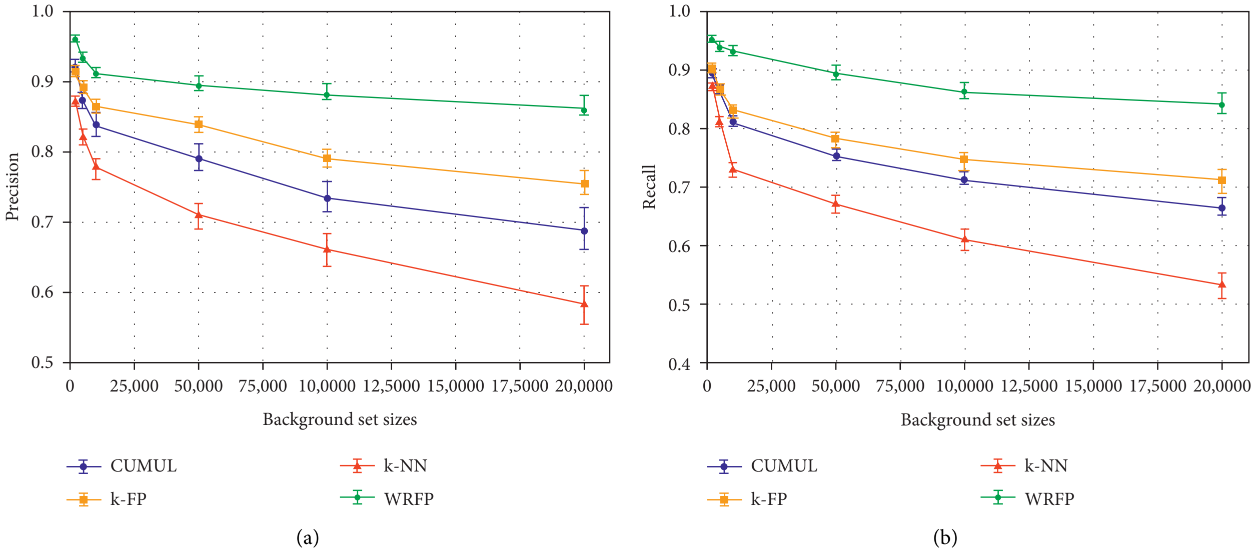


Figure 16: Comparison of test results of the original classifiers. (a) Precision. (b) Recall.

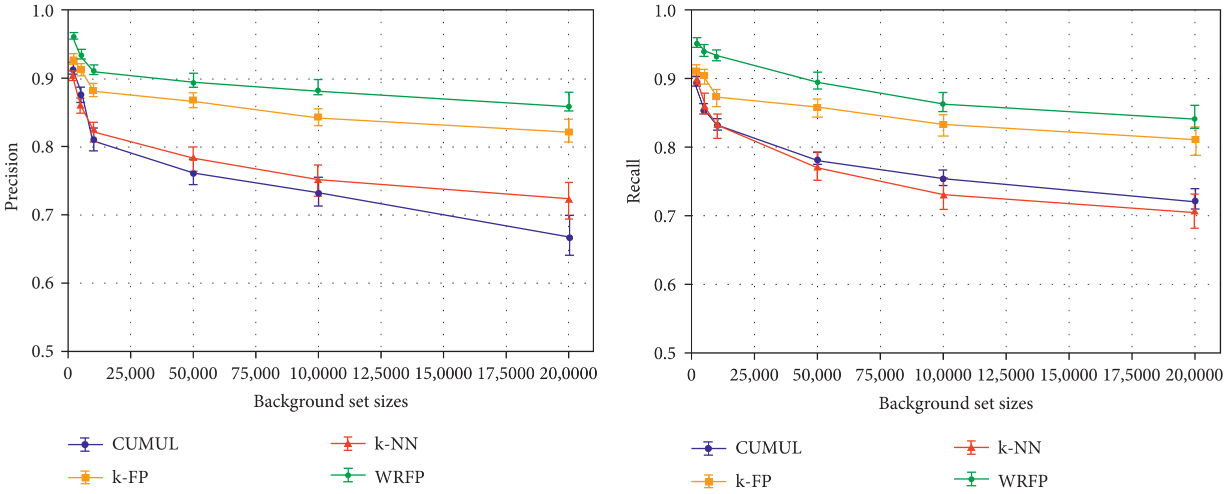
reasonable attack accuracy. The BDR results are shown in Table 8. For the background smaller than 5000 pages, all four classifiers maintain a good and correct page recognition effect. For the background of 50,000 pages, only the WRFP classifier maintained correct page recognition larger than 50%. In the face of unlimited increase of background pages, it is impossible to guarantee the classification accuracy. For example, for the background size of 200,000 pages, less than 20% of them are correctly classified.

The test results of classifiers loaded with response time features are displayed in Figure 17. In contrast with the original classifiers, the precision of k-NN classifier is increased from 58.4% to 72.4% under a background of 200,000 pages, while the recall is increased from 53.5% to 70.5%. The recognition performance of k-FP classifier is also increased slightly, with precision increasing from 75.4% to 82.1% and recall from 71.2% to 81.1%. On the contrary, the recall of CUMUL classifier is only increased by 6%, while the precision is decreased by 2%. This may be due to the fact that CUMUL is biased towards the accumulation of packets and cannot effectively utilize the loaded response time and response time frequency features, which make the recognition skew toward a worse result. It is concluded that the response time features can provide a reliable help to the classifier to judge the response fingerprint.

The results of WRFP classifier recognizing the website servers represented by the pages are shown in Figure 18. Compared with the classification result of a single page,

Table 8: BDRs of classifiers under different background sizes.

Classifier	Background sizes					
	2000	5000	10,000	50,000	100,000	200,000
WRFP	0.987	0.962	0.903	0.658	0.292	0.183
k-FP	0.984	0.944	0.863	0.456	0.219	0.093
k-NN	0.979	0.921	0.776	0.313	0.121	0.042
CUMUL	0.983	0.932	0.813	0.368	0.182	0.081



(a)

(b)

Figure 17: Test results of classifiers loaded with response time features. (a) Precision. (b) Recall.

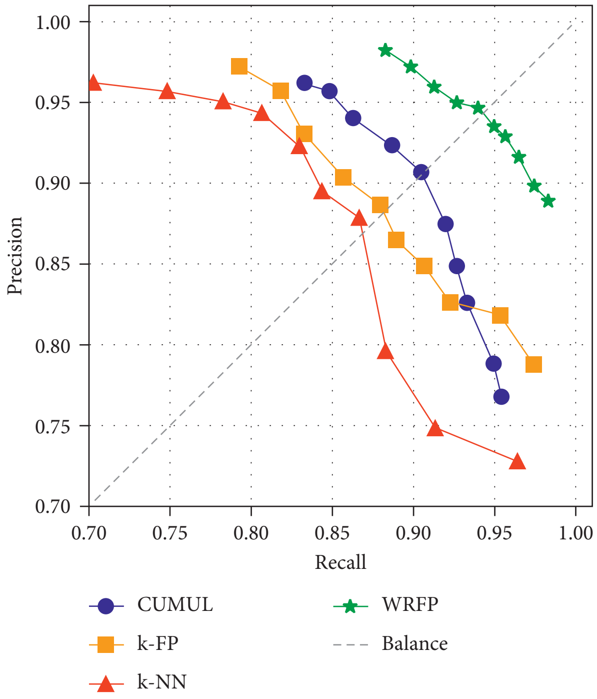


Figure 18: Precision-recall curves of recognizing website servers.

simplification of the scenarios increased the recognition probability. The test is conducted under a background of 26,000 web servers, and 1000 web servers are monitored. When an attacker looks for a balance between precision and recall, WRFP classifier can get the optimal effect of both precision (94.3%) and recall (93.7%). Although the balance value of k-FP classifier is lower than that of CUNUL, the attackers can highlight precision or recall getting the correct recognition effect. The k-NN classifier is not effective in recognizing server scenarios, and for a precision of 94.3%, the recall is only 80.6%.

In addition, the computational efficiency of the classifier is particularly important. The test results of the computational efficiency of the classifiers for classifying 10,000 pages are given in Table 4. Considering the high memory consumption by k-FP classifier, the background is set to 30,000 pages. An Intel Xeon E5-2696 v4 is used as the CPU core in the test. To obtain the optimal accuracy, the false positive pages are ignored and the corresponding optimal parameters of the classifier are selected. The test time includes the algorithm classification efficiency and the frequency of feature extraction and transformation, all of which are linked to the number of features. We test the k-NN classifier proposed by Wang [7]. The model performance is improved by optimizing the number of features (the packet length feature which is useless to the cell is deleted), and the computational efficiency is increased by 310%. However, its computing time is still longer than that of other classifiers. CUMUL classifier is the closest to WRFP classifier in terms of computing time, but its recognition accuracy is 90.7%. k-FP classifier needs a large memory in exchange for higher accuracy.

Our results suggest that WRFP classifier has the two following advantages. First, it needs smaller memory than that of the other classifiers, with a memory usage of 12.3% of that of k-FP. Second, it has a high computational efficiency which is 11 times faster than that of k-NN.

6.5. Subpages of Website Types. In the evaluation experiment in Section 6.4, the test set contained index pages and subpages of the hidden services. The results showed that separate classification of each page causes a great pressure on WRFP classifier, and too many subpages may confuse the judgment results. In reality, the subpages wrap the service content of the website, and there is a certain gap in the service content of each type of website, showing different contours at the traffic level. However, it is based the service profile of the website that the attack against the server determines the server to which the page belongs. Here, the results of WRFP classifier recognizing the subpages of website types of news, porn, forum, and social sites are investigated, and the corresponding recognition strategies are analyzed.

Firstly, the recognition scenario is set to divide the subpages within the website type, and evaluation was made on a small scale to get the most direct recognition effect. For each website type, 8 independent domain name websites each containing 30 different subpage instances are randomly selected. Figure 19 displays the subpage recognition heat map of the website types. The accuracy of the 32 websites is 86%. Among them, the first 8 are subpages of the forum sites, with an accuracy of 90.8%, which is higher than that of the news sites (82.1%). The accuracy of porn sites is also above the mean value. In addition, there exists an interesting observation when observing the incorrectly classified pages. Among 22 confusing page instances in the forum sites, 19 are recognized as forum sites, i.e., there is a high probability that the false positive pages are subpages of the forum sites. Page recognition is conducted in a small environment, and the test pages are independent of each other. It is also seen through specific analysis that the service patterns of forum sites are similar, and these similar patterns improved the possibility of recognition. Therefore, it can be assumed that the website type affects the classifier's judgment of the page.

To investigate the influence of website types on the recognition, the classification of website subpages is focused upward, and the types provided by the server are classified. Based on these, the subpages are recognized for the second time. The original dataset is used in the test, and ten-fold cross-validation is also performed. The classification data of the four types of focused websites are given in Table 9. The classification accuracy of the forum sites is improved to 98.7%, with a TPR of 97.6% and an FPR of 0.04%, and there are only 3 false positive pages. The news sites experience the largest increase, with an accuracy of 93.5%. Most of the main service contents in porn websites are pictures, which makes traffic fingerprinting special and different from other types of websites. The results indicate that, in monitoring servers providing specific website types, fixed-point training with more pages of relevant types makes up for the shortcoming of false positive page recognition in terms of attack strategies.

6.6. Client Traffic and Defense Confrontation. Different versions of Tor Browser Bundle (TBB) can be run at the client. In this section, we investigate whether or not the fluctuation of server response fingerprinting caused by different TBB versions can affect the classification efficiency. There are small differences between different versions of TBB. During the communication process of TBB v6.5 (core Tor 0.2.9), the client is equipped with multiple ingress nodes. TBB v7.5 (Tor 0.3.2) provides third-generation service response request. TBB v8.5 (Tor 0.4.0) supports traffic adaptive filling defense mechanism. (2000 (websites) × 20 (instances)) × 3 groups of website instances are used for evaluation, and 10,000 pages samples are used as the background. Considering that TBB v6.5 can only access v2 websites, v3 addresses are removed from the dataset. The defense mechanism is not added to TBB v8.5 at this stage. We consider separate training sets for each version; the other two versions are used as the test sets. We fix the basic properties of TBB, UseEntryGuards is set to disabled state, and new Guard nodes are enabled for each link communication to ensure the freshness of the link traffic.

The response recognition results generated under the 3 TBB versions are presented in Figure 20. Surprisingly, the TPR value fluctuates between 91% and 96%, and the peak of FPR is 1.6%, suggesting that the version difference is not reflected in the response fingerprint, and WRFP classifier can simply ignore the version change of Tor Browser. TBB v8.5 has the best classification effect, and the attacker can use the current highest version as a training set by default (version 9.0 or above has been released when this paper is published, but the core is still based on Tor 0.4), so there the TBB version has no impact of the previously analyses.

It can be seen from the above conclusion that the TBB change does not affect server recognition. Therefore, the test is extended to evaluate the impact of the client defense model on WRFP classifier. Comparison with the classical fingerprinting defense of CS-BuFLO [10], Tamaraw [11], WTF-PAD [12], and ALPaca [13] is helpful to understand the actual impact of the existing defense technology in resisting the server response fingerprinting attack. In this test, we process 2000 × 50 instances by each defense to obtain a defense fingerprinting set with its own tags. The defense test results of the classifier are presented in Table 10, which includes the overhead resources for statistical processing.

It can be seen in Table 10 that CS-BuFLO and Tamaraw control the packet sending rate and disrupt the response time reception rhythm. These defense models consume more than 140% of the bandwidth and double the data delay, resulting in a TPR of only 4% and an FPR of 70% for WRFP classifier. For WTF-PAD, the classifier has a TPR of nearly 80% and an FRP of 7%, which is better than the actual effect of lightweight defense. It is also observed that WTF-PAD is unable to break the division of bin in response time frequency by padding the traffic gap. It is worth noting that, even when recognizing the ALPaca fingerprinting data specially provided for server defense, the recognition accuracy of TPR is 56.5% under same-source

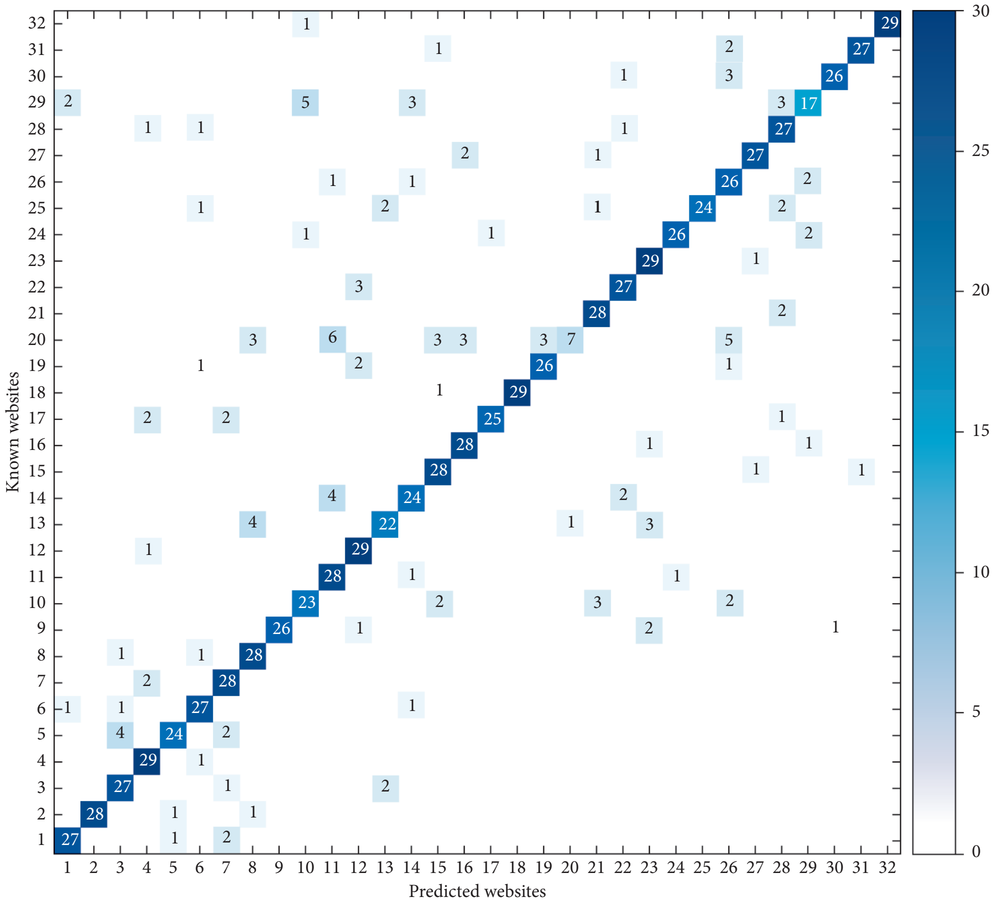


Figure 19: Subpage classification heat map of four types of websites.

Table 9: Classification results of subpages of focused website service types.

Website type	Accuracy	TPR	FPR
Forum	0.987 ± 0.005	0.976 ± 0.007	0.004 ± 0.005
News	0.935 ± 0.024	0.921 ± 0.023	0.013 ± 0.008
Social	0.948 ± 0.014	0.935 ± 0.016	0.021 ± 0.013
Porn	0.912 ± 0.016	0.905 ± 0.013	0.016 ± 0.004

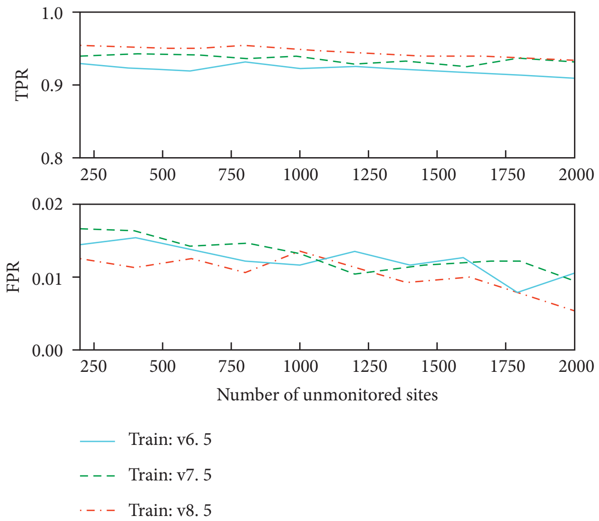


Figure 20: Fluctuation test of TBB versions to response fingerprinting attack.

Table 10: Comparison of test results among defense models.

Defense	Overhead		Same server		Multiserver	
	Bandwidth (%)	Delay (%)	TPR (%)	FPR (%)	TPR (%)	FPR (%)
CS-BuFLO	149	116	4.4	75.8	7.4	72.3
Tamaraw	144	102	4.1	69.8	6.6	66.7
WTF-PAD	22	17	79.5	7.8	81.3	8.3
ALPaca	56	45	56.5	23.3	59.3	19.6

server while consuming 56% of the bandwidth and 45% of the delay. Interestingly, the recognition effect of multisource server is always better than that of the same-source server. This indicates that multisource response fingerprinting is less affected by defense. Our results suggest that the defense model of traffic padding can compete with classifiers, but at the expense of a large bandwidth and an extended delay. In the face of lightweight defense model, it is promising that WRFP classifier is capable of maintaining recognition results with a high precision.

7. Conclusion and Future Works

In this paper, a WRFP attack technique based on response time features was proposed. A hidden service response fingerprinting dataset was constructed, and the basic performance of WRFP classifier was tested based on the extremely randomized tree and the response time measurement standard. The experimental evaluation revealed that, in closed world, both same-source server and multisource server achieve a better accuracy in traffic recognition, and even if the training set is reduced by half, the original accuracy will not be reduced. In the open world with a large size gap between the foreground and background, it was shown that the response fingerprinting classifier is more efficient in terms of accuracy and computational efficiency compared to the previous fingerprinting classifiers based on traditional manual features. In addition, the disturbance caused by factors of TBB versions and fingerprinting defense was considered and analyzed, and the stability and effectiveness of the classifier were confirmed. The test results of subpages showed that WRFP classifier is able to effectively focus on the classification of different website types, and with the increase of subpages, its recognition effect of subpages will not lag behind that of the index pages.

The traffic fingerprinting recognition in response to the hidden service which was proposed in this paper is different from the conventional website fingerprinting attack scenario, thus introducing new challenges in traffic analysis and attack standards. In the future, we will carry out in-depth research on website servers in different geographical locations and take steps to integrate a deep learning algorithm to improve the performance of the classifier in presence of extra noise interference.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China (Nos. 2019QY1302 and 2019QY1305).

References

[1] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: the second-generation onion router,” in *Proceedings of the 13th USENIX Security Symposium*, pp. 303–320, San Diego, CA, USA, August 2004.

[2] B. Zantout and R. Haraty, “I2P data communication system,” in *Proceedings of the 10th International Conference on Network*, pp. 401–409, Valencia, Spain, May 2011.

[3] J. A. Elices and F. Pérez-González, “Locating Tor hidden services through an interval-based traffic-correlation attack,” in *Proceedings of the IEEE Conference on Communications and Network Security*, pp. 385-386, Washington, DC, USA, October 2013.

[4] A. Biryukov, I. Pustogarov, and R. P. Weinmann, “Trawling for tor hidden services: detection, measurement, deanonymization,” in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, pp. 80–94, Westin St. Francis, San Francisco, CA, USA, May 2013.

[5] Z. Ling, J. Luo, K. Wu, and X. Fu, “Protocol-level hidden server discovery,” in *Proceedings of the 2013 IEEE INFOCOM*, pp. 1043–1051, Turin, Italy, April 2013.

[6] S. Matic, P. Kotzias, and J. Caballero, “CARONTE: Detecting location leaks for deanonymizing tor hidden services,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1455–1466, Denver, CO, USA, October 2015.

[7] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, “Effective attacks and provable defenses for website fingerprinting,” in *Proceedings of the 23rd USENIX Conference on Security Symposium*, pp. 143–157, San Diego, CA, USA, August 2014.

[8] A. Panchenko, F. Lanze, A. Zinnen et al., “Website fingerprinting at internet scale,” in *Proceedings of the 23th Annual Network and Distributed System Security Symposium*, pp. 1–15, San Diego, CA, USA, February 2016.

[9] J. Hayes and G. Danezis, “K-fingerprinting: a robust scalable website fingerprinting technique,” in *Proceedings of the 25th USENIX Security Symposium*, pp. 1187–1203, Austin, TX, USA, August 2015.

[10] X. Cai, R. Nithyanand, and R. Johnson, “CS-BuFLO: a congestion sensitive website fingerprinting defense,” in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pp. 121–130, Scottsdale, AZ, USA, November 2014.

[11] X. Cai, R. Nithyanand, T. Wang, R. Johnson, and I. Goldberg, "A systematic approach to developing and evaluating website fingerprinting defenses," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 227–238, Scottsdale, AZ, USA, November 2014.

[12] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an efficient website fingerprinting defense," *Computer Security*, pp. 27–46, Heraklion, Greece, September 2016.

[13] G. Cherubin, J. Hayes, and M. Juarez, "Website fingerprinting defenses at the application layer," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 2, pp. 186–203, 2017.

[14] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A critical evaluation of website fingerprinting attacks," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 263–274, New York, NY, USA, November 2014.

[15] T. Wang and I. Goldberg, "On realistically attacking tor with website fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 21–36, 2016.

[16] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial Naïve-Bayes classifier," in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, pp. 31–42, Chicago, IL, USA, November 2009.

[17] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a distance: website fingerprinting attacks and defenses," *y*, in *Proceedings of the 2012 ACM Conference on Computer and Communications Securit*, pp. 605–616, Raleigh, NC, USA, October 2012.

[18] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, pp. 253–262, Porto, Portugal, February 2017.

[19] A. Kota and G. Shigeki, "Fingerprinting attack on tor anonymity using deep learning," *Proceedings of the Asia-Pacific Advanced Network*, pp. 15–20, 2016.

[20] V. Rimmer, D. Preuveneers, M. Juárez, T. V. Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," in *Proceedings of the 25th Annual Network and Distributed System Security Symposium*, 2017, https://arxiv.org/abs/1708.06376v2.

[21] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep fingerprinting: undermining website fingerprinting defenses with deep learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1928–1943, Toronto, Canada, October 2018.

[22] S. Bhat, D. Lu, A. Kwon, and S. Devadas, "Var-CNN: a data-efficient website fingerprinting attack based on deep learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 4, pp. 292–310, 2019.

[23] T. Pulls and R. Dahlberg, "Website fingerprinting with website oracles," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 1, pp. 235–255, 2020.

[24] S. Zander and S. J. Murdoch, "An improved clockskew measurement technique for revealing hidden services," in *Proceedings of the 17th Conference on Security Symposium*, pp. 211–226, Tallinn, Estonia, November 2008.

[25] Q. Tan, Y. Gao, J. Shi, X. Wang, and B. Fang, "A closer look at Eclipse attacks against Tor hidden services," in *Proceedings of the IEEE International Conference on Communications*, pp. 1–6, 2017.

[26] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, and S. Devadas, "Circuit fingerprinting attacks: passive deanonymization of tor hidden services," in *Proceedings of the 24th USENIX Conference on Security Symposium*, pp. 287–302, Berkeley, CA, USA, August 2015.

[27] J. Marques, L. Velasco, and R. V. Duijn, *Tor: Hidden Service Intelligence Extraction*, https://www.delaat.net/rp/2017-2018/p98/report.pdf, 2018.

[28] R. Louis-Paul, "Statistical properties of Winsorized means for skewed distributions," *Biometrika*, vol. 81, no. 2, pp. 373–383, 1994.

[29] J. Yan and J. Kaur, "Feature selection for website fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 4, pp. 200–219, 2018.

[30] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pp. 431–439, Lake Tahoe, NV, USA, December 2013.

[31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8891183, 13 pages
https://doi.org/10.1155/2020/8891183

Research Article

PyRos: A State Channel-Based Access Control System for a Public Blockchain Network

Siwan Noh[1], **Sang Uk Shin**[2], and **Kyung-Hyune Rhee**[2]

[1]*Department of Information Security, Graduate School, Pukyong National University, Busan 48513, Republic of Korea*
[2]*Department of IT Convergence and Application Engineering, Pukyong National University, Busan 48513, Republic of Korea*

Correspondence should be addressed to Kyung-Hyune Rhee; khrhee@pknu.ac.kr

Received 7 August 2020; Revised 17 October 2020; Accepted 16 November 2020; Published 27 November 2020

Academic Editor: Zhe-Li Liu

Blockchain is a technology that enables the implementation of a decentralized system by replacing the role of the centralized entity with the consensus of participants in the system to solve the problem of subordination to the centralized entity. Blockchain technology is being considered for application in numerous fields; however, the scalability limitation of a public blockchain has led many researchers to consider private blockchains, which reduce the security of the system while improving scalability. A state channel represents a leading approach among several scalability solutions, intended to address public blockchain scalability challenges while ensuring the security of the blockchain network. Participants in the channel perform the process of updating the state of the channel outside the blockchain. This process can proceed very quickly because it does not require the consensus of the blockchain network, but still, like on-chain, it can guarantee features such as irreversibility. In this paper, we propose the PyRos protocol, an access control system that supports the trading and sharing of data between individuals on a public blockchain based on the state channel. As far as we know, the research using the off-chain state channel for access control has not been proposed yet, so PyRos is a new approach in this field. In PyRos, user-defined access control policies are stored off-chain, and policy updates are always rapid regardless of the performance of the blockchain network. Moreover, PyRos provides means to prevent malicious participants from arbitrarily using the channel's previous state while resolving constraints due to scalability problems, along with privacy guarantees for the transaction content. To evaluate the efficiency and security of PyRos, we provide qualitative analysis of security requirements and analysis in terms of the performance of public blockchain platforms.

1. Introduction

The development of Internet of Things (IoT) technology has enabled us to generate unimaginable quantities of data in the course of our daily lives. A variety of data is produced with smart mobile devices such as smartphones, smart bands, or devices connected to smart home networks (TVs, lights, etc.). According to a recent survey [1], the data generated in this manner is expected to reach 175 zettabytes per year by 2025. Big data is a technology that analyzes such large quantities of data to extract new information. It is used in numerous fields, including healthcare and logistics [2, 3]. However, this requires the collection of extensive user data. Machine data [4] refers to data collected through machines, such as industrial equipment, sensors, or weblogs that record users' behavior on the Web. The amount of data acquired by the dissemination of IoT devices is expected to increase exponentially.

In a traditional IT platform environment, users do not have the proper authority over their data. Global IT companies, such as Google and Facebook, or service providers have taken control of the users' data, which has caused numerous security concerns [5, 6]. The MyData industry [7] presents a paradigm in which the subject of information manages, controls, and utilizes their data based on the right to data portability of individuals instead of companies or governments. In the MyData industry, blockchain is considered as a key technology for decentralized data self-control, and numerous related projects are being proposed [8, 9].

The blockchain node collects data through peer-to-peer networks and stores it in a chain-structured distributed data storage. Characteristically, based on the consensus protocol, it is possible to implement a reliable operation among nontrust nodes without a central authority whom the nodes commonly trust. The blockchain network uses a variety of consensus protocols to solve problems arising from the absence of a central authority. Only data verified through consensus protocols is stored as new data in the blockchain.

Blockchain-based access control [10–12] is one of several blockchain-based applications. Instead of being managed by the centralized access control server for storage and control of resources, access control policies are kept and verified in the blockchain layer built above the storage layer. However, in practical terms, resources are stored outside the blockchain (e.g., cloud storage) and only access control policies are kept in the blockchain, as storing the data itself in the blockchain causes an unaffordable overhead for users on the network. The decentralization, transparency, and irreversibility of the blockchain are expected to enable the delivery of new services by overcoming the limitations of the traditional access control system.

A public blockchain, however, has the disadvantage of the absence of a system administrator, which limits the processing performance of the system. To ensure reliable operation in a blockchain network composed of only untrusted nodes, Bitcoin blockchain employed a very strong consensus protocol called Proof-of-Works (PoW); however, this resulted in only about seven transactions per second. Numerous blockchain projects have recently solved this problem by limiting the nodes of the blockchain network to authorized users (permissioned blockchain) or organizing only specific groups of users (private blockchain) [13]. This approach remains a problem that is being discussed today, as it abandons decentralization to improve scalability [14]. Scalability, decentralization, and security are called blockchain trilemma as factors that are difficult to satisfy simultaneously on the blockchain. Recently, many solutions have adopted a method that has been recentralized and security-vulnerable to improve efficiency. However, in this paper, we do not consider this approach. Because the motivation behind blockchain-based access control is to eliminate the access control server and implement user-centric access control, it is not desirable to apply recentralization solutions to access control applications. However, if access control applications are implemented on the public blockchain, the limited processing performance of blockchain networks makes it difficult for user-defined policies to be reflected without delay.

To overcome the above-mentioned problem, in this study, we propose the PyRos protocol based on the off-chain state channel, one of the blockchain scalability solutions.

To summarize, our contributions are listed as follows:

(i) We propose the PyRos protocol, an access control application that operates on a public blockchain with limited processing performance.

(ii) PyRos operates based on the off-chain state channel solution and provides a validation method for access control policies recorded on the off-chain channel.

(ii) PyRos does not sacrifice the security or decentralization of the system that operates to improve scalability.

2. Background

We present an overview and related research on blockchain-based access control, blockchain scalability limitations, and the off-chain state channel.

2.1. Blockchain and Blockchain-Based Access Control. Bitcoin [15], the most widely known cryptocurrency, records information on its ownership in the blockchain ledger. Users update the ownership information of the Bitcoin recorded in the blockchain ledger through the creation of transactions, including their digital signatures and new owner information (e.g., blockchain address), to use the Bitcoin they own. If the information contained in the transaction is valid, it will be disseminated to the majority of users of the Bitcoin blockchain network, and it will later be included in the block through mining and reflected in the blockchain ledger. The Bitcoin blockchain selects miners at certain time intervals based on the PoW algorithm to maintain a single blockchain ledger on the network. The PoW algorithm makes only single ledger exist in the network, even if several miners attempt to update their blockchain ledger at the same time. The PoW algorithm adds blocks of users, who first find values that make the cryptographic hash results of the block header exist within a certain range to the blockchain as a new block. Finally, the blockchain takes the form of a hash chain, which ensures the irreversibility and transparency of the blockchain.

The transparency and irreversibility of the blockchain can have a huge impact on improving the reliability of the database management. In particular, applying the blockchain to the access control system makes it possible to manage policies for the requester without a centralized authority. The key element of blockchain-based access control is similar to cryptocurrency. In cryptocurrency blockchain, users manage their cryptocurrency without the help of banks. The blockchain is a ledger that records cryptocurrency ownership information for all users of the network and that has been recording all details since the launch of the cryptocurrency. In contrast, blockchain-based access control records access control policies for digital objects in the blockchain ledger instead of recording ownership information for the cryptocurrency.

In [10, 11], each transaction represents the subject's right to access the object. The rights recorded in the blockchain can be transferred to another user without the help of the owner, and any user can inspect who has the rights at any time through the blockchain. However, it is not desirable for the buyer to resell the seller's data in the data trading model. In [12], Xia et al. proposed blockchain-based data sharing for electronic medical records stored in the cloud. Verifiers can confirm the membership of a user by using cryptographic keys that are generated by the issuer before storing the request to the blockchain. Therefore, all users can efficiently manage their data without the help of a third party. However, the authors do not consider the users' privacy and the limited throughput of the public blockchain.

In the blockchain-based access control system, users' access control policies are open to all participants in the network. This transparency of the blockchain ensures transparent management of the Access Control List (ACL); however, at the same time, it has the disadvantage of making user-defined policies public to all participants in the network. Because the system is affected by security problems within in the blockchain, the relationship between the access control system and the blockchain security concerns must also be considered.

2.2. Blockchain Scalability and State Channel. Blockchain ensures transparency and irreversibility of systems, which have been difficult to achieve for centralized systems. Therefore, many industries are considering converting their operating systems into the blockchain. However, research on the blockchain technology has gradually highlighted unique problems of the blockchain [16, 17], and their evaluation before switching the system to the blockchain is becoming important [18, 19]. Scalability is one of the most representative problems, which means that the speed of transaction processing in the network does not increase even when more resources are put into the blockchain network. This is because the block creation cycle and size are limited for a stable consensus in the blockchain network. Several cryptocurrency developers have attempted to improve transaction throughput in the blockchain network by reducing or eliminating this restriction. However, Croman [20] showed that increasing the block size or decreasing the block generation cycle in the blockchain P2P communication protocol increased the propagation delay in the network [21] and consequently reduced the security of the blockchain network. The blockchain trilemma is the biggest challenge in the blockchain industry due to the difficulty of satisfying all three factors, security, decentralization, and scalability, in the blockchain system. Numerous attempts have been proposed to improve the performance of the blockchain and challenge the trilemma. Currently, there is a private blockchain that is widely used. The private blockchain limits network participants to authorized users and reduces the level of consensus to network administrators to ensure scalability by sacrificing decentralization, thereby failing to solve the trilemma. Hence, the segregated witness (Segwit) [22] of Bitcoin, the sharding and Casper algorithm of Ethereum, and an Algorand's Pure Proof-of-Stake (PPoS) protocol [23] have been proposed as solutions to avoid the blockchain trilemma. Another proposed solution is the state channel, which is the focus in this study.

A state channel has been employed in numerous studies [24, 25] as a solution to solve the problem of scalability due to the finality of the blockchain by introducing off-chain processing methods. Finality guarantees that the block will not change after it is added to the blockchain, which means that the blockchain transaction is irreversible. However, in the public blockchain network, there is a possibility that blocks already added to the blockchain will branch out (i.e., fork) and be discarded due to competitive block generation algorithms. When a fork occurs, groups in the network arise which have two or more different blockchains. After the subsequent block generation process, groups that lost the competition discard their blockchain and replace it with the blocks of the group that won the competition instead. In the process, the transactions in the discarded blockchain are likewise canceled and later included in the block again. Thus, the public blockchain cannot guarantee an absolute finality and only provide a probabilistic one [26]. In contrast, a private blockchain can provide absolute finality by applying noncompetitive consensus algorithms, such as Practical Byzantine Fault Tolerance (PBFT). The probabilistic finality of the public blockchain also affects the transaction throughput of the blockchain network. A private blockchain can achieve higher throughput compared to the public blockchain due to absolute finality. However, in the public blockchain, a certain amount of confirmation time is required after the block is included in the chain to ensure that the block is stochastically safe enough (Bitcoin requires an average of approximately 60 minutes, and Ethereum requires approximately 6 minutes for confirmation). As shown in Table 1, the probability that an attacker can invalidate blocks that have already been confirmed increases with the hash rate that the attacker has in the entire network. However, as the number of confirmed blocks increases, the probability of a successful attack decreases, meaning that the block is highly unlikely to be modified in the presence of sufficient confirmed blocks. Consequently, probabilistic finality makes it difficult to apply the public blockchain to systems that require rapid processing in real time.

A state channel can solve the blockchain trilemma and significantly improve transaction throughput by processing transactions between users on off-chain channels and recording only the results on the blockchain. The transaction processing in the state channel is conducted outside of the blockchain (called the off-chain), such that fast transaction throughput can be guaranteed regardless of the probabilistic finality of the public blockchain. Further, state channels have two advantages: First, transaction processing takes place outside the blockchain, such that transaction processing fees are not required, because the blockchain network does not consume resources. Second, when continuous transactions occur among users, privacy protection may be provided by recording only the first state and final state of transactions in the blockchain instead of all of them, as shown in Figure 1.

The state channel is valid from the time when the initial state of the channel, which all channel participants agreed to, is recorded on the blockchain until one of the channels' various states, which was exchanged on the off-chain, is propagated to the blockchain network by one of the

Table 1: Probability of success of a double-spending attack based on the attacker's hash rate (y-axis) and the number of confirmations (x-axis).

q (%)	1 (%)	2 (%)	3 (%)	4 (%)	5 (%)	6 (%)	7 (%)
2	4	0.237	0.016	0.001	0	0	0
10	20	5.6	1.712	0.546	0.178	0.059	0.02
20	40	20.8	11.584	6.669	3.916	2.331	1.401
30	60	43.2	32.616	25.207	19.762	15.645	12.475
40	80	70.4	63.488	57.958	53.314	49.3	45.769
50	100	100	100	100	100	100	100

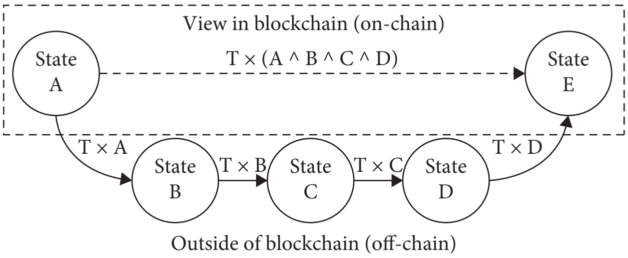


Figure 1: State channel overview.

participants. However, all off-chain states that are generated on the channel are valid states that allow participants in the channel to propagate them to the blockchain network at any time. Thus, participants may propagate the previously agreed past state to the blockchain network as the final state, without the consent of the counterparty and for their own benefit, instead of propagating the final state agreed between the participants.

For example, Alice has a contract to pay Bob a dollar a day for a month. Instead of creating a daily transaction for Bob, Alice can use a state channel to change the balance state of the two participants every day. The initial state of the channel with Alice's balance of \$30 and Bob's balance of \$0 (state_1) will be recorded in the blockchain by Alice. Alice generates a state change transaction $\text{tx}_{1\longrightarrow \text{day}}$ every day which reduces her balance by one dollar and increases Bob's balance by one dollar. Every day, Bob generates and delivers his digital signature to Alice in agreement with the state change transaction that Alice generates. After a month, the off-chain balance state will be \$0 for Alice and \$30 for Bob (state_{30}). Alice or Bob can propagate the last generated state change transaction $\text{tx}_{1\longrightarrow 30}$ to the blockchain network to record it as the final state of the channel and close the channel.

However, on the last day of the contract, Alice could propagate the state change transaction $\text{tx}_{1\longrightarrow 2}$ she created on the first day to the network instead of the last transaction $\text{tx}_{1\longrightarrow 30}$ to avoid paying. As a result, there is no normal transition of state ($\text{state}_1 \longrightarrow \text{state}_{30}$), and only the partial transition of state ($\text{state}_1 \longrightarrow \text{state}_2$) occurs in the blockchain, and the channel is closed.

To prevent the above problem, the use of previous states, except for the most recently agreed state, must be prevented. Decker et al. proposed a method to add a time-lock to the off-chain state, such that it cannot be included in the blockchain until a certain amount of time has passed, even if the previously agreed state is propagated to the network [21]. When generating an off-chain state, participants add a time-lock shorter than the time-lock included in the previous state, such that the most recently agreed off-chain state can be added to the blockchain at any time. However, the interval of the time-lock set on the channel gave rise to the expiration time for the channel to operate. Poon and Dryja proposed a replace-by-revocation [24] that implicitly revokes the previous state and agrees on a new state. In [24], when updating the state of the channel, participants create and exchange transactions that discard the previous state. If one of the participants propagates the channel's previous state (revoked state) to a blockchain network without the counterparty's consent, the counterparty can propagate the previously exchanged revoked transaction to the network within a particular time and eventually consume all deposits that were locked in the channel as a penalty.

3. PyRos System

We propose PyRos, a system that improves the problem of scalability of public blockchain applications. The PyRos system is composed of three layers, as shown in Figure 2.

(i) The **Data Owner (DO)** stores data they want to share with others into the cloud storage. To avoid data exposure by unauthorized users, they must encrypt their data before storing it. DO establishes a state channel to give other users access to these data and manages access to the data based on the off-chain channel's state transition.

(ii) The **Data Requester (DQ)** wants to access DO's data stored in the cloud storage. After obtaining appropriate access rights to DO's data through them, DQ requests the storage keeper to access these data.

(iii) The **Storage Keeper** keeps the stored data securely and provides the requested data only to users with the appropriate permissions. When DQ submits the off-chain state for access to the stored data with the corresponding evidence, they verify the validity of the submitted state and evidence based on the information recorded on-chain.

The first layer is an *application layer*, where the owner of the data and the user requesting access to the data create a state channel to correctly manage data access. In PyRos, the state of the off-chain channel represents the access rights of the channel participants to specific data held by the data owner. The second layer is a *blockchain layer* that records the state of off-chain channels created in the application layer on the blockchain and uses it to validate access authority at the storage layer. The third layer is a *storage layer* that stores data that users want to produce and share with other users. Access to the storage layer is controlled by the storage keeper, who will only provide the requested data to users with appropriate permissions.

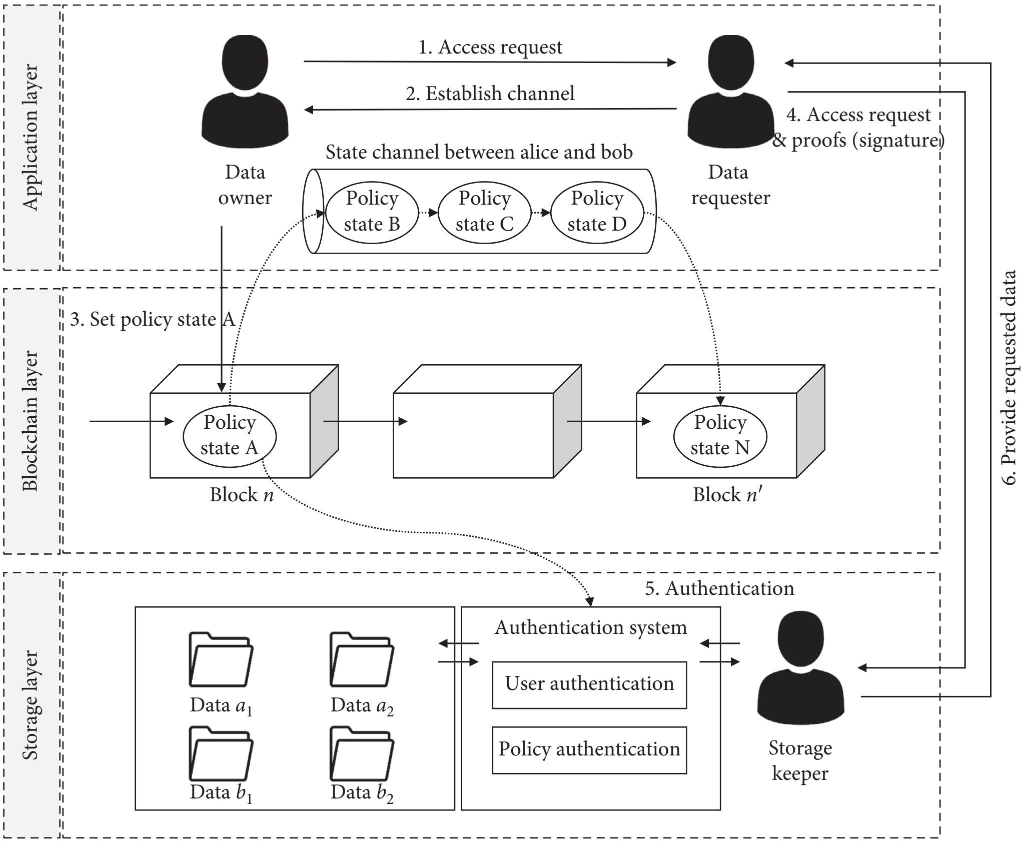


Figure 2: Proposed system architecture.

In the remainder of this section, we present the role of participants in the proposed system and the goals that the system seeks to achieve.

3.1. Overview of PyRos. We employ the system architecture shown in Figure 2 to design an access control system over public blockchain networks. PyRos comprises a data owner, data requester, and storage keeper as system participants, and each participant's role in the system is as follows:

The proposed PyRos system consists of *setup*, *channel management*, and *close channel* phases. In the setup phase, the data owner and the data requester create an off-chain channel to control access to the data stored in the external storage by the data owner. In the *channel management* phase, the data owner creates a transaction that changes the state of the off-chain channel created in the setup phase, and the two participants store it individually. In PyRos, the state of the off-chain channel represents the access control policy for specific data stored in the storage layer. The storage keeper validates the data requester's request based on transactions that transform the initial state of the channel stored on-chain into the proper access control policy. Each time a channel's state is changed, the data owner executes an implicit revocation that prevents the data requester from using the channel's previous state in the request. Finally, the *close channel* phase deals with closing the off-chain channel when access control is no longer required between the data owner and the data requester. To perform data access control in PyRos, users create three transactions as follows:

(i) Funding transaction (T_{State_0}): As the first transaction to create an off-chain channel, both users (i.e., DO and DQ) create a funding transaction to deposit their cryptocurrency on the channel. The funding transaction consists of two types of transactions in which users' deposits are transferred to a 2-of-2 multisignature address (the initial state) and in which the channel's deposits are returned to their original owners after a certain time t_{settle} (the refund transaction).

(ii) State transaction (T_{State_n}): By creating state transactions, participants in the channel can change the state of the channel (i.e., redistribution of deposits recorded in the initial state State_0) until the refund transaction recorded on the on-chain is included in the blockchain after t_{settle}. A valid state transaction contains the digital signatures of all participants in the channel and the blockchain addresses of the data owner and requester, such as the standard transaction structure of cryptocurrency (e.g., Bitcoin, Ethereum). However, unlike the standard transaction structure, the signature in the state transaction contains a hash value of the data that users want to share as a message of the signature. Therefore, even if state transactions are propagated over a blockchain network without the consent of the other party, the propagated transaction cannot be included in the blockchain (because it has an invalid structure). After obtaining the valid state transaction, the data requester generates and submits their digital signature to the corresponding storage keeper with the storage transaction generated on their off-chain channel. When the data owner wants to modify the access control policy, they create a new state transaction that changes the hash value contained in the signature to the hash value of the new data without having to establish a new channel.

(iii) Revoked state transaction ($\mathrm{RT}_{\mathrm{State}_n}$): Unlike the state transaction, signatures in the revoked state transaction do not contain a hash value of shared data. Consequently, the revoked state transaction can be propagated to the blockchain network and be included in the blockchain, which is used to prevent the data requester from using the channel's out-of-date status in access requests.

4. Security Goals

We consider two threat models for the proposed system, and to design a more realistic and practical system, we adopt several assumptions. First, we assume that a platform exists for the matching of data owners and data requesters. Our proposed system focuses on the sharing of data stored in external repositories which takes place between two users after this matching. Second, we assume that the storage keeper is a trusted entity. The storage keeper honestly verifies the request of the data requester and provides the requested data only to the requester who has presented the valid permissions. Because the focus in this study is the proposal of a decentralized approach control method, centralization of the storage layer is assumed. Finally, we assume that users participating in our system have generated a parent private key/public key pair, with child private key/public key pairs derived from it, and that corresponding addresses are generated from their child public key using BIP 0032 HD Wallets [27]. These child key pairs and addresses are denoted as $\mathrm{Kx} = \{\mathrm{sk}_{x,1}, \mathrm{pk}_{x,1}, \mathrm{addr}_{x,1}, \ldots, \mathrm{sk}_{x,l}, \mathrm{pk}_{x,l}, \mathrm{addr}_{x,l}\}$, where x denotes the user's identity and l is the number of indexes.

(i) *Threats within a channel*: Within the established channel, a malicious data requester can attempt to access unauthorized data by modifying the permissions they have been granted from the data owner or by using states that were revoked by the data owner in the past.

(ii) *Threats outside a channel*: Adversaries outside the channel can inspect a blockchain ledger and extract the information needed for an attack from the public information. In the case of an active attacker, an attack on a blockchain network [16, 17] could pose a threat to the security of not only the proposed system but also of all systems operating on the target blockchain network.

Under the threat model noted above, we consider the following security goals for a decentralized access control system on the public blockchain.

(i) *State privacy*: Third parties in the public blockchain network (except the data owner, data requester, and storage keeper) must not know details of the access control. According to the *need-to-know* principle, user access controls and authorization procedures and its objective is to ensure that only authorized individuals gain access to information or systems necessary to undertake their duties.

(ii) *Scalability*: The reliability of the permission to access objects in the proposed system is based on the features of the public blockchain (i.e., decentralization, transparency, and irreversibility). However, the problem of the public blockchain scalability is a major constraint on these features contributing to the proposed system. Hence, the creation, modification, and disposal of access control policies must be done quickly, regardless of the network performance of the blockchain, even if the system operates on the public blockchain that offers only limited scalability.

(iii) *Revocation*: The data owner and data requester perform access control through the state transition of their off-chain channel state. The data owner manages access control policies by generating transactions that cause the off-chain channel's state transition ($\mathrm{state}_1 \longrightarrow \mathrm{state}_n$). Until the channel is closed, the data owner creates transactions that can change the off-chain channel's state and shares it with the data requester. The transaction is not propagated to the blockchain network, and it is kept personally by two participants in the off-chain before being used in the authentication process when the data requester requests access to the storage keeper. However, because transactions are shared only between the two participants (off-chain), the data requester may present transactions for change to the channel's past state ($\mathrm{state}_1 \longrightarrow \mathrm{state}_{n-k}$) for other purposes instead of transactions for the transition to the channel's current state ($\mathrm{state}_1 \longrightarrow \mathrm{state}_n$). To avoid this problem, the data owner must have a measure that prevents the previous state of the channel from being used by the requester in the data access process.

4.1. Phase 1: Setup. Both parties individually create the funding transaction of the same structure that transfers their funds (predefined amounts in negotiation) to a single 2-of-2 multisignature address as a deposit (except for the counterparty's digital signature). To prevent unauthorized modification of the transaction due to the order of the exchange of signatures [24], both parties do not exchange their signature until they have individually created a refund transaction. Both parties execute the following steps, as shown in Figure 3:

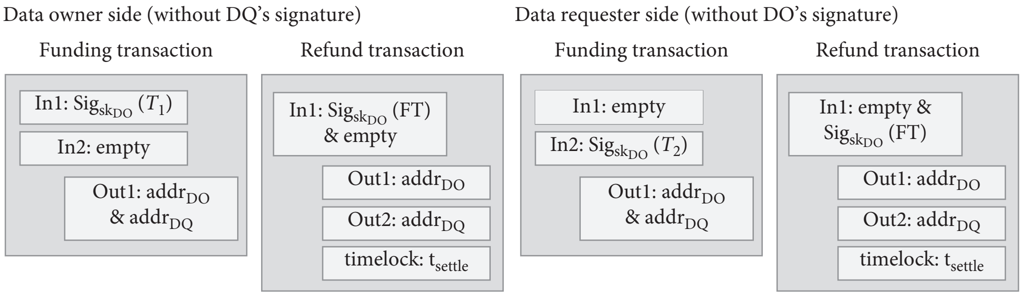

Data owner side (without DQ's signature)
Data requester side (without DO's signature)
Funding transaction
Refund transaction
Funding transaction
Refund transaction
In1: Sig$_{sk_{DO}}$ (T_1)
In2: empty
Out1: addr$_{DO}$ & addr$_{DQ}$
In1: Sig$_{sk_{DO}}$ (FT) & empty
Out1: addr$_{DO}$
Out2: addr$_{DQ}$
timelock: t$_{settle}$
In1: empty
In2: Sig$_{sk_{DO}}$ (T_2)
Out1: addr$_{DO}$ & addr$_{DQ}$
In1: empty & Sig$_{sk_{DO}}$ (FT)
Out1: addr$_{DO}$
Out2: addr$_{DQ}$
timelock: t$_{settle}$

Figure 3: Setup phase transaction structure.

(1) DQ provides their first address addrDQ,1 derived from their first private key and public key pair <sk$_{DQ,1}$, pk$_{DQ,1}$> to DO

(2) DO and DQ generate the funding transaction FT, which sends their deposits to the channel, and the refund transactions, which return the deposits after t_{settle}

(3) Exchange each other's FT and refund transaction

(4) Add their signature to the incomplete transaction that has been received

(5) Finally, DO and DQ establish their off-chain channel by propagating completed transactions to the blockchain network

4.2. Phase 2: Channel Management. After FT is finalized in the blockchain (i.e., the block depth including FT is six or higher), DO and DQ create a first state transaction off-chain to indicate the new access control policy. In this phase, they create transactions with different structures, as shown in Figure 4. The state transaction redistributes the deposit locked in the off-chain channel to DO and DQ. As described earlier, when a platform exists for matching DO and DQ, we assume that DQ already knows the hash value of the data h_{data}.

(1) DQ sends the hash value of the randomly selected value h_{r_1} to DO

(2) DO and DQ create a state transaction T_{state_1}, as shown in Figure 4 (where signatures contain the hash value of the data $h_{\text{data},1}$ as a digest message)

(3) DO and DQ attach their digital signatures to the state transaction and exchange it with each other

(4) DO and DQ complete the state transaction by adding their digital signature to the incomplete transaction that has been received

DQ requests data from the storage keeper by presenting proofs for user authentication and state transaction T_{state_1}. The storage keeper validates the access request based on proofs presented by DQ and the information shown on the blockchain ledger (Algorithm 1).

(1) DQ sends a request message $m = \{h_{\text{data1}}, \text{DO}, \text{addr}_{\text{DO}}, \text{addr}_{\text{DQ}}, \text{FT}, T_{\text{state},1}, r_1\}$ with their digital signature Sig$_{\text{skDQ}}$ (m)

(2) The storage keeper uses the *stateValidate* algorithm to validate the access request. The *stateValidate* algorithm verifies whether the request meets the following:

   (A) The validity of the off-chain channel

   (B) Whether the signature contained in $T_{\text{state},1}$ can be verified with the address contained in FT

   (C) Whether the signature presented by DQ can be verified using the address included in $T_{\text{state},1}$

   (D) Whether the signature contained in $T_{\text{state},1}$ can be verified using the hash operation results for r_1 presented by DQ as a digest message

A storage keeper can verify whether the message digest of the signatures in the state transaction contains the data requested by the DQ to determine the right of access to the object. However, if DO creates a new state transaction that includes signatures for new data to modify the access control policy of DQ, it cannot guarantee that the DQ does not use the state transaction created in the past. Therefore, we applied the replace-by-revocation used in [24] to PyRos, such that if the DQ used a ticket that had been revoked in the past to access an object whose access rights had been revoked, they would lose the amount deposited on the channel, as shown in Figure 5 and described as follows:

(1) DQ sends the hash value of the randomly selected value h_{r_2} to DO with their new address addr$_{\text{DQ},2}$.

(2) DO and DQ create a new state transaction T_{state_2}.

(3) DQ creates a revoked state transaction RT$_{\text{state}_1}$, which has the same structure as the state transaction T_{state_1}; however, it does not contain hash values of shared data $h_{\text{data},1}$ in the signature message digest.

(4) DO and DQ add their digital signature to the new state transaction and exchange it with each other. Additionally, only DQ performs the same process for RT$_{\text{state},1}$ and generates a signature to claim ownership of their deposits in the RT$_{\text{state},1}$, after which they send it to the DO.

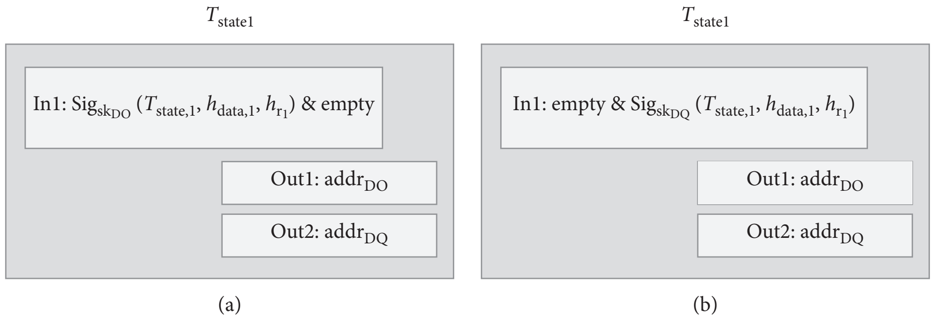

T_{state1}
In1: $\text{Sig}_{\text{sk}_{\text{DO}}}$ ($T_{\text{state},1}$, $h_{\text{data},1}$, h_{r_1}) & empty
Out1: addr_{DO}
Out2: addr_{DQ}
(a)
T_{state1}
In1: empty & $\text{Sig}_{\text{sk}_{\text{DQ}}}$ ($T_{\text{state},1}$, $h_{\text{data},1}$, h_{r_1})
Out1: addr_{DO}
Out2: addr_{DQ}
(b)

Figure 4: Channel management transaction structure (grant access). (a) Data owner side (without DQ's signature). (b) Data requester side (without DO's signature).

```
**Procedure** Channel_Validate
        Search (FT) in Blockchain network
        **if** FT.output was spent **then return** False
        **else return** True
**end procedure**
**Procedure** Verify_Signature in $T_{\text{state}}$
        **If** Verify$T_{\text{state}}$.input.$\text{Sig}_{\text{skx}}$, FT.output.$\text{addr}_x$ is True **then return** Ture
        **else return** False
**end procedure**
**Procedure** Verify_Signature in $m$
        **If** Verify ($\text{Sig}_{\text{Skx}}$ ($m$), $T_{\text{state}}$.output.$\text{addr}_x$) is True **then return** Ture
        **else return** False
**end procedure**
**Procedure** Verify_Hash ($m$)
        HashValue $\leftarrow$ Hash ($m.r$)
        $M \leftarrow$ Hash ($T_{\text{state}} || h_{\text{data}} ||$HashValue)
        **If** Verify ($T_{\text{state}}$.input.$\text{Sig}_{\text{skx}}$ ($M$), $\text{addr}_x$ is True **then return** Ture
        **else return** False
**end procedure**
```

Algorithm 1: *StateValidate.*

(5) DO and DQ complete received transactions by adding their digital signature to the incomplete transaction that has been received.

4.3. Phase 3: Close Channel. In the proposed system, we consider closing the channel in the following cases:

(A) When there is no further transaction between the data owner and the data requester, they create a closing transaction and propagate it to the blockchain network to apply the channel's final state to the blockchain. If the channel's final state is propagated to the blockchain network, all state transactions previously created on the channel are automatically invalid and return the deposit locked in the setup phase.

(B) In the case of a nonresponsive counterparty, the deposit in the channel can be returned to participants by automatically closing the channel as the refund transaction that had a time-lock t_{settle} is included in the blockchain after a time t_{settle}.

(C) If the use of the previously revoked state is detected during the verification process, the honest storage keeper will inform the use of the revoked state T_{state_x} that was received from DQ to DO. After the use of the revoked state has been confirmed, DO propagates the revoked state transaction $\text{RT}_{\text{state}_x}$ to the blockchain network and transfers DQ's deposits in $\text{RT}_{\text{state}_x}$ to their account using DQ's received signature in the modify permission process.

The state transition of the channel from phase 1 to phase 3 is shown in Figure 6. Figure 6 illustrates a scenario in which the access control policy was updated three times, and the channel was closed normally.

5. Security Analysis

5.1. State Privacy. Our goal is to protect users' privacy by preventing third parties that do not participate in access control for a particular user in PyRos system (i.e., except for the data owner, data requester, and storage keeper) from knowing the content of the transactions. The goal of state privacy is to protect the user's transaction information against the adversaries that monitor the blockchain network. First, the initial state of the channel, in which the participants transfer their deposits to the channel, and the refund

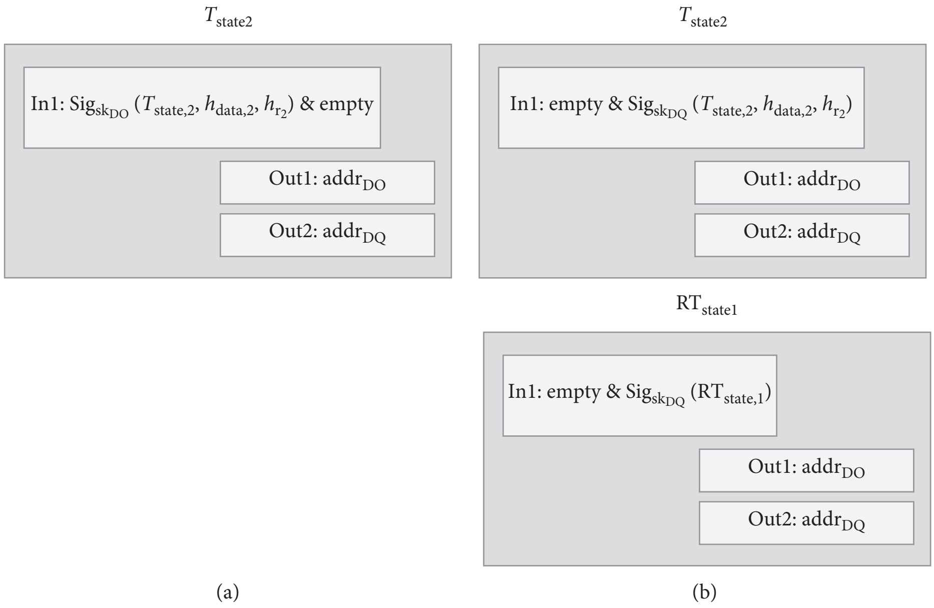

T_{state2}
In1: $\text{Sig}_{\text{sk}_{\text{DO}}}$ ($T_{\text{state,2}}$, $h_{\text{data,2}}$, h_{r_2}) & empty
Out1: addr_{DO}
Out2: addr_{DQ}
(a)
T_{state2}
In1: empty & $\text{Sig}_{\text{sk}_{\text{DQ}}}$ ($T_{\text{state,2}}$, $h_{\text{data,2}}$, h_{r_2})
Out1: addr_{DO}
Out2: addr_{DQ}
$\text{RT}_{\text{state1}}$
In1: empty & $\text{Sig}_{\text{sk}_{\text{DQ}}}$ ($\text{RT}_{\text{state,1}}$)
Out1: addr_{DO}
Out2: addr_{DQ}
(b)

Figure 5: Channel management transaction structure (modify permissions). (a) Data owner side (without DQ's signature). (b) Data requester side (without DO's signature).

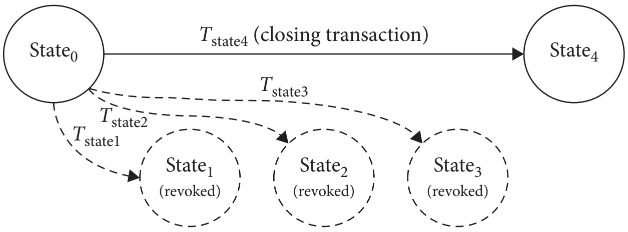

State_0
T_{state4} (closing transaction)
State_4
T_{state3}
T_{state2}
T_{state1}
State_1 (revoked)
State_2 (revoked)
State_3 (revoked)

Figure 6: State transition of off-chain channel.

transaction, in which the channel-locked deposits are returned to the participants, must be known in the blockchain network. As shown in Figure 3, the only information disclosed in funding transactions for channel establishment is the address, deposit of the participants who generate the channel, and their signature. If the participants close the established channel, the redistribution of deposits by the participants will be recorded in the blockchain. In the event of fraudulent use of the state, the channel will record the state transaction that has been revoked in the past by the honest participant and transfer the deposits of malicious participants locked in the revoked state to the other user. However, transactions that contain information about the shared object are not recorded in the blockchain, and all off-chain transactions are communicated only through the personal communication channel between participants. Consequently, it is highly unlikely that adversaries obtain significant transaction information only from transactions recorded in the blockchain. Moreover, because all transactions recorded in the blockchain in PyRos system follow the standard structure of cryptocurrencies, the adversaries cannot distinguish between transactions for payment and PyRos transactions. Even if the adversary chooses to target and monitor all of their packets, it will be difficult to find significant transactions, because the data requester will change their addresses used for each creation of the state transaction.

5.2. Scalability. The purpose behind solving the scalability problem is to ensure that the system will operate without delay, regardless of the throughput of the blockchain network, while assuming that the adversary can attempt various known attacks [16, 27] on the network. In the previous section, we assumed that access control through channels is performed after the FT establishing channels has been finalized in the blockchain. Therefore, the block containing FT cannot be modified in the blockchain after FT has been finalized with sufficient confirmation. In the proposed system, all transactions (except FT) need not be propagated to the blockchain network until the channel is closed. The implementation of existing centralized systems in the decentralized network required a majority of the network consensus instead of trust institutions; however, this resulted in a large transaction processing delay. Attacks on blockchain networks took advantage of these delays to achieve malicious purposes, such as double payments. However, as described in Section 2, a state channel that only requires agreement from the channel participants is free from this delay and can be operated regardless of the availability of the blockchain network, if the integrity of the initial state, which is the basis of the channel's reliability, is guaranteed.

5.3. Revocation. In the PyRos system, the data owner's access control policies can be expressed in the off-chain state of the channel. However, as described in Section 2, the off-chain state of the channel is not recorded in the blockchain, such that explicit revocation of the past state is practically impossible. Therefore, we employed the implicit revocation used in [24] to require the data requester to pay the penalty for fraudulent use, although they may use a state transaction that was revoked in the past. In modifying the permission phase, the data requester creates a revoked state transaction RT_{state} in the form of the standard transaction excluding shared data information in the signature message digest. Further, the data requester generates a digital signature that enables the use of their deposit transferred to the revoked state transaction and sends it to the data owner with the latter. If the data requester attempts to access data using a revoked state, the data owner can propagate RT_{state} to the blockchain network to close the channel and use the data requester's deposit as a penalty.

However, we assumed that all system participants, except the storage keeper, could be malicious. Therefore, a malicious data owner may propagate a revoked state transaction to the blockchain network, regardless of whether the state is used fraudulently or not. To prevent this problem, we have added new conditions for consuming the data requester's deposit in the revoked state transaction. The standard transaction structure records the address of the new owner in the blockchain for the amount used in the transaction. The new owner then attempts to use cryptocurrency by attaching a digital signature, which is generated by the key corresponding to the address recorded in the blockchain, to the new transaction. Only if this digital signature is valid will the transaction be recorded in the blockchain. However, we added the hash value h_{r_k} of the r_k selected randomly by the data requester to the condition for the consumption of the data requester's deposit in the revoked state transaction. Hence, the data owner requires a preimage of the hash result r_k, included in the revoked state transaction RT_{state} with the digital signature of the data requester to consume the data requester's deposit.

6. Evaluation

We evaluate this proposal through comparison with other studies. PyRos implements blockchain-based access control using the off-chain state channel. It exhibits a major characteristic of improving performance by applying off-chain computation processing based on the state channel. The evaluation focused on the delay required for access control.

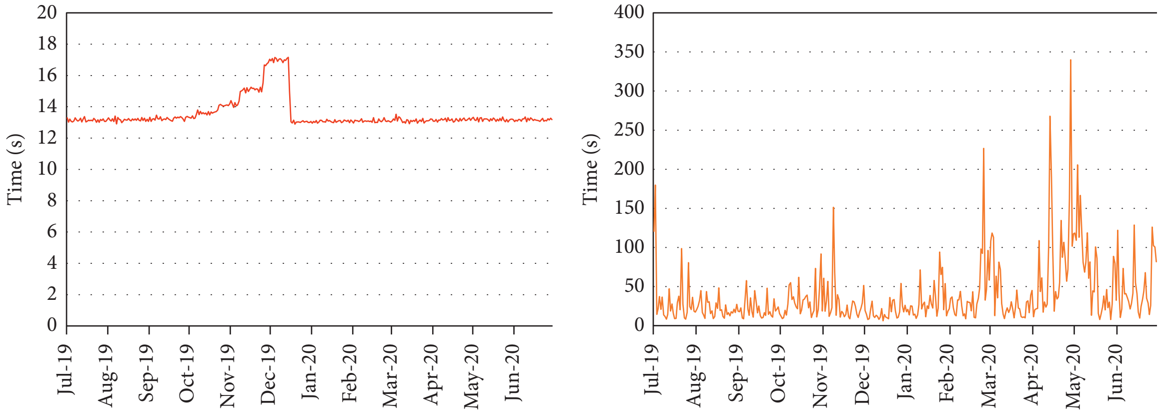
In the related studies on blockchain-based access control [10–12] mentioned in Section 2, a method of recording data related to access control was employed in the irreversible blockchain database. Blockchain technology provided users with key features in access control without the participation of centralized managers, which enabled the implementation of decentralized access control. However, considering a realistic data society environment, when an access control application is implemented in the public blockchain, its problem of scalability has become a major constraint. Access control applications can be implemented on private blockchain networks to solve performance problems. However, if the access control application is implemented on a private blockchain, the presence of the blockchain network administrator will not guarantee decentralization of access control and will not be able to implement dynamic access control services due to a limited pool of network participants. Nonreversive and decentralized databases are highly efficient for the storage and verification of access control policies. However, the probabilistic finality of the public blockchain will require minimal time for the irreversibility of stored access control policies to reach a secure level. Table 2 lists features associated with block generation of known public blockchain platforms.

The public blockchain platform uses a consensus protocol to maintain and manage a single and unique database without the trusted third party among unreliable network members. A consensus protocol ensures that the blockchain network operates even if there is no more than a certain percentage of malicious users (byzantine node) or users who cannot participate in the protocol (fault node) in the network. This algorithm contributes to maintaining a highly secure blockchain network without the trusted third party; however, it causes delays in the database's update process. As shown in Table 1, all public blockchain platforms limit the average block generation cycle through a consensus protocol. Most public blockchain platforms have a limited number of transactions that can be processed per unit time (known as TPS). Considering that this consensus process is required in all processes on the public blockchain application, the recording and updating of access control policies will consistently have the minimum delay required to create blocks on the blockchain platform. Figure 7 shows the average time it takes for blocks to be included in the Ethereum and Bitcoin blockchains. Ethereum takes an average of 14 s, and Bitcoin takes about a minute to connect a block. Hence, new data will not be quickly reflected in the blockchain if transactions that newly register or renew access control policies are excessively concentrated at a specific time, which could have a fatal impact on the availability of access control applications. In contrast, in PyRos, access control policies are represented as an off-chain state, such that the on-chain consensus process is not necessary. The availability of existing proposals depends on the performance of the blockchain network, onto which the application is implemented, whereas in PyRos, the performance of the network does not affect the availability of the access control application at all, except for the setup phase.

Table 3 shows the results of comparing PyRos and other researches [10–12] from a performance perspective. In the blockchain application, the biggest impact on performance is the network topology and consensus mechanism [28]. The public blockchain network enables secure management of access control policies. However, it takes a lot of time before requests for the state transfer in the blockchain state DB are reflected in the majority of network nodes. As shown in Figure 7, the time taken in this process changes to flexible depending on various factors such as the size of the blockchain network and consensus algorithm. However, regardless of the blockchain platform, this time commonly refers to the process of transactions being contained in the block by miners after they are propagated/verified to nodes in the network. References [10–12] are commonly based on the Bitcoin blockchain. Thus, all transactions associated with

Table 2: Information related to block generation of public blockchain platforms.

	Bitcoin	Ethereum	Ripple	Monero
Blockchain type	Permisionless	Permisionless	Permissioned	Permisionless
Block size (average)	1 MB	Variable (1,500,000 gas limit, averages in ~20–30 KB)	N/A	Variable (twice the median size of the last 100 blocks; the limit is 60 KB)
Block cycle (average)	10 min	10–19 s	N/A	2 min
Consensus algorithm	Proof-of-Work	Proof-of-Work	Ripple Protocol Consensus Algorithm (RPCA)	Proof-of-Work



(a)

(b)

Figure 7: Historical average time required for a block to be included in the Ethereum blockchain (a) and the Bitcoin blockchain (b).

Table 3: Comparison from a performance perspective.

	Ouaddah et al. [10]	Maesa et al. [11]	Xia et al. [12]	PyRos
Transaction/block validation	Majority of full nodes	Majority of full nodes	Majority of full nodes	Channel participant (after the channel is created)
Transaction/block propagation	Majority of full nodes	Majority of full nodes	Majority of full nodes	Channel participant (after the channel is created)
Block mining (consensus)	Required	Required	Required	Not required (after the channel is created)
Scalability	No	No	Customized block structure	State channel

the access control protocol must be propagated and verified by a majority of full nodes in the network. Reference [12] applied a method to reduce the size of block data propagated in the blockchain network through a customized block structure. However, they still had limitations in the process of the propagation and validation of the transaction. In contrast, PyRos can save significant processing time by omitting the propagation and validation process using a state channel. Instead of transferring the blockchain state DB, channel participants' requests that occur after the channel is created transfer only the state of the channel which is shared only among channel participants. Therefore, all access control events that occur in PyRos can be processed quickly without delay due to network processing.

7. Use Case and Future Studies

The proposed system can be applied to a variety of fields; however, we expect its particularly widespread use in the healthcare sector. As an example, we assume a scenario in which sellers and buyers promise periodic data transactions over a period of time, rather than simple data transactions that occur only once. A patient suffering from diabetes and a company studying diabetes drugs may sign a contract, in which the company receives health data from the patient once a week. Patients provide their health data to companies every week, and companies pay cryptocurrency, such as Bitcoin, in return. In this process, the data seller encrypts their data and keeps it in external storage. After the contract is signed with the data requester, the data seller creates channels on the blockchain with the requester instead of transmitting the data directly. Subsequently, the seller periodically grants an access right to the new data and the ability to decode it, and at the same time, the requester pays the seller cryptocurrency for the data.

However, we assume that the storage keeper is a semitrusted entity that is expected to act honestly upon legitimate requests. The storage keeper consistently provides the requested data after user authentication; however, this is an assumption that violates the decentralization aspect within the system's purpose. To solve this problem, we aim to attempt the implementation of a decentralized storage layer in a future study. P2P storage, such as the interplanetary file system (IPFS), serves as a good platform for this study, and we plan to conduct research that will assume control of access to encrypted data stored in distributed repositories across the blockchain.

8. Conclusions

We proposed PyRos, a system that supports data trading and sharing between individuals on top of the public blockchain. The public blockchain is more reliable than the private blockchain, as it is increasingly difficult for more users to manage the blockchain and attackers to attack all blocks. However, the scalability problem in the public blockchain network makes it difficult to quickly synchronize blockchain databases. Therefore, we proposed a system that supports the data sharing application between individuals by combining access control service based on the off-chain state channel on the public blockchain. In PyRos, the user's access control policy is represented by the state of the off-chain channel. The state of the off-chain channel can be changed by the agreement of the channel's participants, which can greatly reduce the costs required for agreement compared to the on-chain. Moreover, this approach is easy to implement in existing systems and does not require the addition of any new elements. We hope that this proposed system will contribute as a step toward a user-centric data society.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2015-0-00403), supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and partially supported by Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (no. 2019R1I1A3A01060652).

References

[1] D. Reinsel, J. Gantz, and J. Rydning, *The Digitization of the World from Edge to Core*, IDC White Paper, 2018.

[2] M. Viceconti, P. Hunter, and R. Hose, "Big data, big knowledge: big data for personalized healthcare," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 4, pp. 1209–1215, 2015.

[3] G. Wang, A. Gunasekaran, E. W. T. Ngai, and T. Papadopoulos, "Big data analytics in logistics and supply chain management: certain investigations for research and applications," *International Journal of Production Economics*, vol. 176, pp. 98–110, 2016.

[4] "Sources of big data: where does it come from?," 2020, https://www.cloudmoyo.com/blog/data-architecture/what-is-big-data-and-where-it-comes-from/.

[5] D. Rushe, *Facebook Sorry–Almost–for Secret Psychological Experiment on Users*, The Guardian, 2014.

[6] C. Cadwalladr and E. Mraham-Harrison, *Revealed: 50 Million Facebook Profiles Harvested for Cambridge Analytica in Major Data Breach*, The Guardian, 2018.

[7] *About–MyData.org*, https://mydata.org/about/, 2020.

[8] *Ocean Protocol—A Decentralized Data Exchange Protocol to Unlock Data for AI*, The Ocean Protocol Whitepaper, 2018, https://oceanprotocol.com.

[9] M. Anastasiu, S. Giacomelli, D. Hanson, C. Pennachin, and M. Argentieri, *SingularityNET: A Decentralized, Open Market and Inter-network for AIs*, The SingularityNET Whitepaper, 2020, https://public.singularitynet.io/whitepaper.pdf.

[10] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "Towards a novel privacy-preserving access control model based on blockchain technology in IoT," in *Europe and MENA Co-operation Advances in Information and Communication Technologies*Springer, Berlin, Germany, 2017.

[11] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," *IFIP International Conference on Distributed Applications and Interoperable Systems*, Springer, Berlin, Germany, 2017.

[12] Q. Xia, E. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, p. 44, 2017.

[13] K. Wüst and A. Gervais, "Do you need a blockchain?" in *Proceedings of the Crypto Valley Conference on Blockchain Technology (CVCBT)*, Zug, Switzerland, 2018.

[14] K. Qin and A. Gervais, *An Overview of Blockchain Scalability, Interoperability and Sustainability*, Hochschule Luzern Imperial College London Liquidity Network, London, UK, 2018.

[15] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, http://bitcoin.org/bitcoin.pdf.

[16] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *Proceedings of the USENIX Security Symposium*, pp. 129–144, Washington, DC, USA, 2015.

[17] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *proceedings of the International conference on financial cryptography and data security*, pp. 436–454, Christ Church, Barbados, March 2014.

[18] *Evaluation Forms for Blockchain-Based System Ver 1.0*, Ministry of Economy, Trade and Industry in Japan, 2020, http://www.meti.go.jp/press/2016/03/20170329004/20170329004.html.

[19] G. Fridgen, F. Guggenmoos, J. Lockl, A. Rieger, and A. Schweizer, "Developing an evaluation framework for blockchain in the public sector: the example of the German Asylum process," in *Proceedings of the 1st ERCIM Blockchain Workshop*, Amsterdam, Netherlands, July 2018.

[20] K. Croman, "On scaling decentralized blockchains," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 106–125, Christ Church, Barbados, February 2016.

[21] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proceedings of the IEEE Thirteenth International Conference on Peer-To-Peer Computing (P2P)*, pp. 1–10, Trento, Italy, 2013.

[22] E. Lombrozo, J. Lau, and P. Wuille, *BIP 0141: Segregated Witness (Consensus Layer)*, 2015, https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki.

[23] J. Chen and S. Micali, "Algorand," 2017, http://arxiv.org/abs/1607.01341.

[24] J. Poon and T. Dryja, "The bitcoin lightning network: scalable off-chain instant payments," *Medium*, vol. 9, p. 14, 2016 (Draft Version) 0.5.

[25] A. Miller, I. Bentov, R. Kumaresan, C. Cordi, and P. McCorry, "Sprites and state channels: payment networks that go faster than lightning," 2020, http://arxiv.org/abs/1702.05812.

[26] A. Gauba, "Finality in Blockchain Consensus," *Medium*, https://medium.com/mechanism-labs/finality-in-blockchain-consensus-d1f83c120a9a, 2018.

[27] P. Wuille, *BIP 0032: Hierarchical Deterministic Wallets*, 2012, https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki.

[28] P. W. Eklund and R. Beck, "Factors that impact blockchain scalability," in *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, pp. 126–133, Limassol, Cyprus, November 2019.