IJNS
International Journal of Network Security

ISSN 1816-353X (Print)
ISSN 1816-3548 (Online)

Vol. 26, No. 3 (May 2024)

INTERNATIONAL JOURNAL OF NETWORK SECURITY

PUBLISHING OFFICE

Min-Shiang Hwang
Department of Computer Science & Information Engineering, Asia University, Taichung 41354, Taiwan, R.O.C.
Email: mshwang@asia.edu.tw

International Journal of Network Security is published both in traditional paper form (ISSN 1816-353X) and in Internet (ISSN 1816-3548) at http://ijns.jalaxy.com.tw

PUBLISHER: Candy C. H. Lin

Volume: 26, No: 3 (May 1, 2024)

International Journal of Network Security

A New Deep Learning Paradigm for IoT Security: Expanding Beyond Traditional DDoS Detection

Saima Siraj Qureshi[1], Jingsha He[1], Nafei Zhu[1], Min Jia[2], Sirajuddin Qureshi[1], Faheem Ullah[1], Ahsan Nazir[1], and Ahsan Wajahat1[1]

(Corresponding author: Nafei Zhu)

Faculty of Information Technology, Beijing University of Technology[1]
Beijing 100124, China
School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150080, China[2]
Email: znf@bjut.edu.cn

(Received Oct. 11, 2023; Revised and Accepted Jan. 23, 2024; First Online Apr. 25, 2024)

The Special Issue on Cybersecurity and Privacy in the Industrial Internet of Things (IIoT)
Guest Editor: Prof. Zhengyi Chai (Tiangong University, China)

Abstract

With the proliferation of Things connected to the Internet (IoT), network vulnerabilities to Attacks known as distributed denial of service (DDoS) have escalated. Conventional DDoS detection methods often falter in the multifaceted IoT landscape. Addressing this, our research introduces a novel hybrid deep learning model, termed CNN-LSTM-GRU, which synergistically integrates (CNN) Convolutional Neural Networks,(LSTM) Long Short-Term Memory, and (GRU) Gated Recurrent Units. Early findings indicate a marked enhancement in detection precision and a reduction in false alarms when juxtaposed with existing methodologies. This paper champions a cutting-edge, versatile deep learning strategy utilizing the CNN-LSTM-GRU fusion to adeptly discern varied network threats. Our methodology harnesses feature clusters from UNSW-NB15 and BOT-IoT Flow datasets, encompassing protocols like DNS, FTP, HTTP, MQTT, and TCP. Based on metrics like accuracy, recall, precision, and F1-score, performance evaluation reveals that our hybrid deep learning model boasts a 98.45% detection rate against IoT-centric threats. Additionally, a comparative analysis underscores the superiority of our model against other leading detection frameworks.

Keywords: CNN; DDOS; Deep Learning; GRU; LSTM; Network Attacks

1 Introduction

At the beginning of the digital age, the Internet of Things (IoT) ushered in a new era in terms of how people connect to and make use of technology. The Internet of Things has made it possible for objects to communicate with one another, share information with one another, and collaborate in real time. This level of connection was previously inconceivable. Today, the effect of the Internet of Things can be found virtually everywhere [16]. DDoS attacks aim to disrupt services by overwhelming target systems with packets beyond their processing capacity. To amplify these attacks, culprits utilize "zombie" computers, which are essentially devices compromised by malware. As a result of these attacks, legitimate users often find their requests unanswered due to the network congestion caused by the flood of malicious packets. Among the various DDoS attack types, including the SYN, ICMP, and UDP floods, and http flood are the most commonly observed [20]. A flood attack occurs when an attacker uses the User Datagram Protocol (UDP) to send out a large number of packets without authorization. which is a fast data sharing technology [12]. A SYN flood assault is a form of attack that is based on the occupancy of servers by delivering packets with a spoofed IP address to the victim's servers. This sort of attack takes advantage of a vulnerability in the triple handshake the Transmission Control Protocol (TCP) protocol.An Internet Control Message Protocol (ICMP) flood attack is an attack in which an excessive number of pings are sent to the computer of the target by taking advantage of sending ICMP packets, which are the messaging protocol for regulating network traffic, waiting without for any response. This type of attack is known as a DoS attack [4]. An HTTP flood attack is a kind of cyberattack that propagates fake request headers to the targeted websites via zombie machines, hence causing service disruptions. The server's resources may be exhausted by this kind of attack. Both automated factories and smart cities fall un-

der this category. Something along the lines of this old proverb states, "With great power comes great responsibility." Distributed denial of service attacks have become a common danger due to fraudsters' increased access to a wider audience as a result of the widespread use of Internet of Things devices [27].

A disruptive denial-of-service attack, or DDoS attack, requires an excessive amount of traffic to be directed towards the targeted computer system, network, or online service [25]. The main goal is to deplete the target's resources to the extent that it becomes unusable and real users are denied access. The target's security will be compromised in order to do this. In the complex web of the Internet of Things, a successful distributed denial of service attack can have catastrophic consequences. For instance, picture a world where vital resources like water and electricity supplies, hospital medical equipment, and power systems are all under threat. Such disasters could have a cascading impact that endangers people's lives, destroys economies, and erodes confidence in digital infrastructure [5].

For a very long time, the cybersecurity community relied on conventional DDoS detection methods to protect the integrity of their systems. The mainstays of this field of study have been both signature-based strategies, which rely on previously identified patterns of known assaults, and anomaly-based methods, which look for significant deviations from norms [19]. However, the limitations of these strategies have been brought to light by the diversity and sheer volume of Internet of Things devices, as well as the dynamic nature of DDoS attacks. Never before has there been a moment when a more trustworthy, adaptable, and intelligent detecting system was not urgently needed.

One branch of machine learning called deep learning has shown itself to be highly skilled at finding subtle patterns in large amounts of data. It is the following stage of the procedure. The detection of distributed denial of service assaults in the Internet of Things may be revolutionized by deep learning models, which take their cues from the neural networks found in the human brain [14]. IoT traffic is unique in that it uses many different protocols, has a wide range of data speeds, and exhibits a wide range of device behaviors. Because of this, handling all of these characteristics of the traffic requires a customized solution.

We present a novel deep learning architecture created specifically for the detection of IoT DDoS in light of these difficulties. This model combines the features of GRU, LSTM, and CNN. The three distinct neural network types that this model successfully combines are as follows: Convolutional neural networks, or CNNs, are capable of spotting patterns in Internet of Things (IoT) data and capturing the subtleties of interactions between devices [7]. Their ability to extract spatial properties is widely acknowledged. With their expertise in modeling temporal sequences, the LSTM network can be used to track how traffic patterns change over time. These networks could be able to detect minute irregularities that point to a potential attack.

Last and Thirdly, Gated Recurrent Units (GRUs), renowned for their efficient learning dynamics and assurance of precise and quick identification, help modify the model's performance. These courses are well-known for their effective learning and have been around for a while. The main goal of the CNN-LSTM-GRU model is to combine the best features of these architectures to offer a comprehensive detection solution. As a result, the model will be able to identify the intricate patterns and sequences typical of Internet of Things data. This paper aims to explore this idea in greater detail by elucidating its methodology, experimental setup, results, and long-term consequences for Internet of Things security. With terms like "IoT security," "deep learning," "neural networks," "DDoS detection," and "hybrid model," this introduction sets the reader up for a thorough examination of the novel CNN-LSTM-GRU technique and its potential to fortify the defenses of the internet of things (IoT) against DDoS attacks. The section also uses phrases like "DDoS detection," "IoT security," and "hybrid model."Because of deep learning, this study suggests a hybrid IoT threat analysis technique that is robust, dependable, and efficient. Deep neural networks (CNN-LSTM-GRU) were employed in the proposed hybrid model to detect new cyber threats and attacks.

The primary contributions of the paper are as follows:

- The paper suggests a unique, flexible, and adaptive DL-based inquiry methodology that effectively identifies different threat classes in a conventional network through hybrid (CNN-LSTM-GRU) computations.
- We discovered that 29 features in the Bot-IoT are either measurable or equivalent to the features in the UNSW-NB15 data set after comparing the features in the two data sets with the attributes in the suggested system.
- The suggested method has been evaluated using common performance evaluation metrics, including F1-score, accuracy, recall, and precision.
- A comparison is also made between the present model and other hybrid deep learning-driven classifiers, such as long short-term memory, deep neural networks, and other earlier research. A thorough evaluation of the suggested method using 10-fold cross-validation has been conducted.

The remainder of the article, Section 2, addresses ideas for current literature from previous years.The shortcomings and difficulties with previous research are also listed in this section. Section 3 presents the approach (i.e., datasets, pre-processing, methodologies, and algorithms) for the proposed hybrid architecture. In Section 4, the results and assessment of the proposed method are outlined, together with a synopsis of the performance evalu-

ation standards that were applied. Section 5 contains the paper's conclusion as well as a plan for future study.

2 Related Work

Advanced research and countermeasures have become necessary due to the increase of Distributed Denial of Service (DDoS) assaults, which are distinguished by their increasing complexity and regularity. Four general types of DDoS assaults can be distinguished: http flood, ICMP flood, SYN flood, and UDP flood.

Attackers use the User Datagram Protocol (UDP) to quickly send out a large number of packets without the recipient's permission in a UDP flood attack. The SYN flood attack, on the other hand, floods the target's servers with packets containing forged IP addresses by taking advantage of a flaw in the Transmission Control Protocol's (TCP) triple handshake procedure. ICMP packets, which are necessary for controlling network traffic, are used in the ICMP flood attack to bombard the victim's system with ping requests without waiting for a response. Finally, http flood assaults cause disruptions to services by using zombie devices to send erroneous requests to websites that they target, so using up server resources.

An entropy method known as Shannon entropy has been used to identify these DDoS attacks. In order to create the detection model, this approach primarily focuses on particular attributes, such as the source IP address. However, utilizing programs like scapy and hping, attackers have come up with ways to quickly change the original IP address. The validity of employing the diversity of this property as a detection criterion has been called into question due to its flexibility.

Numerous investigations in this field have focused on the source IP address and used the Shannon entropy method to detect DDoS attacks [3,10]. But attackers utilizing scapy and hping can quickly change this address, raising doubts about its effectiveness.13] argued that a crucial component of DDoS detection, the variety of the originating IP address, might not be a reliable measure. In [22] Deep learning intrusion detection methods, such as DNN, CNN, and RNN architectures, have been developed in the context of Agriculture 4.0. These models use binary and multiclass classifications to assess network performances. They used the CIC-DDoS2019 and TON_IoT datasets to train their algorithms. In [6,23] unveiled a thorough DDoS attack detection system for 5G and B5G that combines an effective feature extraction technique with a composite multilayer perceptron. Their suggested framework demonstrated a low loss of 0.011 and an astounding accuracy of 99.66%. In [23] presented an advanced network intrusion detection system (A-NIDS) that uses an LSTM classifier in conjunction with an improved Onevs-One approach NSL-KDD and CIC-IDS2017 datasets were used to evaluate this system's efficacy. [9] presented a brand-new deep learning system that uses a feed-forward neural network model with embedding layers for multi-class classification to detect Internet of Things intrusions. [8] created a hybrid model that combines two deep learning techniques to detect DDoS attacks. The autoencoder part of their model was quite good at extracting features and identifying the most important feature sets. Their model's Multi-layer Perceptron Network segment achieved an F1-score of over 98% while addressing performance overhead for various forms of DDoS attacks. [28] assessed the performance of feature selection methods on modern datasets, providing summaries of different approaches. Following feature extraction, they compared the lengths of feature selection and training on the same dataset. [13, 18] presented a DL model based on LSTM that may identify DDoS assaults in the SDN control layer with a 98.88% accuracy on the ISCX 2012 and IDS CTU-13 Botnet datasets. [11] presented a hybrid CNN-based intrusion detection technique that combines a GRU model with a CNN. The GRU module was selected because it can retrieve important information from previously collected data by using memory cells and capturing long-dependence properties. [2] developed a 96% accurate Bidirectional LSTM-based framework for IoT-botnet packet detection. CNN and RNN were used to analyze network traffic flow with 99.3% accuracy [1]. [17] benefited from website content and metadata by using a deep learning-based LSTM to detect bots with a 98% accuracy rate. [24] used a variety of deep learning (DL) methods, including CNN, RNN, and LSTM algorithms, to identify domain names independently of data context. The suggested method produced a 90% detection accuracy. This thorough analysis study emphasizes how DDoS detection methods are constantly changing and how attempts are being made to increase their effectiveness.

3 Methodology

In this section, we introduce the architecture of our proposed hybrid DL model, as illustrated in Figure 1. These models involve a series of crucial steps: first, In order to identify comparable features between the UNSW-NB15 and Bot-IoT datasets, a feature comparison is performed. This is followed by feature selection, data pre-processing, refinement, and finally, the training of the model using a hybrid approach that combines CNN, LSTM, and GRU deep learning techniques.

Feature Comparison: Within our system, we compared features from both datasets. From the Bot-IoT dataset, 29 traits were identified by our study. either matched or could be equated to those in the UNSW-NB15 dataset.

Feature Selection: For our system, we categorized features from the BOT-IoT and UNSW-NB15 datasets into clusters based on flow, DNS/FTP/HTTP, MQTT, and TCP. A significant number of these features were grouped into the flow and TCP clusters, as detailed in Table 1. This clustering was informed by

a thorough analysis of each feature's description as provided by the original authors. Our goal was to retain a minimal set of features that still encompassed both the application and transport layers. The application layer is primarily represented by flow features, while the transport layer is dominated by the TCP protocol. By focusing on these clusters, we optimized the scenarios to retain the maximum packet information, which in turn considerably decreased the time spent computing during the learning stage.

Data Preprocessing: Here, we have delve into the various data preprocessing stages:

1) **Data Type Resolution**: Certain features in our model, such as 'saddr', 'daddr', and 'proto', are categorical and need conversion to a format suitable for algorithms. Specifically, 'saddr' and 'daddr' represent source and destination IP addresses, while 'proto' indicates the flow's protocol type. We gave each of these IP addresses a number.

   In the UNSW-NB15 dataset, there are 49 IP addresses, and the Bot-IoT dataset contains 301. When merging the datasets, We used 350 instead of the IP addresses' unique, randomly generated integers. This not only prevents overfitting but also retains the significance of IP addresses in training and validation datasets, especially for features that rely on them. Similarly, the 'proto' feature was converted to an integer type.

2) **Handling of Missing Port Numbers** : In the complete Bot-IoT dataset, packets using the ARP protocol lack source and destination port numbers. This omission is expected. As noted by Koroniotis *et al.* in [20], ARP port numbers (used by 5% of the Bot-IoT dataset) were assigned the value -1. We adopted this approach for our model, assigning this value to the port number in the entire dataset where the ARP protocol appeared.

3) **Z-scale Normalization**: Normalization ensures that data across different features have a similar distribution, allowing the model to assign comparable importance to each feature. If we consider a feature subspace with N rows and M columns, represented as $X = RN \times M$, z-scale normalization can be applied in the following manner:

   - **Hybrid CNN-LSTM-GRU** After processing, the input data is directed to the training phase. Subsequently, we conducted tests using DNN-LSTM, CNN-LSTM, CNN-BiLSTM, and our newly proposed CNN-LSTM-GRU. The promising outcomes from the CNN-LSTM inspired us to design a hybrid model that combines the strengths of CNNs, LSTM, and GRU. This innovative approach yielded superior results. The structure of this hybrid model can be viewed in the training step of Figure 1.

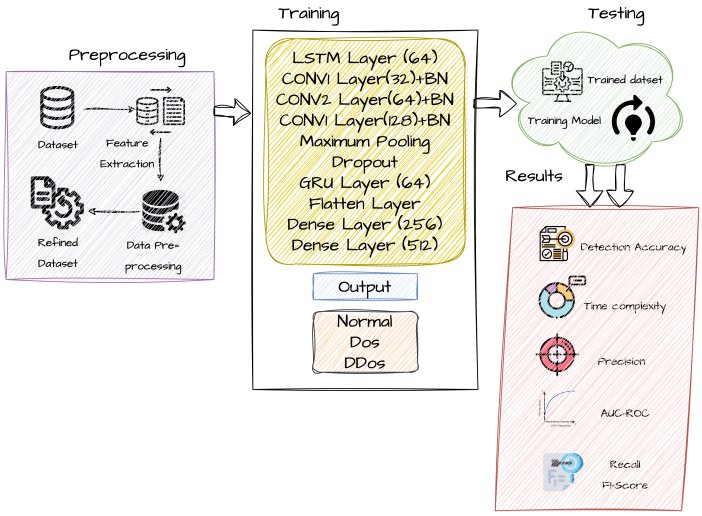


Figure 1: Proposed hybrid CNN-LSTM-GRU architecture

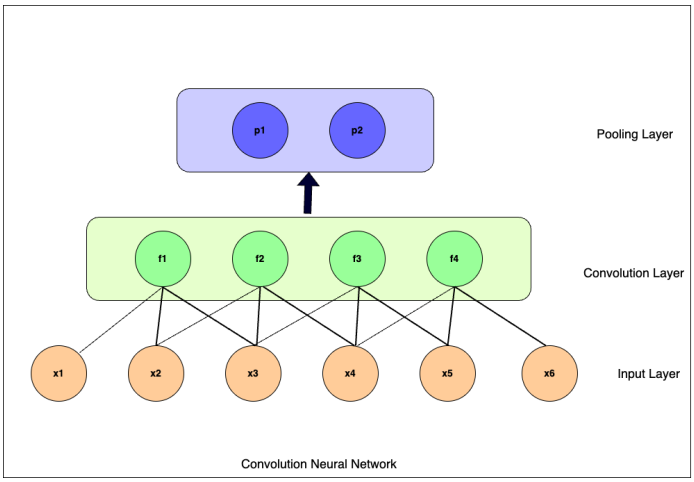


Figure 2: Simple convolution and pooling layer architecture

CNNs are usually used to handle two-dimensional data and are primarily developed for image classification. But time series analysis, which works with one-dimensional data, has also proven useful. The weight-sharing idea is a fundamental component of CNNs and provides improved performance for nonlinear tasks. Figure 2 shows the complex operation of the convolution and pooling layers. Figure 2 shows how input data points such as x1 through x6 are converted into feature maps f1 through f4 by applying convolution. These feature maps are further refined by a pooling layer after the convolutional layer, further abstracting them for usage in conjunction with memory cells and hidden layers.

RNNs, however, are not without their difficulties. The exploding and vanishing gradient problem afflicts them. This issue could lead to the gradient for long-term temporal components becoming exponentially quicker than for short-term ones., especially with expanding gradients. GRU and LSTM are the two most common forms of RNNs. RNNs have backward connections, which can occasionally negatively impact model accuracy, in contrast to CNNs. LSTMs, however, deal with these drawbacks. They are an example of a sophisticated RNN architecture designed with long-range temporal feature dependencies in mind. Looking closer, we can see that the LSTM is made up of cell blocks. These blocks switch between cell and hidden states, and memory blocks use gates to hold onto state information. The three gates of input, forget, and output define an LSTM cell. A GRU, on the other hand, just has two: the update (Z) and reset (Y) gates. The reset gate combines the input sequence of the next cell with the memory of the previous one, while the update gate decides how much of the previous cell's memory is still active. LSTMs are well known for their ability to assess long series and retain knowledge across datasets. According to [26], they outperform a lot of other deep learning algorithms in terms of test completion speed. To gain a deeper understanding of the LSTM cell, consider that it consists of two states (cell and hidden) and three gates (input, forget, and output). Below are the mathematical expressions for these LSTM gates.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f).$$

After deciding to keep the data, the next step is to update the cell's state, which is done by use of an input gate, i_t:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i).$$

The function than, which is a hyperbolic tangent, produces a vector of new potential values, c_t:

$$c_t = \tanh\left(W_c.[h_{t-1}, x_t] + b_c\right).$$

Multiplying the current candidate value by the previous value and f_t yields the new value under consideration. The equation is further complicated by the addition of i_t* c_t is added to the equation.

$$c_t = f_t * c_{t-1} + i_t * c_t$$

The final result is a filtered representation of the cell state, denoted byo_t .

$$\begin{aligned} o_t &= \sigma(W_o * [h_t - 1, x_t] + b_o) \\ h_t &= o_t * \tanh(c_t) \end{aligned}$$

The basic LSTM cell accepts organized data as input, and additionally, the input layer is linked to hidden layers. The size of the output layer is determined by the quantity of classes that must be classified. But LSTM is a little different in a few respects. To start, whereas the GRU cell has two gates, LSTM has three. Second, the input and forget gates in the LSTM are combined to create the update gate, and the reset gate for the hidden state is applied immediately.

A popular paradigm for deep learning algorithms is GRU. GRU is thought to train models 3.6% quicker than other deep learning algorithms, making it the fastest learning model [15]. The cell state is swapped out for a concealed state for data transfer in the modified GRU design. A reset gate and an update gate are the two gates in the GRU model. By managing the data flow through the model with these two gates, the model may refine the output. Information can be retained in a longer sample sequence using a gated recurrent model.The updated gate functions as an input and forget gate for the LSTM. Therefore, the updated gate chooses which data to erase and keep in certain cells. When and what are forgotten are decided by the reset gate. The GRU learns more quickly than the LSTM because it uses fewer tensor operations. The GRU equations that examine the values of two gates and the state of the cell using the GRU algorithm are defined below. Figure 3 displays the general architecture of the RNN, LSTM, and GRU.

$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \tag{1}$$

Input is multiplied by weight in Equation (1) to determine the update gate at time step t.

$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \tag{2}$$

Equation (2) depicts the computation at the reset gate, where the input is multiplied by weight by the update gate at time step t.

$$\widetilde{h}_t = \tanh(W.[r_t * h_{t-1}, x_t]) \tag{3}$$

The current memory is shown in Equation (3) when input is multiplied by weight.

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t \tag{4}$$

Equation (4) depicts the final memory of the time step in which the update gate is multiplied element by element.

A CNN (Convolutional Neural Network) is a sort of model of deep learning that is designed to perform particularly well when processing structured grid data, such as pictures. CNNs automatically learn hierarchical characteristics from the data that is fed into them, and they do this by utilizing layers such as convolutional, pooling, and fully connected [15, 21]. They begin by identifying simple patterns and then progress to recognizing more complicated structures; as a result, they play an essential role in activities such as the classification of images, the detection of objects, and the identification of faces. CNNs have revolutionized computer vision applications because of their ability to learn spatial characteristics in an adaptable manner and reduce the requirement for feature extraction manuals.

In the proposed framework Figures 1 and 4, employ the LSTM layer for prioritizing sequential modeling over

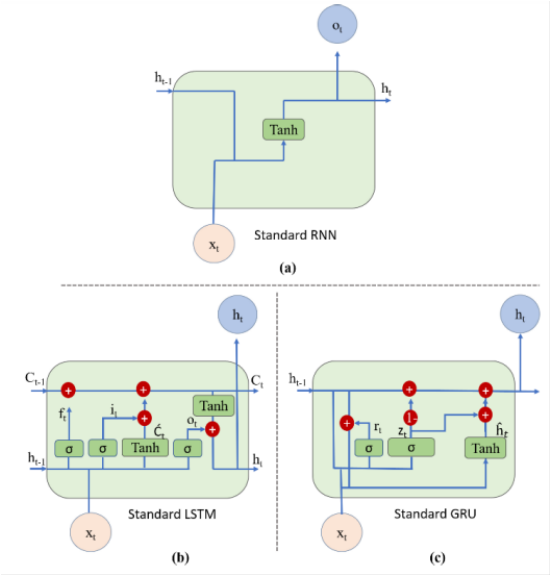


Figure 3: Standard architecture of RNN, LSTM and GRU

spatial feature extraction. A GRU learns a sequence effectively after using CNN features for sequence representation. After the input data has been adjusted, the CNN layers are utilized to extract spatial characteristics, which are then fed into GRU. In this research, we employed three CNN layers with a kernel size of three and a Relu activation function. The first, second, and third layers' filters were 1×32, 1×64, and 1×128 correspondingly. The spatial features are extracted, and then they are fed into GRU layers. Temporal features are modeled by a GRU layer, and IDS prediction is done by a dense layer. The datasets are divided into two parts: 80% and 20%, respectively, for training and testing. Input Layer: Depending on your specific task, the input data can be sequences (e.g., text or time series) or images.

LSTM Layers: The input data is directly fed into LSTM layer as the first step. These LSTM layers are responsible for capturing sequential dependencies and temporal patterns in the data [21].

CNN Layers: Extract relevant features from the sequences generated by the LSTM layers. Flatten or Global Max Pooling Layer: After the CNN layers, you can flatten the output or use global max pooling to convert the 2D feature maps into a 1D vector.

GRU Layers: Optionally, after the CNN layers, add GRU layer to further model sequential information. This can be especially useful if there are complex temporal dependencies that the LSTM layers may not capture adequately.

Output Layers: Add appropriate output layers, such as dense layers for sequence tasks.

Output: The final output of the model is used for making IDS predictions

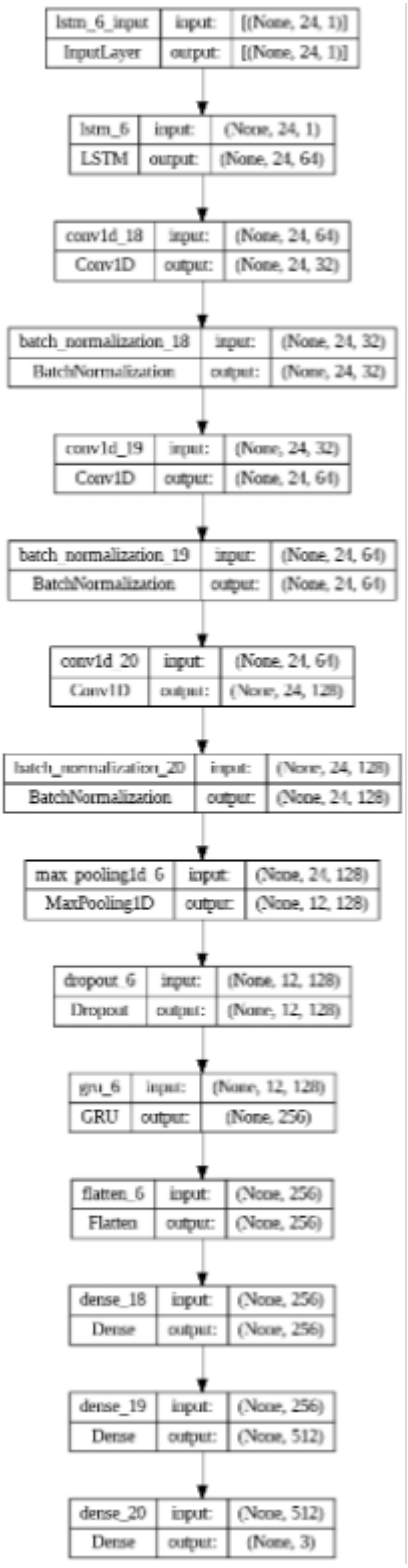


Figure 4: Graphical representation of CNN-LSTM-GRU model

4 Result and Discussion

We conducted a thorough evaluation of our proposed intrusion detection method using standard performance metrics such as accuracy, precision, recall, F1-score, and more. These metrics are derived from the confusion matrix using mathematical computations. Additionally, we've illustrated the AU-ROC curves to visually represent the relationship between positive and negative rates. Essential parameters like true positive (TP), false positive (FP), true negative (TN), and false negative (FN) are also extracted from the confusion matrix. Here's a concise overview and mathematical foundation of these performance metrics:

1) **Confusion Matrix**: This 2D matrix showcases the relationship between actual and predicted values. True rates reflect the classifier's overall correct predictions, whereas negative rates highlight incorrect predictions.

2) **Accuracy**: A primary metric, accuracy gauges the classifier's overall performance. It captures the proportion of samples correctly classified, both positives and negatives. Its formula is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

3) **Precision**: Precision quantifies the proportion of true positive detections to the total positive detections. Its formula is:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

4) **Recall**: Recall, on the other hand, is the ratio of the true positive rate to the sum of the true positive and false negative rates. Its formula is:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

5) **F-measure**:Representing the harmonic mean of recall and precision, the F1-score's formula is:

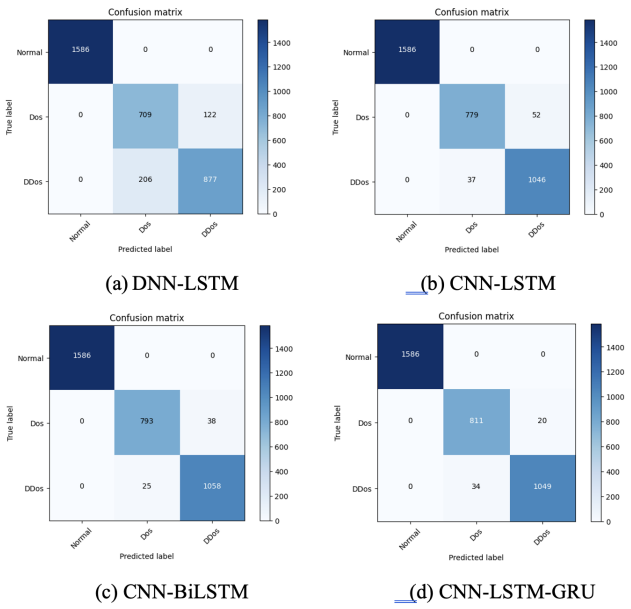
$$F1 - score = \frac{2 * TP}{2 * TP + FP + FN} \tag{8}$$

6) **AU-ROC**: This metric illustrates the classifier's diagnostic capability graphically. The curve plots the true positive rate against the false positive rate across varying thresholds.

$$P(X_1 > X_0\ ) = P(X_1 - X_0 > 0) \tag{9}$$

Its formula involves X1, the random variable denoting the rate for random positive samples, and X0, the continuous random variable representing the rate for randomly chosen negative samples.

$$AU - ROC = \int_0^1 TPR(FPR)d(FPR) \tag{10}$$



(a) DNN-LSTM (b) CNN-LSTM

(c) CNN-BiLSTM (d) CNN-LSTM-GRU

Figure 5: Confusion matrix for multi-classification derived from various deep learning methods

$$AU - ROC = \int_0^1 TPR(FPR_1(x))dx \tag{11}$$

The outcomes of our evaluation are tabulated in Table 1. A glance reveals the superior performance of our CNN-LSTM-GRU model relative to other methods. Specifically, the DNN-LSTM algorithm lags behind other deep learning models, with the standard CNN achieving an accuracy of 90.62%. Remarkably, the fusion of CNN with LSTM surpasses all other algorithms, achieving an impressive accuracy of 98.45%. This underscores the potency of our hybrid CNN-LSTM-GRU model in intrusion detection. Additionally, the hybrid CNN-LSTM model boasts superior precision and recall compared to its counterparts. Yet, when considering the F1-score across three classes, the CNN-LSTM-GRU model emerges as the clear frontrunner.

Furthermore, we present the efficacy of our suggested approach in classifying both regular and malicious data, specifically DOS and DDoS attacks. Figure 1showcases the confusion matrix (CM) derived from the testing phase for various deep-learning strategies. Every instance in this test set is categorized as either regular or malicious activity. Notably, our advanced CNN-LSTM-GRU model demonstrates superior precision in accurately identifying malicious events. Our CNN-LSTM-GRU model performed well in experiments, with F1-scores between 0.9766 and 0.9811. Table 2 shows the model's high precision and recall metrics, which indicate strong predictive dependability and demonstrate its ability to effectively identify and define dataset cases."

Figure 5 shows the confusion matrix for multi-classification derived from various deep learning methods. To delve deeper into the performance of our ad-

Table 1: Comparison of different models

Model	Accuracy	Precision	Recall	F1-Score
DNN-LSTM	0.9062	0.9087	0.9062	0.9066
CNN-LSTM	0.9745	0.9745	0.9745	0.9745
CNN-BiLSTM	0.9820	0.9820	0.9820	0.9819
CNN-LSTM-GRU	0.9845	0.9846	0.9845	0.9845

Table 2: Metrics for Models 1-10

Metrics	Model	1	2	3	4	5	6	7	8	9	10
Accuracy	DNN-LSTM	0.7697	0.7617	0.7742	0.7725	0.7685	0.672	0.7684	0.7661	0.7581	0.7026
	CNN-LSTM	0.9771	0.9748	0.9662	0.9668	0.9148	0.972	0.9817	0.9251	0.9651	0.8982
	CNN-BiLSTM	0.9657	0.9782	0.972	0.9634	0.9742	0.9634	0.9834	0.9748	0.9645	0.9285
	CNN-LSTM-GRU	0.9765	0.9782	0.9714	0.9857	0.9748	0.9851	0.9788	0.9765	0.9799	0.9811
Precision	DNN-LSTM	0.8666	0.6958	0.8490	0.7875	0.6348	0.6712	0.8354	0.7312	0.8175	0.809
	CNN-LSTM	0.9777	0.9749	0.9662	0.9668	0.9354	0.9719	0.9818	0.9330	0.9651	0.9266
	CNN-BiLSTM	0.9658	0.9785	0.9719	0.9634	0.9759	0.9635	0.9835	0.9760	0.9659	0.9372
	CNN-LSTM-GRU	0.9767	0.9784	0.9714	0.9858	0.9748	0.9856	0.9788	0.9773	0.9803	0.9811
Recall	DNN-LSTM	0.7697	0.7617	0.7742	0.7725	0.7685	0.672	0.7684	0.7661	0.7581	0.7026
	CNN-LSTM	0.9771	0.9748	0.9662	0.9668	0.9148	0.972	0.9817	0.9251	0.9651	0.8982
	CNN-BiLSTM	0.9657	0.9782	0.972	0.9634	0.9742	0.9634	0.9834	0.9748	0.9645	0.9285
	CNN-LSTM-GRU	0.9765	0.9782	0.9714	0.9857	0.9748	0.9851	0.9788	0.9765	0.9799	0.9811
F1-score	DNN-LSTM	0.6864	0.6873	0.6974	0.7002	0.6838	0.5751	0.7514	0.6922	0.7409	0.6367
	CNN-LSTM	0.9772	0.9748	0.9662	0.9668	0.9149	0.9719	0.9817	0.9255	0.9651	0.8978
	CNN-BiLSTM	0.9657	0.9783	0.9719	0.9634	0.9743	0.9634	0.9834	0.9749	0.9646	0.9289
	CNN-LSTM-GRU	0.9766	0.9783	0.9714	0.9857	0.9748	0.9851	0.9788	0.9766	0.9800	0.9811

Table 3: FDR, FNR, FOR and FPR values of DNN-LSTM, CNN-LSTM, CNN-BiLSTM and CNN-LSTM-GRU

Metrics	DNN-LSTM	CNN-LSTM	CNN-BiLSTM	CNN-LSTM-GRU
FDR	0.1221	0.0473	0.0346	0.0187
FNR	0.1902	0.0341	0.0230	0.0313
FOR	0.2251	0.0453	0.0305	0.040
FPR	0.1468	0.0625	0.0457	0.024

vanced CNN-LSTM-GRU model, we employ the receiver operating characteristics (ROC) curve, depicted in Figure 6. This curve elucidates the relationship between true-positive and false-positive rates, with the area under the curve (AUC) serving as an indicator of the model's proficiency. Impressively, our CNN-LSTM model boasts the highest AUC at 0.972. This is closely followed by the CNN-BiLSTM and CNN-LSTM-GRU algorithms, registering AUC values of 0.965 and 0.951, respectively. On the other end of the spectrum, the DNN-LSTM lags behind, recording the lowest AUC at 0.831, suggesting its subpar efficacy in detecting network anomalies.

We have also determined values for FNR, FPR, FDR, and FOR, comparing our proposed methods with existing algorithms, as detailed in Figure 7. Table 3 reveals occasional misclassification of benign class samples. Additionally, our proposed method's TNR, MCC, and NPV metrics are depicted in Figure 8 and Table 4. With optimal values ranging between 90 and 95 for TNR, MCC, and NPV, it underscores the classifier's robust performance, making it apt for deployment in IIoT systems and networks for intrusion detection. Furthermore, Figure 9 illustrates the processing speed of our proposed method in comparison to other contemporary classifiers. Specifically, the CNN-LSTM processed 1000 samples in a mere 300 microseconds during testing. When we extended the experiment to various models, it provided insights into the performance dynamics of different deep learning classifiers. Notably, our CNN-LSTM-GRU algorithm outshines its counterparts in terms of time efficiency. The graph suggests that while the CNN-LSTM-GRU does have a slight trade-off, it remains competitive in testing time compared to recent algorithms.

Comparison of Techniques with Existing Techniques For a thorough assessment of our proposed method, we juxtaposed our results with those of benchmarked existing techniques. Table 5 provides a detailed comparative analysis, highlighting how our envisioned approach stacks up against leading-edge IoT intrusion detection systems tailored for industrial IoT.

Table 4: TNR, MCC, and NPV values of DNN-LSTM, CNN-LSTM, CNN-BiLSTM, and CNN-LSTM-GRU

Metrics	DNN-LSTM	CNN-LSTM	CNN-BiLSTM	CNN-LSTM-GRU
TNR	0.8531	0.9374	0.9542	0.9759
MCC	0.6578	0.9052	0.9329	0.9427
NPV	0.7748	0.9546	0.9694	0.9697

Table 5: Comparison of Algorithms

Work	Algorithms	Accuracy	Precision	Recall	F1-score
ours	CNN-LSTM-GRU	98.45%	98.46%	98.45%	98.45%
[24]	Cu-ConvLSTM2D	97.74%	98.11%	98.22%	98.22%
[24]	Hybrid(CNN-LSTM)	97.29%	97.25%	97.50%	97.29%

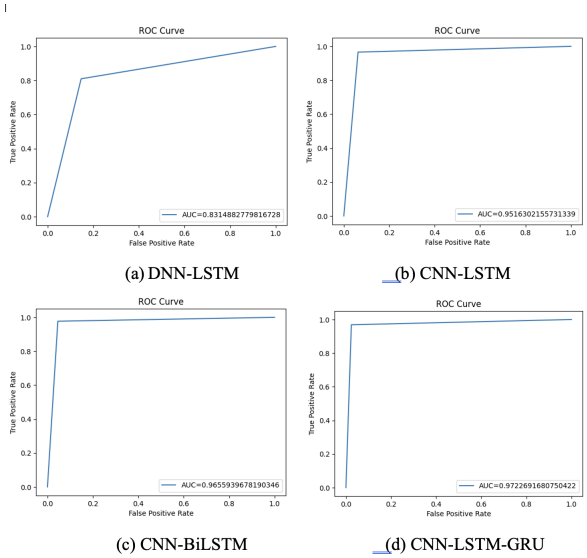


Figure 6: (ROC) curve elucidates the relationship between true-positive and false-positive rates

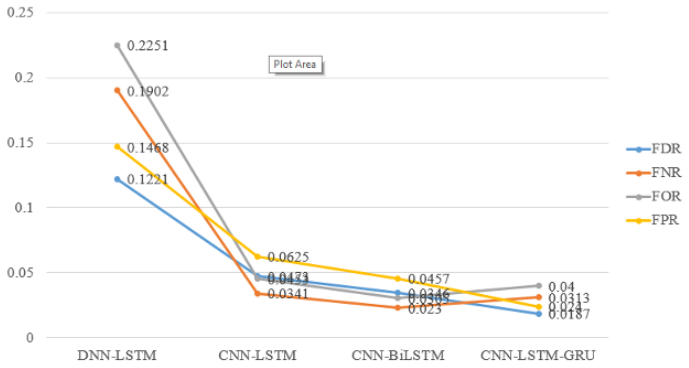


Figure 8: FDR, FNR, FOR and FPR values of DNN-LSTM, CNN-LSTM, CNN-BiLSTM and CNN-LSTMGRU

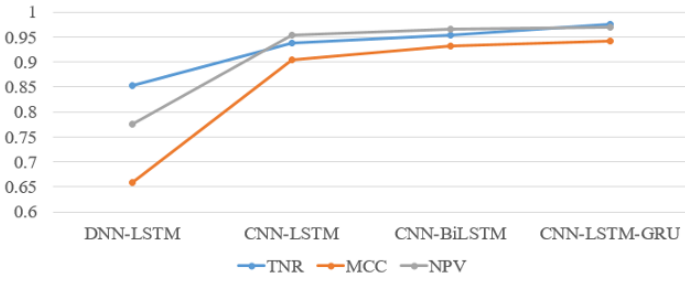


Figure 7: TNR, MCC and NPV values of DNN-LSTM, CNN-LSTM, CNN-BiLSTM and CNN-LSTM-GRU

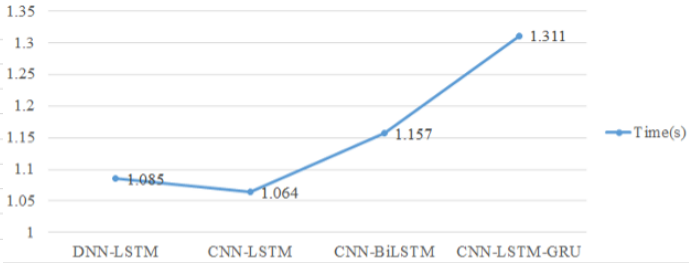


Figure 9: Testing Time of DNN-LSTM, CNN-LSTM, CNN-BiLSTM and CNN-LSTM-GRU

5 Conclusion

In this modern era of smart devices, the increased interconnectedness has inadvertently prepared the way for major cybersecurity concerns, particularly Distributed Denial of Service (DDoS) assaults. These kinds of attacks can take down a whole network by overwhelming it with requests for services. We used important measures such as accuracy, recall, precision, and F1-score in order to evaluate the adapted version of our solution that we had provided for this shifting environment. The hybrid deep learning solution that we presented, which integrated the strengths of CNN, LSTM, and GRU, displayed an impressive 98.45% detection rate when put to the test against IoT-centric issues. Beyond its comparative superiority to other leading detection procedures, our methodology highlights the potential of integrated architectures in strengthening threat detection in our interconnected digital world. This marks a pivotal contribution to our research and is one of the most important takeaways from it.

6 Funding Information

The work presented in this paper has been supported by Beijing Natural Science Foundation (No. IS23054).

References

[1] M. M. Alani, "Botstop: Packet-based efficient and explainable iot botnet detection using machine learning," *Computer Communications*, vol. 193, pp. 53–62, 2022.

[2] L. A. Aldossary, M. Ali, and A. Alasaadi, "Securing scada systems against cyber-attacks using artificial intelligence," in *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 739–745. IEEE, September 2021.

[3] M. A. Amanullah, R. A. A. Habeeb, F. H. Nasaruddin, A. Gani, E. Ahmed, A. S. M. Nainar, N. M. Akim, and M. Imran, "Deep learning and big data technologies for iot security," *Computer Communications*, vol. 151, pp. 495–517, 2020.

[4] J. B. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran, and R. K. Mehra, "Proactive detection of distributed denial of service attacks using mib traffic variables-a feasibility study," in *Proceedings of the 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No. 01EX470)*, pp. 609–622. IEEE, May 2001.

[5] D. Curran, "Surveillance capitalism and systemic digital risk: The imperative to collect and connect and the risks of interconnectedness," *Big Data & Society*, vol. 10, no. 1, p. 20539517231177621, 2023.

[6] M. A. Ferrag, L. Shu, H. Djallel, and K. K. R. Choo, "Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0," *Electronics*, vol. 10, no. 11, p. 1257, 2021.

[7] S. Gallacher, D. Wilson, A. Fairbrass, D. Turmukhambetov, M. Firman, S. Kreitmayer, O. Mac Aodha, G. Brostow, and K. Jones, "Shazam for bats: Internet of things for continuous real-time biodiversity monitoring," *IET Smart Cities*, vol. 3, no. 3, pp. 171–183, 2021.

[8] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for internet of things," *Computer Networks*, vol. 186, p. 107784, 2021.

[9] N. Gupta, V. Jindal, and P. Bedi, "Lio-ids: Handling class imbalance using lstm and improved one-vs-one technique in intrusion detection system," *Computer Networks*, vol. 192, p. 108076, 2021.

[10] S. Hosseini and M. Azizi, "The hybrid technique for ddos detection with supervised learning algorithms," *Computer Networks*, vol. 158, pp. 35–45, 2019.

[11] R. H. Hwang, M. C. Peng, C. W. Huang, P. C. Lin, and V. L. Nguyen, "An unsupervised deep learning model for early network traffic anomaly detection," *IEEE Access*, vol. 8, pp. 30387–30399, 2020.

[12] T. Khempetch and P. Wuttidittachotti, "Ddos attack detection using deep learning," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, p. 382, 2021.

[13] M. Di Mauro, G. Galatro, G. Fortino, and A. Liotta, "Supervised feature selection techniques in network intrusion detection: A critical review," *Engineering Applications of Artificial Intelligence*, vol. 101, p. 104216, 2021.

[14] N. Mishra and S. Pandya, "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review," *IEEE Access*, vol. 9, pp. 59353–59377, 2021.

[15] M. Mudassir, D. Unal, M. Hammoudeh, and F. Azzedin, "Detection of botnet attacks against industrial iot systems by multilayer deep learning approaches," *Wireless Communications*.

[16] A. Nazir, J. He, N. Zhu, A. Wajahat, X. Ma, F. Ullah, S. Qureshi, and M. S. Pathan, "Advancing iot security: A systematic review of machine learning approaches for the detection of iot botnets," *Journal of King Saud University-Computer and Information Sciences*, p. 101820, 2023.

[17] A. Pektaş and T. Acarman, "Deep learning to detect botnet via network flow summaries," *Neural Computing and Applications*, vol. 31, pp. 8021–8033, 2019.

[18] R. Priyadarshini and R. K. Barik, "A deep learning based intelligent framework to mitigate ddos attack in fog environment," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 3, pp. 825–831, 2022.

[19] L. L. Puryear. *A quantitative correlational study of malicious software (malware) identification and security practitioners' training*. PhD thesis, University of Phoenix, 2019.

[20] S. Qureshi, J. He, S. Tunio, N. Zhu, F. Ullah, A. Nazir, and A. Wajahat, "An adaptive multi-layer architecture for iot based idps for attacks using deep learning method," *International Journal of Network Security*, vol. 24, no. 5, pp. 815–827, 2022.

[21] S. Qureshi, J. He, S. Tunio, N. Zhu, F. Akhtar, F. Ullah, A. Nazir, and A. Wajahat, "A hybrid DL-based detection mechanism for cyber threats in secure networks," *IEEE Access*, vol. 9, pp. 73938-73947, 2021.

[22] S. Qureshi, J. He, S. Tunio, N. Zhu, F. Ullah, A. Nazir, and A. Wajahat, "Analysis distributed denial-of-service attack deploy deep learning techniques," *International Journal of Network Security*, vol. 25, no. 5, pp. 745–757, 2023.

[23] S. S. Qureshi, J. He, N. Zhu, Z. A. Zardari, T. Mahmood, and A. Wajahat, "Sdn-enabled deep learning based detection mechanism (ddm) to tackle ddos attacks in iots," *Journal of Intelligent & Fuzzy Systems*, vol. 44, no. 6, pp. 10675–10687, 2023.

[24] S. Rao, A. K. Verma, and T. Bhatia, "A review on social spam detection: Challenges, open issues, and future directions," *Expert Systems with Applications*, vol. 186, p. 115742, 2021.

[25] M. M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in iot: A survey," *The Journal of Supercomputing*, vol. 76, pp. 5320–5363, 2020.

[26] S. Salmi and L. Oughdir, "Performance evaluation of deep learning techniques for dos attacks detection in wireless sensor network," *Journal of Big Data*, vol. 10, no. 1, pp. 1–25, 2023.

[27] M. Shafiq, Z. Gu, O. Cheikhrouhou, W. Alhakami, and H. Hamam, "The rise of "internet of things": Review and open research issues related to detection and prevention of iot-based security attacks," *Wireless Communications and Mobile Computing*, pp. 1–12, 2022.

[28] Y. Wei, J. Jang-Jaccard, F. Sabrina, W. Xu, S. Camtepe, and A. Dunmore, "Reconstruction-based lstm-autoencoder for anomaly-based ddos attack detection over multivariate time-series data," *arXiv preprint arXiv:2305.09475*, 2023.

Biography

SAIMA SIRAJ QURESHI received the BSIT(Hons) with gold medal from Sindh Agriculture University Tandojam, Pakistan. Afterwards, she pursued her MSIT from Isra University Hyderabad, Pakistan. Currently, she is pursuing PhD in Information Technology at the Beijing University of Technology, China. She has more than five research publications to her credit as main author and co-author, which featured national and international journals and conferences. Saimas research areas include but not limited to Information security. IoT security, Digital Forensics, Cyber Security, Computer Networks.

JINGSHA HE received a bachelor's degree in computer science from Xi'an Jiaotong University, China, and the master's and Ph.D. degrees in computer engineering from the University of Maryland, College Park, MD, USA. He worked for several multinational companies in USA, including IBM Corp., MCI Communications Corp., and Fujitsu Laboratories. He is currently a Professor with the Faculty of Information Technology, Beijing University of Technology(BJUT), Beijing. He has published more than ten articles. He holds 12 U.S. patents. Since August 2003, he has been published over 300 papers in scholarly journals and international conferences. He also holds over 84 patents and 57 software copyrights in China and authored nine books. He was a principal investigator of more than 40 research and development projects. His research interests include information security, wireless networks, and digital forensics.

NAFEI ZHE received the B.S. and M.S. degrees from Central South University, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science and technology from the Beijing University of Technology, Beijing, China, in 2012. She was a Postdoctoral Research Fellow with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, from 2015 to 2017. She is currently an Associate Professor at the Faculty of Information Technology, Beijing University of Technology. She has published over 20 research papers in scholarly journals and international conferences. Her research interests include information security and privacy, wireless communications, and network measurement.

Min Jia (Senior Member, IEEE) received the M.Sc. degree in information and communication engineering from the Harbin Institute of Technology (HIT), Harbin, China, in 2006, and the Ph.D. degree from Sungkyunkwan University and HIT in 2010.,She is currently a Professor and a Ph.D. Supervisor with the School of Electronics and Information Engineering, HIT. Her research interests include advanced mobile communication technology for LTE and 5G, cognitive radios, digital signal processing, and advanced broadband satellite communication systems. She is a member of the Steering Committee of the WiSATs International Conference. She has won six best paper awards at several international conferences. She is also the Winner of the Science Fund for Excellent Young Scholars for Heilongjiang Province. She is the General Chair of the IEEE GLOBECOM 2019 Workshop Intelligent and Cognitive Space, Terrestrial and Ocean Internet, Systems and Applications.

SIRAJUDDIN QURESHI received his bachelor degree in Computer Sciences from Quaid-e-Awam University of Engineering, Science & Technology, Pakistan. Afterwards, he pursued his Master's in Information Technology from Sindh Agricultural University Tan-

dojam,Pakistan. Currently, he is pursuing PhD in Information Technology at the Beijing University of Technology, China. He has nine research publications to his credit as main author and co-author, which featured national and international journals and conferences. Sirajuddin research areas include but are not limited to Network Forensics Analysis, Digital Forensics, Cyber security, Computer Networks and Network Security.

FAHEEM ULLAH received M.S degrees from the Xian Jiaotong University, China,in 2017. He is currently pursuing a Ph.D. degree at the Beijing University of Technology, Beijing, China. His research interests include information security, Blockchain, and data mining. He has received awards and honors from the China Scholarship Council (CSC), China.

AHSAN NAZIR has received his M.Sc degree from the University of Engineering and Technology Lahore in 2016. From September 2015 to August 2018 he worked as Software Engineer at Dunya Media group Lahore since September 2018 he is doing Ph.D. in Software Engineering from Beijing University of Technology, Beijing China . He has published more than 10 journals and conference papers .His area of research include eGovernment,IoT, Software Engineering and Machine learning applications .

AHSAN WAJAHAT received the B.S. and M.S degrees in information technology from the Sindh Agriculture University, Pakistan in 2012 and 2016, respectively. He is currently pursuing a Ph.D. degree at the Beijing University of Technology, Beijing, China. His research interests include machine learning, information security, forensic networks and data mining. He has received awards and honors from the China Scholarship Council (CSC), China.

Efficient Safety Vector Computation and Its Popularization

Hui Xia[1], Chunhua Wang[2#], Lantao You[2], and Weiji Yang[3]
(Corresponding author: Weiji Yang)
[#] *Hui Xia and Chunhua Wang contribute equally to the article.*
Shenyang Normal University, Shenyang 110034, China[1]
Suzhou Industrial Park Institute of Service Outsourcing, Suzhou 215123, Jiangsu, China[2]
Zhejiang Chinese Medical University, HangZhou 310000, China[3]
Email: yangweiji@163.com
(Received Nov. 17, 2022; Revised and Accepted Sept. 12, 2023; First Online Apr. 25, 2024)

Abstract

Secure multi-party computation is an important cryptography research direction and a hotspot in international cryptography. Because vectors can describe many practical problems, studying the vector secret computation is of great theoretical and practical significance. Currently, most vector secret computation problems are studied on the integer set. There is little research on the vector problem of rational numbers. This paper focuses on the secure multi-party computation of vectors over the rational number field, including vector dot product, vector equality, and superiority. A secure and efficient computation protocol is designed, expanding the vector secure computation application scope. The security analysis and efficiency analysis of the protocol in this paper shows that the protocol in this paper has apparent advantages over existing protocols in terms of security and efficiency. Finally, new vector and computational geometry problems are solved using the designed protocol.

Keywords: Cryptography; Secure Multi-Party Computing; Vector Advantage

1 Introduction

The rapid development of information technology has brought mankind into the information society. The use of the Internet to obtain information, exchange information, and conduct joint computing has become a very important feature of the information society. Accordingly, the information security problem is becoming increasingly severe, making information security a research hotspot in information science. Information sharing integrates information acquisition, transmission, processing, and utilization, and people pay special attention to information security in information sharing. Secure Multiparty Computing (SMC) is a key technology to realize private information sharing and is a research hotspot in the international cryptography field. In 1982, Yao Qizhi [2] proposed the secure multiparty computing problem for two participants. In 1988, Ben and Goldwasser [1] introduced the secure multiparty computing problem for multiple participants and conducted in-depth research on it.

Secure multiparty computing is about a group of participants who do not trust each other to perform collaborative computing to protect their private data. SMC should ensure the privacy of each participant's input data and the calculation results' correctness. Cramer [15], an internationally famous computer scientist and cryptographer, pointed out that if any function can be calculated confidentially, Computing science has a new powerful tool. Goldreich*et al.* [4, 8] laid the theoretical foundation for secure multiparty computing. They proved that all secure multiparty computing problems are theoretically solvable and proposed a universal solution. However, due to efficiency, It is impractical to apply general solutions to a specific secure multiparty computing problem. It is a way to solve practical problems by studying specific solutions according to specific problems' characteristics. In recent years, many cryptography researchers have constantly proposed new secure multiparty computing problems with practical application prospects and studied their solutions, which has promoted the development of secure multiparty computing research. At present, the main research issues are a comparison of confidential information [16, 17], Confidential data mining [7, 11, 18], confidential geometric calculation [5, 10, 14], confidential database query [13], confidential auction [3], etc.

At present, the vector dot product problem is the most widely studied Literature [12] designed a vector dot product protocol based on an ideal preprocessing process, but the implementation of the preprocessing phase requires a third-party participant; Literature [6] designed a vector dot product protocol for computing in a multi-key cloud environment, which has high computational complexity;

Literature [9] designed two dot product protocols based on the third-party server and the invertible matrix. In the protocol based on the invertible matrix, each participant will disclose n/2 linear relations of its n-dimensional vector (when the vector dimension or data range is small, the protocol has certain security risks), and the calculation of the inverse matrix is required, resulting in high computing costs; Literature [6] designed a point product protocol based on polynomial sharing. Because the protocol needs to employ a semi-honest third party, the protocol is vulnerable to collusion attacks; Literature [13] designed an even dimension dot product protocol based on literature [3]. The contributions of this paper are as follows:

1) A simple and efficient vector dot product protocol is designed based on basic algebra knowledge, and vector equality and vector dominance secrecy determination protocol is designed on this basis.

2) The protocol designed in this paper only needs a basic arithmetic operation and does not use any public key encryption scheme, so it has high computational efficiency. Simulation examples are used to prove that the protocol is secure under the semi-honest model. Theoretical analysis and experimental results show that the proposed protocol has higher security and computational efficiency than the existing protocols.

3) The protocol in this paper applies to the calculation of rational number vectors and has broad applicability. Examples are given to illustrate the design idea of this protocol and to design and construct safe and efficient solutions for more extensive vector computation and other practical application problems.

2 Related Work

This section introduces some basic concepts and notations related to this paper. The content of this part is basically taken from literature [8].

Bidirectional calculation: A bidirectional calculation is a random process of mapping any given input pair to an output pair, expressed as $f : (x, y) \to (f1(x, y), f2(x, y))$. That is, for each input pair (x, y), the output pair is a random variable $(f1(x, y), f2(x, y))$. Let's call this $f = (f1, f2)$.

Semi-honest model: The so-called semi-honest participants refer to those participants who faithfully perform the agreement in accordance with the requirements of the agreement during the implementation process, but they may record all the information collected during the implementation of the agreement and try to calculate the input of other participants based on the recorded information after the implementation of the agreement. If all the participants are semi-honest participants, such a computational model is called a semi-honest model. Semi-honest participants do not perform active attacks on agreements, so semi-honest models are also known as honest-but-curious models or passive models. This article assumes that all participants are semi-honest.

Suppose the two parties involved in the calculation are P_1 and P_2. Let $f = (f1, f2)$ be a probabilistic polynomial time function, representing a two-party protocol for computing function F. When the input of $P_i(i = 1, 2)$ is x_i, the sequence of information obtained by P_i during the implementation of the protocol π is denoted as:

$$view_i^{\pi}(x_1, x_2) = (x_i, r^i, m_1^i, \cdots, f_i(x_1, x_2))$$

Where r^i represents the random number generated by P_i, m_j^i represents the j^{th} message received by P_i, and $f_i(x_1, y_2)$ represents the output result obtained by P_i.

Definition 1. *(Protocol security under semi-honest model): For the above function f and protocol π, if there are probabilistic polynomial time algorithms S_1 and S_2, so that*

$$\{S_1(x_1, f_1(x_1, x_2))\}_{x_1,x_2} \equiv^c \{view_1^{\pi}(x_1, x_2)\}_{x_1,x_2}$$
$$\{S_2(x_1, f_1(x_1, x_2))\}_{x_1,x_2} \equiv^c \{view_2^{\pi}(x_1, x_2)\}_{x_1,x_2}$$

If the above two expressions are true, it is said that the agreement π has calculated function f confidentially, where the function f is a composite function that depends on the protocol π. It means that the calculation is indistinguishable.

To prove that a two-party secure computing protocol is secure, simulators S_1 and S_2 that satisfy the above two formulae must be constructed. In the execution of the protocol, if a participant does not get any output, it is agreed that the output of the participant is an empty string λ.

3 Efficient Vector Dot Product Secure Computing Protocol

Definition 2. *Suppose that two participants, Alice and Bob, have private n dimensional vectors $X = (x_1, \cdots, x_n)$ and $Y = (y_1, \cdots, y_n)$ respectively, and they wish to cooperate in secretly computing the dot product of the two vectors $XY = x_1y_1 + \cdots + x_ny_n$ if the output result is X and Y, the protocol is called vector dot product protocol. If, at the end of the agreement, one party gets $S \neq 0$ and the other party gets $sX \cdot Y$ or $X \cdot Y + S$, we call such an agreement a shared vector dot product coagreement.*

At present, the main problem existing in the vector dot product (or shared dot product) problem is that the protocols based on public key encryption scheme have good security but high complexity, while most of the protocols

that avoid using public key encryption system have different degrees of information leakage. Below we first apply the basic algebra knowledge and certain design skills to design an efficient and safe vector dot product (sharing) protocol. In the following context, a vector is called a rational number vector if all its components are rational.

We first construct an efficient secret computing protocol for shared vector dot product and prove its security.

Protocol 1. Shared vector dot product confidential computing protocol.

Input: Alice inputs the rational number vector $X = (x_1, \cdots, x_n)$, Bob inputs the rational number vector $Y = (y_1, \cdots, y_n)$.

Output: Alice outputs S,Bob outputs $F(X, Y) = sX \cdot Y$.

Step:

1) Alice decomposes vector $X = (x_1, \cdots, x_n)$ in the following way: Randomly selects rational number ai and rational number vector $X = (x_{i_1}, \cdots, x_{i_n})(i \in [1, t] = \{1, \cdots, t\}, 2 \leq t \leq n+1)$, such that $X = a_1X_1 + \cdots + a_tX_t$ and $a_1 + \cdots + a_t \neq 0$. Alice sends the vector $X_1, \cdots, X_t$ to Bob.
2) Bob randomly selects numbers or vectors:
   a. Bob randomly selects rational number b_j and rational number vector $Y_j = (y_{j_1}, \cdots, y_{j_n})(j = 1, 2)$, so that $Y = b_1Y_1 + b_2Y_2$.
   b. Bob selects non-zero random rational number k_1, k_2, r_1, r_2, and calculates

$$\begin{aligned} z_{11} &= k_1X_1Y_1 + r_1, \cdots, z_{1t} = k_1X_tY_1 + r_1 \\ z_{21} &= k_2X_1Y_2 + r_2, \cdots, z_{2t} = k_2X_tY_2 + r_2 \end{aligned}$$

   Send (z_{11}, z_{1t}) and (z_{21}, z_{2t}) to Alice.
3) Alice calculates

$$\begin{aligned} z_1 &= s(a_1z_{11} + \cdots + a_tz_{1t}) \\ z_2 &= s(a_1z_{21} + \cdots + a_tz_{2t}) \end{aligned}$$

   Where $s = \frac{1}{a_1+\cdots+a_t}$. And sends z_1, z_2 to Bob.
4) Bob calculates

$$z = b_1\frac{(z_1 - r_1)}{k_1} + b_2\frac{(z_2 - r_2)}{k_2}$$

5) Alice sends s, Bob sends z.

Correctness of Protocol 1. We just have to prove that $z = sX \cdot Y$ true.

According to the operation properties of the inner product, for any rational number vector X and Y, and for any decomposition of X and Y:

$$\begin{aligned} X &= a_1X_1 + \cdots + a_tX_t \\ Y &= b_1Y_1 + b_2Y_2 \end{aligned}$$

The following equation holds:

$$\begin{aligned} z_1 &= s(a_1z_{11} + \cdots + a_tz_{1t}) = sk_1X \cdot Y_1 + r_1 \\ z_2 &= s(a_1z_{21} + \cdots + a_tz_{2t}) = sk_2X \cdot Y_2 + r_2 \end{aligned}$$

So

$$\begin{aligned} z &= b_1\frac{(z_1 - r_1)}{k_1} + b_2\frac{(z_2 - r_2)}{k_2} \\ &= b_1(sX \cdot Y_1) + b_2(sX \cdot Y_2) \\ &= sX \cdot (b_1Y_1 + b_2Y_2) \\ &= sX \cdot Y \end{aligned}$$

Therefore, Agreement 1 is correct.

Security of Protocol 1. To analyze the security of Protocol 1, it is necessary to examine the security of each participant's private vector during the protocol execution and analyze in detail the potential information that may be inferred after the protocol execution.

Firstly, the security of Alice vector is considered. Alice secretly factorizes X into $X = a_1X_1 + \cdots + a_tX_t$, During the implementation of the whole protocol, Bob gets the $X_1, \cdots, X_t$ and z_1, z_2 sent to him by Alice. Bob can obtain the corresponding equations according to the decomposition formula $X = a_1X_1 + \cdots + a_tX_t$,

$$\begin{cases} x_1 = a_1x_{11} + \cdots + a_tx_{t1} \\ \cdots \\ x_n = a_1x_{1n} + \cdots + a_tx_{tn} \end{cases} \quad (1)$$

When $t < n$, a linear relation between $t+1$ components of X vector can be obtained from the simultaneous elimination of $a_1, \cdots, a_t$ in Equation (1).

When $t = n$ (or $t = n+1$), the n equations of System (1) contain $2n$ (or $2n+1$) unknowns $x_1, \cdots, x_n$ and $a_1, \cdots, a_t$. It can be seen from the solving process of the linear system that, It is necessary for $n+1$ (or $n+2$) equations to be simultaneous to eliminate the secret data $a_1, \cdots, a_t$ and obtain a linear relation between $n+1$ (or $n+2$) components of the X vector. So when $t = n$ (or $t = n+1$), Bob can't get any information about Alice vector X according to Equation (1).

For Bob, according to $z_1 = sk_1X \cdot Y_1 + r_1$, $z_2 = sk_2X \cdot Y_2 + r_2$, there are:

$$\begin{cases} \frac{z_1 - r_1}{k_1} = sX \cdot Y_1 = s(x_1y_{11} + \cdots + x_ny_{1n} \\ \frac{z_2 - r_2}{k_2} = sX \cdot Y_2 = s(x_1y_{21} + \cdots + x_ny_{2n} \end{cases} \quad (2)$$

Since the two equations in System (2) contain $n+1$ unknowns, Bob can obtain by simultaneous equations:

$$\frac{x_1y_{11} + \cdots + x_ny_{1n}}{x_1y_{21} + \cdots + x_ny_{2n}} = l \quad (3)$$

So $x_1(y_{11} - ly_{21}) + \cdots + x_n(y_{1n} - ly_{2n}) = 0$.

Further, when Equations (1) and (2) are combined, Bob can obtain a linear relation about $a_1, \cdots, a_t$ at:

$$\begin{aligned} &a_1[x_{11}(y_{11} - ly_{21} + \cdots + x_{1n}(y_{1n} - ly_{2n})] + \cdots \\ &+a_t[x_{t1}(y_{t1} - ly_{21} + \cdots + x_{tn}(y_{1n} - ly_{2n})] = 0 \quad (4) \end{aligned}$$

Equation (4) and Equation (1) are combined to obtain an $n+1$ equations with $x_1, \cdots, x_t$ and $a_1, \cdots, a_t$ as unknowns. Therefore, when $t < n$, Bob is squared by Equation (4) Group (1) $n+1$ equations are simultaneous, and linear relations between t components of X vector can be obtained at most. When $t = n$ (or $t = n+1$), Bob can only get To a linear relation (3) about vector X. In summary, Alice vector X is safe, and the security of vector X is proportional to t value.

Now consider the security of Bob vectors. In the execution of the protocol, Alice only gets the information sent by Bob $(z_{1_1}, \cdots, z_{1_t})$ and $(z_{2_1}, \cdots, z_{2_t})$, namely:

$$\begin{cases} z_{11} = k_1(x_{11}y_{11} + \cdots + x_{1n}y_{1n}) + r_1 \\ z_{1t} = k_1(x_{t1}y_{t1} + \cdots + x_{tn}y_{1n}) + r_1 \end{cases} \tag{5}$$

and

$$\begin{cases} z_{21} = k_2(x_{11}y_{21} + \cdots + x_{1n}y_{2n}) + r_2 \\ z_{2t} = k_2(x_{t1}y_{21} + \cdots + x_{tn}y_{2n}) + r_2 \end{cases} \tag{6}$$

When $t = n$, Alice assumes that the matrix $A = (X_1, \cdots, X_n)$ has an invertible matrix A^{-1}, by calculating

$$\begin{aligned} (z_{11}, \cdots, z_{1n})A_{-1} &= k_1Y_1 + r_1(1, \cdots, 1)A^{-1} \\ (z_{21}, \cdots, z_{2n})A_{-1} &= k_2Y_2 + r_2(1, \cdots, 1)A^{-1} \end{aligned}$$

Get

$$\begin{aligned} Y &= \frac{b_1}{k_1}(k_1 \cdot Y_1) + \frac{b_2}{k_2}(k_2 \cdot Y_1) \\ &= \frac{b_1}{k_1}[(z_{11}, \cdots, z_{1n})A^{-1} - r_1(1, \cdots, 1)A^{-1}] \\ &\quad + \frac{b_2}{k_2}[(z_{21}, \cdots, z_{2n})A^{-1} - r_2(1, \cdots, 1)A^{-1}] \end{aligned} \tag{7}$$

Where $r_1, r_2, \frac{b_1}{k_1}$, and $\frac{b_2}{k_2}$ are private data of Bob, Equation (7) contains n equations and $n+4$ unknowns, so Alice cannot determine Bob vector Y according to Equation (7). However, when $n > 4$, Alice can obtain the linear relationship between the four components of vector Y.

When $t < n$ (or $t = n+1$), first of all, there is no reversible matrix in matrix $A = (X_1, \cdots, X_t)$, then the relation (7) does not exist, and there is no way to eliminate Bob's private random number $r_1, r_2, \frac{b_1}{k_1}$, and $\frac{b_2}{k_2}$, so Alice cannot get Bob Vector Y. Secondly, the system (5) (or (6)) has t equations, but contains $n+2$ unknowns, $y_{1_1}, \cdots, y_{1_n}$ and k_1, t_1 (or $y_{2_1}, \cdots, y_{2_n}$ and k_2, t_2). Since $n+2 > t$ and all the unknowns are rational numbers, So even though Alice With infinite computing power, it is also impossible to directly obtain vector Y_1 (or Y_2) by solving Equation (5) or Equation (6), and thus cannot obtain vector Y. but Equation (5) can obtain the following relation (8).

Equation (8) is a linear system of equations with $t-2$ equations and $y_{1_1}, \cdots, y_{1_n}$ as the unknown quantity. Because $t-2 < n$, Alice Linear relations of any $n-t+3$ components of vector Y_1 can be obtained at most (similarly, Linear relations of any $n-t+3$ components of vector Y_2 can be obtained by Alice at most). Further, it can be seen from $Y = b_1Y_1 + b_2Y_2$ that Alice needs to combine vector Y_1 and Y_2 to solve vector Y. However, Alice only knows the independent linear relation of each self-component of vector Y_1 and Y_2, and does not know the value of random number b_1 and b_2, so it cannot solve them simultaneously, so it cannot obtain any information of vector Y. To sum up, when $t < n$ (or $t = n+1$), Alice can not get the information of Bob vector Y, and vector Y is safe.

$$\begin{cases} \frac{z_{13}-z_{11}}{(}z_{12} - z_{11} = \frac{(x_{31}-x_{11}y_{11}+\cdots+(x_{3n}-x_{1n}y_{1n}}{(x_{21}-x_{11}y_{11}+\cdots+(x_{2n}-x_{1n}y_{1n}} \\ \frac{z_{1t}-z_{11}}{(}z_{12} - z_{11} = \frac{(x_{t1}-x_{11}y_{11}+\cdots+(x_{tn}-x_{1n}y_{1n}}{(x_{21}-x_{11}y_{11}+\cdots+(x_{2n}-x_{1n}y_{1n}} \end{cases} \tag{8}$$

According to the above analysis, when $t < n$, Bob vector Y is safe, Bob can get the relations between t components of vector X at most; When $t = n$, Alice can get the relationship between 4 components of vector Y at most, while Bob can only get one relation (3) about vector X. When $t = n+1$, Bob vector Y is safe and Bob can only get one relation (3) about vector X. Therefore, when $t = n+1$, Protocol 1 has the best security, and there is only a slight difference compared with the protocol under the ideal model: Bob can get a linear relation of Alice vector X.

The security of Protocol 1 has the following theorem.

Theorem 1. *Shared vector dot product Protocol 1 is secure.*

Proof. The following is a rigorous proof of theorem 1 using a simulation example, that is, a simulator S_1 (or S_2) needs to be constructed for both equations to be true. S_1 is constructed first. After receiving input (X, s), S_1 runs as follows:

1) S_1 firstly randomly selects the rational number vector $Y' = (y'_1, \cdots, y'_n)$, and randomly selects the rational number b'_i and the rational number vector $Y'_j = (y'_{j1}, \cdots, y'_{jn})$ $(j = 1, 2)$, so that $Y' = b'_1Y'_1 + b'_2Y'_2$.

2) S_1 Randomly select non-zero random rational numbers k'_1, k'_2, r'_1, r'_2, and calculate

$$\begin{aligned} z'_{11} &= k'_1X_1 \cdot Y'_1 + r'_1, \\ &\cdots \\ z'_{1t} &= k'_1X_t \cdot Y'_1 + r'_1 \\ z'_{21} &= k'_2X_1 \cdot Y'_2 + r'_2, \\ &\cdots \\ z'_{2t} &= k'_2X_t \cdot Y'_2 + r'_2 \end{aligned}$$

Because in the execution of the agreement,

$$view'_1(X, Y) = (X, (z_{11}, \cdots, z_{1t}), (z_{21}, \cdots, z_{2t}), s)$$

While the information sequence generated by S_1 in the simulation process is

$$S_1(X, f_1(X, Y)) = (X, (z'_{11}, \cdots, z'_{1t}), (z'_{21}, \cdots, z'_{2t}), s)$$

Because of rational numbers b_j, k_1, k_2, r_1, r_2 and rational number vector $Y_j = (y_{j1}, \cdots, y_{jn})(j = 1, 2)$ are chosen by Bob at random, for Alice, has

$$\begin{aligned} b_j &\equiv^c b'_j \\ Y_j &\equiv^c Y'_j \\ k_1 &\equiv^c k'_1 \\ k_2 &\equiv^c k'_2, \\ r_1 &\equiv^c r'_1, \\ r_2 &\equiv^c r'_2 \end{aligned}$$

Therefore $(z_{11}, \cdots, z_{1t}) \equiv^c (z'_{11}, \cdots, z'_{2t})$, $(z_{21}, \cdots, z_{2t}) \equiv^c (z'_{21}, \cdots, z'_{2t})$ so

$$\{S_1(X, f_1(X, Y))\}_{x_u, y_i \in Q} \equiv^c \{view_1^{\pi}(X, Y)\}_{x_i, y_i \in Q}$$

After receiving input $(Y, f_2(X, Y) = F(X, Y))$, run S_2 as following:

First of all, S_2 chooses any rational number vector $x' = (x'_1, \cdots, x'_n)$, and choose rational numbers a'_i and rational number vector $X'_i = (x'_{i1}, \cdots, x'_{in})(i \in [1, t] = \{1, cdots, t\}, 2 \le t \le n + 1)$ at random. Making $F(X', Y) = F(X, Y)$, $X' = a'_1 X'_1 + \cdots + a'_t X'_t$ and $a'_1 + \cdots + a'_t \neq 0$. S_2 calculates

$$\begin{aligned} z'_{11} &= k'_1 X_1 \cdot Y'_1 + r'_1, \\ &\cdots \\ z'_{1t} &= k'_1 X_t \cdot Y'_1 + r'_1 \\ z'_{21} &= k'_2 X_1 \cdot Y'_2 + r'_2, \\ &\cdots \\ z'_{2t} &= k'_2 X_t \cdot Y'_2 + r'_2 \end{aligned}$$

and

$$\begin{aligned} z'_1 &= s'(a'_1 z'_{11} + \cdots + a'_t z'_{1t}), \\ z'_2 &= s'(a'_1 z'_{21} + \cdots + a'_t z'_{2t}) \end{aligned}$$

Wherein $s' = \frac{1}{a'_1 + \cdots + a'_t}$

3) S_2 calculates

$$z' = b_1 \frac{z'_1 - r_1}{k_1} + b_2 \frac{z'_2 - r_2}{k_2}$$

Due to the implementation of the agreement,

$$view_2^{\pi}(X, Y) = (Y, (X_1, \cdots, X_t), (z_1, z_2), F(X, Y)).$$

And the sequence of information S_2 generated during the simulation is

$$S_2(Y, f_2(X, Y)) = (Y, (X'_1, \cdots, X'_t), (z'_1, z'_2), F(X', Y)).$$

First ,becaues of a random number that $a_i(i \in [1, t])$ picks for Alice, so, $a'_i \equiv^c a_i$, $s'_i \equiv^c s$, therefore $(X'_1, \cdots, X'_t) \equiv^c (X_1, \cdots, X_t)$, $(z'_1, z'_2) \equiv^c (z_1, z_2)$. Because $F(X', Y) = F(X, Y)$, so $\{S_2(Y, f_2(X, Y))\}_{x_i, y_i \in Q} \equiv^c \{view_2^{\pi}(X, Y)\}_{x_i, y_i \in Q}$.

□

Annotation 1. *Assumption $t = n + 1$, if Alice takes $s = 1$, in Protocol 1, in other words $a_1 + \cdots a_t = 1$, the protocol output is $F(X, Y) = X \cdot Y$, Protocol 1 becomes a dot product protocol. At this point, at this point, the data z_1, z_2 that Bob received from Alice no longer contains the unknown number s, Bob can get at most two relationships:*

$$\begin{aligned} \frac{z_1 - r_1}{k_1} &= X \cdot Y_1 \\ &= x_1 y_{11} + \cdots + x_n y_{1n}, \\ \frac{z_2 - r_2}{k_2} &= X \cdot Y_2 \\ &= x_1 y_{21} + \cdots + x_n y_{2n} \end{aligned}$$

And if Equation (1) and $a_1 + \cdots + a_t = 1$ are set together, no information about vector X can be obtained, so Alice vector X is still safe. In this process, the data $z_{11}, \cdots, z_{1t})$ and $z_{21}, \cdots, z_{2t})$ that Alice received from Bob did not change, so the security of vector Y was not affected.

Annotation 2. *Because of the idea of vector decomposition, the vectors of participants in Protocol 1 need to be decomposed into at least two random vectors, so it can be known that the vector dot product problem of dimension $n \ge 3$ can be solved directly by using Protocol 1 in this paper. For the vector dot product problem of dimension $n = 2$, Protocol 1 is used, Bob still gets only a linear relation about Alice vector X; Alice can't get rid of the random numbers k_1, r_1 (or k_2, r_2) based on (z_{11}, z_{12}) (or z_{21}, z_{22})), so Alice can't get any information about vector Y_1, Y_2, so she can't get any information about vector Y. Therefore, Protocol 1 in this paper is suitable for the vector dot product problem of $n \ge 2$.*

If the vector dimension involved in calculation is small and each component is limited to 0 or 1, Protocol 1 in this paper cannot safely solve the vector dot product problem.

For example, when $n = 3$ and $x_i, y_i \in \{0, 1\}, i = 1, 2, 3$, there are only eight possible values of $X = (x_1, x_2, x_3)$. Bob's (3) has three unknowns x_1, x_2, x_3, Some values of (x_1, x_2, x_3) satisfy Equation (3) and some do not, by directly substituting 0 and1 for tests, thus giving away some information about vector X. The shared dot product protocol based on reversible matrix design in literature [3] also has similar security problems. When the vector dimension n is small and the values of the restricted components are integers, either party can derive enough private information of the other party through the $n/2$ equations obtained by it.

4 Vector Equality Security Determination Problem

Alice and Bob have rational number vectors $X = (x_1, \cdots, x_n)$ and $Y = (y_1, \cdots, y_n)$ respectively, They want to decide confidentially whether the two vectors X

and Y are equal. If they are not equal, they should not disclose any information about the vectors X or Y to each other.

Calculation principle:
First, prove the following.

Proposition 1. *For any two rational vectors $X = (x_1, \cdots, x_n)$ and $Y = (y_1, \cdots, y_n)$, $X = Y$ only if the following equation is true:*

$$|X|^2 + |Y|^2 = 2X \cdot Y. \tag{9}$$

Among $|X|^2 = x_1^2 + \cdots + x_n^2$, $|Y|^2 = y_1^2 + \cdots + y_n^2$.

Proof. Because of

$$\begin{aligned} X = Y &\iff |X - Y| = 0 \\ &\iff (x_1^2 + \cdots + x_n^2) + (y_1^2 + \cdots + y_n^2) - 2(x_1 y_1 + \cdots + x_n y_n) = 0 \\ &\iff |X|^2 + |Y|^2 = 2X \cdot Y \end{aligned}$$

So Proposition 1 is proved. □

Proposition 1 is the basic principle of determining whether the two vectors are equal, that is, the problem of determining whether the two vectors X and Y are equal is transformed into the problem of determining whether Condition (9) is valid. Next, we construct vector equality confidentiality determination protocol based on Protocol 1,Alice and Bob call Protocol 1 (convention $T = n+1$), Alice gets the random number $s > 2$, Bob gets $z = sX \cdot Y$, Bob calculates $u = z - |Y|^2 = sX \cdot Y - |Y|^2$ and sends u to Alice; Alice calculates and sends w to Bob. Bob decides whether w is equal to $z = sX \cdot Y$. If

$$\begin{aligned} w &= \frac{s}{s-2}(u - |X|^2) \\ &= \frac{s}{s-2}(sX \cdot Y - |X|^2 - |Y|^2) \\ &= \frac{s}{s-2}((s-2)X \cdot Y + (2X \cdot Y - |X|^2 - |Y|^2)) \\ &= sX \cdot Y + \frac{s}{s-2}(2X \cdot Y - |X|^2 - |Y|^2) \\ &= z, \end{aligned}$$

then $2X \cdot Y - |X|^2 - |Y|^2 = 0$, that's $X = Y$;
Otherwise $X \neq Y$. Now, for the sake of statement, we define binary predicates:

$$P(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

Protocol 2:
Vector equality confidentiality determination protocol.

Input: Alice and Bob input rational number vectors $X = (x_1, \cdots, x_n)$ and $Y = (y_1, \cdots, y_n)$.

Output: Bob outputs $P(X, Y)$.

Steps:

1) Using X and Y as input vectors for Protocol 1, Alice and Bob call Protocol 1, Alice gets a random number $s > 2$, Bob gets $z = sX \cdot Y$.
2) Bob calculates $u = z - |Y|^2$, and sends u to Alice.
3) Alice calculates $w = \frac{s}{s-2}(u - |X|^2)$, and sends w to Bob.
4) Bob outputs $y = P(w, z)$.

Correctness of Protocol 2:
According to the definition of the binary predicate $P(X, Y)$, needs to prove $w = z \iff X = Y$. So we just have to prove that this is true:

$$w = \frac{s}{s-2}(z - |X|^2 - |Y|^2).$$

According to the operation properties of vectors, for any rational number vector, X and Y, and for any relation $z = sX \cdot Y$, the following equation holds:

$$\begin{aligned} w &= \frac{s}{s-2}(u - |X|^2) \\ &= \frac{s}{s-2}(z - |X|^2 - |Y|^2) \end{aligned}$$

Therefore, Agreement 2 is correct.

Protocol 2 security: In Protocol 2 execution, Protocol 1 is called first, Alice gets a secret random number s, and Bob gets a secret value $z = sX \cdot Y$.

Let's first consider the security of the Alice vector. According to the security of Protocol 1, Bob can get a linear relation of vector X at most in the process of calling Protocol 1, and receive 2 sent by Alice in the subsequent execution of protocol $w = \frac{s}{s-2}(u - |X|^2)$. When the vector $X \neq Y$, Bob gets nothing about vector X from the relation $w = \frac{s}{s-2}(u - |X|^2)$ (where S is Alice's private data). Therefore, the security of Alice vector X in Protocol 2 is consistent with that in Protocol 1.

Now let's think about the security of the Bob vector. In the process of calling Protocol 1, Alice got the linear relation of any two components of Bob vector Y_1 (and Y_2), but could not get the information of vector Y. In the subsequent execution of Protocol 2, Alice gets the $u = z - |Y|^2$ sent by Bob. For Alice, $y_1, \cdots, y_n$ are regarded as an unknown quantity, and only a quadratic relation about Y can be obtained from the relation $u = z - |Y|^2$.

According to the above analysis, during the implementation of Protocol 2, Bob can get at most a linear relation of Alice vector X, and Alice can get at most a quadratic relation of Bob vector Y. According to the denseness of rational numbers, even if the participants have infinite computing power, they cannot get the private vector of each other. The security of Protocol 2 is similar to that of Protocol 1.

Regarding the security of Protocol 2, the following Theorem 2 is only described. The proof of Theorem 2 is similar to that of Theorem 1, so it is omitted.

Theorem 2. *Vector equality confidentiality determination Protocol 2 is secure.*

Table 1: Efficiency analysis and range analysis of the protocols

Literatures	Calculation Function	Computational Complexity (M or B)	Communication Data Volume	Scope of Application
Literature [13]	Dot product	25n/2-2 (B)	3n	Rational number
Literature [12]	advantage	2 (mn+1)logN+n (M)	mn+2	positive integer
Agreement herein 1	Dot product	8n+19 (B)	2n+6	Rational number
Agreement herein 3	advantage	6n (B)	4	Rational number

5 Algorithm Analysis

The communication data volume required by the dot product Protocol 1 in this paper is t $(n+2)+2$; The point product protocol in reference [13] requires $3n$ of communication data. When $t = 2$, the communication data of Protocol 1 in this paper is $2n + 6$. With the increase of vector dimension n, the communication efficiency advantage of Protocol 1 in this paper is more obvious.

When $t = 2$, the security of Protocol 1. According to the previous analysis, Bob can get at most a linear relationship between any $t + 1 = 3$ components of Alice vector X, but Alice can't get any information about Bob vector Y In reference [13], Alice and Bob can respectively obtain the values of the relational expressions $y_{2k+1} + y_{2k}$ and $x_{2k-1} + x_{2k}$ than between adjacent components of each other's private vector.

Therefore, when $t = 2$ is selected, the security and efficiency of Protocol 1 in this paper are improved compared with that in literature [13], and it is more suitable for solving vector secret computing problems in the case of big data Moreover, this paper can flexibly select the number of decompositions according to the security requirements of the specific vector dot product security computing problem. Protocol 1 in this paper has higher flexibility and wide applicability.

Analysis and comparison of vector equality protocols. Protocol 2 in this paper studies the problem of determining the equality of two vectors. As far as we know, there are few literatures that directly study the problem of vector equality. Protocol 3 in literature [12] can be used to compare the equality of two vectors Since the scheme in reference [12] is only applicable to positive integer vectors, the scope of application is limited, and unilateral errors may occur, while Protocol 2 in this paper is designed based on shared dot product Protocol 1, which is more efficient and has a wider scope of application.

The amount of communication data required by Protocol 3 in this paper is 4; The amount of communication data required by Protocol 2 in reference [6] is mn The communication load of Protocol 3 in this paper is low.

The detailed protocol efficiency comparison results are shown in Table 1. In Table 1, the column of calculation function is the specific content of the listed literature research; In the calculation complexity column, M (or B) represents the number of modular multiplication (or basic arithmetic) operations, and the communication complexity is the amount of communication data.

According to Paillier public key encryption system, N is the product of two large prime numbers, with a general length of 1024 bits. We have noticed that in real life, the vector dimension n generally satisfies $n << \log N$. In this case, the computational complexity of Protocol 1 in this paper is lower than that of existing literature, and the communication complexity is not higher than that of existing protocols; The computational complexity of Protocol 3 is far lower than that of the existing literature, while our protocol is applicable to vector computing within the range of rational numbers and has wider applicability. In addition, Protocol 1 and Protocol 3 in this paper have higher security in the case of rational number vectors, and the leaked information has little impact on security.

In the protocol efficiency experiment. The efficiency of the protocol designed in this paper has been theoretically analyzed and compared with the existing results. Next, we will conduct further experimental tests, and compare the execution results of the protocol in this paper with those of existing protocols with high efficiency. Here, we will determine the number of vector decompositions in protocol $t = 2$.

1) The computer configuration of the experimental platform is as follows:
   The operating system is Windows10 Enterprise Edition, Intel (R) Core (TM) i5-6600 CPU 3.30GHz, installed memory 8.00GB, 64 bit operating system The protocols are programmed and implemented on MyEclipse using Java programming language. It is agreed that all simulation experiments in this paper will be conducted under this environment.

2) Experimental results. Since the protocols 1 and 3 in this paper and literature [13] perform basic arithmetic (exponential) operations, literature [3, 12] applies Paillier encryption scheme. The following simulation experiments are conducted in two different environments.

   The experiment sets the large prime number used in the Paillier encryption algorithm, the bits of p, q are 256 bits, and the range of confidential data is uniformly limited to [-100, 100]. The following is

Table 2: Analysis of simulation result of Protocol 1

	Literature [13]	Literature [3]	Agreement herein 1
Average time consumption of 10000 experiments (ms)	90.308	18926.5	63.092

Table 3: Analysis of simulation result of Protocol 3

	Literature [3]	Literature [12]	Agreement herein 3
Average time consumption of 10000 experiments (ms)	5798.4	4323.8	11.9033

the actual calculation of the protocol in this paper 1 and [3,13], as well as the protocol in this paper 3 and [10,14]. Each protocol is tested for many times in different dimensional vectors, and 50 groups of data are randomly selected from the experimental results to calculate the average value Since Protocol 1 and Protocol 3 in this paper take too little time to display when conducting an experiment, the results can be obtained by averaging each group of data for 10000 cycles and 100 cycles respectively. The results are shown in Table 2 and Table 3.

6 Conclusion

In this paper, we mainly study the safe multi-party computation of vectors over rational number field, including vector dot product, vector equality, vector superiority and so on. A safe and efficient computing protocol is designed to extend the application range of vector safe computing. The security analysis and validity analysis of the protocol show that the proposed protocol has obvious advantages in security and validity compared with existing protocols. Finally, some new vector problems and computational geometry problems are solved by using the designed protocol.

Acknowledgments

This work is supported by 2023 Zhejiang Traditional Chinese Medical Science and Technology Program (Fund No. 2023ZF010). This article has been funded by the fifth research project of vocational education and teaching reform in Jiangsu Province. (Project No. ZYB705), and Jiangsu Province 2022 University Philosophy and Social Science Research Project. (Project No. 2022SJYB1608).

References

[1] F. Benhamouda, S. Halevi, T. Halevi, "Supporting Private Data on Hyperledger Fabric with Secure Multiparty Computation," in *IEEE International Conference on Cloud Engineering (IC2E'18)*, pp. 357-363, 2018.

[2] L. Chao, B. Dha, C. Xh, et al., "Blockchain-based system for secure outsourcing of bilinear pairings," *Information Sciences*, vol. 527, pp. 590-601, 2020.

[3] P. K. Fong, J. H. Weber-Jahnke, "Privacy preserving decision tree learning using unrealized data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 2, pp. 353-364, 2012.

[4] S. Garg, R. Vashisht, "A Permissioned Blockchain System for Secure Multiparty Computation," *Journal of Physics: Conference Series*, vol. 1998, no. 1, 2021.

[5] T. Halevi, F. Benhamouda, A. D. Caro, *et al.*, "Initial Public Offering (IPO) on Permissioned Blockchain Using Secure Multiparty Computation," in *2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 91-98, 2019.

[6] H. Huang, X. Y. Li, Y. Sun, L. S. Huang, "PPS: Privacy-preserving strategyproof social-efficient spectrum auction mechanisms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1393-1404, 2015.

[7] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*, Hoboken, USA: John Wiley & Sons, 2011.

[8] E. Kim, H. S. Lee, J. Park, "Towards Round-Optimal Secure Multiparty Computations: Multikey FHE Without a CRS," *Lecture Notes in Computer Science*, vol 10946, 2018.

[9] M. J. Li, J. S. T. Juan, J. H. C. Tsai, "Practical electronic auction scheme with strong anonymity and bidding privacy," *Information Sciences*, vol. 181, no. 12, pp. 2576-2586, 2011.

[10] S. D. Li, C. Y. Wu, D. S. Wang, Y. Q. Dai, "Secure multiparty computation of solid geometric problems and their applications," *Information Sciences*, vol. 282, pp. 401-413, 2014.

[11] Y. P. Li, M. H. Chen, Q. W. Li, W. Zhang, "Enabling multilevel trust in privacy preserving data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 9, pp. 1598-1612, 2012.

[12] L. G. Liu, H. Sun, H. L. Jia, Y. Zhang, "CGIM: classificatory group index method for efficient ranked

search of encrypted cloud data," *Chinese Journal of Electronics*, vol. 47, no. 2, pp. 331-336, 2019.

[13] Y. J. Liu, X. Luo, A. Joneja, C. X. Ma, X. L. Fu, D. W. Song, "User-adaptive sketch-based 3D CAD model retrieval," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 783-795, 2013.

[14] S. Pentyala, D. Railsback, R. Maia, *et al.*, "Training Differentially Private Models with Secure Multiparty Computation," *arXiv preprint*, arXiv:2202.02625 2022.

[15] A. Smahi, Q. Xia, H. Xia, N. Sulemana, A. A. Fateh, J. Gao, X. Du, M. Guizani, "A blockchainized privacy-preserving support vector machine classification on mobile crowd sensed data," *Pervasive and Mobile Computing*, vol. 66, no. 1, 2020.

[16] C. M. Tang, G. H. Shi, Z. A. Yao, "Secure multiparty computation protocol for sequencing problem," *Science China Information Sciences*, vol. 54, no. 8, pp. 1654-1662, 2011.

[17] T. Toft, "Sub-linear, secure comparison with two non-colluding parties," in *Proceedingg of International Conference on Practice and Theory in Public Key Cryptography Conference on Public Key Cryptography*, pp. 174-191, 2011.

[18] X. Yi, F. Y. Rao, E. Bertino, A. Bouguettaya, "Privacy-preserving association rule mining in cloud computing," in *Proceedins of the 10th ACM Symposium on Information, Computer and Communications Security*, pp. 439-450, 2015.

Biography

Hui Xia is currently an associate professor in Software College of Shenyang Normal University. He received received the B.S. and M.S. degree from XiDian University, China in 2003 and 2006, respectively.He has authored or coauthored more than twenty journal and conference papers. His current Acknowledgments research interests include data mining, privacy preserving and network security.

Chunhua Wang, lecturer, Master of Engineering. Main research directions: computer application technology, image processing, pattern recognition, Analysis of algorithms, data mining, data analysis, vocational education.

Weiji Yang works in Zhejiang TCM university, got bachelor's degree of computer and science in 2005, received double master's degrees of engineering and medicine in 2009 and 2014 respectively, the main research area is artificial intelligence, digital medical image processing and analysis, and smart health care, etc.

Research on the Influence of Cooperative Interference on the Physical Layer Security Performance of Wireless Networks in Wireless Communication

Liyun Xing
(Corresponding author: Liyun Xing)

Chongqing Three Gorges Vocational College, Chongqing 404155, China
Email: xingliy1983@outlook.com

(Received Jan. 18, 2023; Revised and Accepted Oct. 21, 2023; First Online Apr. 25, 2024)

Abstract

Information protection is very important in wireless communication. This paper briefly introduces the wireless communication model based on cooperative interference protection and then uses a genetic algorithm (GA) to allocate the node transmission power of cooperative interference. After that, simulation experiments were carried out in MATLAB software to test the performance of a wireless communication network with or without cooperative interference under different numbers of legitimate transmitting nodes. The results showed that the integrity of the information obtained by the eavesdropping nodes in the wireless network was greatly reduced, the probability of secure connection in the network and the system capacity were greatly improved, but the number of nodes participating in cooperative interference was limited. Too many interference nodes can not effectively improve the system capacity and reduce the probability of a secure connection.

Keywords: Cooperative Interference; Genetic Algorithm; Physical Layer; Wireless Communication

1 Introduction

Compared with wired communication technology, wireless communication technology is more free in space deployment because it is not limited by connecting wires and other devices [12]. For users of wireless communication technology, as long as they are within the communication range, they can receive information without space restrictions, which is very convenient. However, compared with wired communication technology, wireless communication technology is more tested in communication security [5]. Wireless communication technology uses electromagnetic waves in a specific frequency band to transmit information, and the broadcast characteristics of electromagnetic waves make the transmitted information directly exposed to the outside world. As long as it is within the range of signal coverage, any device can receive the signal, and there is a possibility of being eavesdropped.

For the problem of wireless communication being eavesdropped, the encryption algorithm is usually used to encrypt the information [9], so as to ensure that the information will not reveal the content even if it is eavesdropped. However, the way of information encryption requires the energy and computing power of communication nodes, and the existing encryption algorithms are challenging to satisfy the demands for both low complexity and high security at the same time. In addition to encrypting information, the physical characteristics of the wireless channel can also be used to ensure communication security. Cooperative interference is a communication security measure that utilizes the physical characteristics of wireless channels. Its basic principle is to optimize the communication quality of legitimate channels or degrade the communication quality of eavesdropping channels.

Zeng *et al.* [15] developed a cross-layer optimization framework for the cooperative interference model in multi-hop networks. Simulation results showed that the session throughput could be significantly improved (more than 50%) by using cooperative interference. Ibrahim *et al.* [3] studied the selection of relay and jammer in two-way cooperative networks to improve their physical layer security. Wang *et al.* [13] put forward a relay and jammer selection strategy to improve the security against eavesdropping attacks. This paper briefly introduces the wireless communication model based on cooperative interference protection and then uses a genetic algorithm (GA) to allocate the node transmission power of cooperative interference. After that, simulation experiments were carried out in MATLAB software.

2 Security Protection of Wireless Network Based on Cooperative Interference

The cooperative interference method is one of the physical protection methods [2]. Its basic principle is that when the sending node sends information to the receiving node, other legitimate nodes in the whole wireless communication area also send interference signals at the same time, and the interference signals are used to reduce the quality of the eavesdropping channel without affecting the legitimate channel as much as possible. Figure 1 shows the wireless communication model based on cooperative interference protection [10]. When sending node S sends information to receiving node D, it also broadcasts the information to the eavesdropping node. At the same time, other legitimate nodes will also send signals to the outside, which are considered interference signals for eavesdropping nodes. The interference signals sent by other legitimate nodes will also interfere with receiving node D. Therefore, when using collaborative interference techniques for wireless communication protection, it is necessary to allocate the transmission power of the sending node in a way that maximizes the interference on eavesdropping nodes and minimizes the interference on receiving nodes [4].

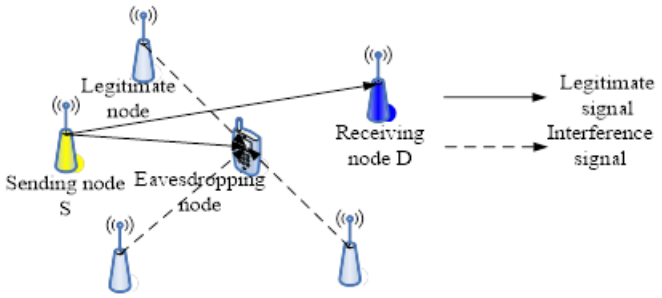


Figure 1: Wireless communication model based on cooperative interference protection

In wireless communication, the received signal-to-noise ratio (SNR) of any receiving node needs to be higher than the minimum demodulation threshold γ_0 to effectively receive information. Therefore, in the cooperative interference protection method, the transmitting node needs to make its transmission power as small as possible to reduce the signal coverage under the premise of ensuring the SNR of the receiving node is higher than γ_0 [6]. Other legitimate nodes used for interference need to increase the transmission power on the premise that the SNR of their receiving nodes is not lower than γ_0, so as to increase the interference to the eavesdropping nodes [1]. In this paper, a GA is used to optimize the transmitting power amplification coefficient to maximize the secure connection probability. The process is as follows.

Step 1. The necessary wireless network-related parameters are input, including γ_0, U (the set of transmitting nodes), D (the set of receiving nodes), D^*_{mat} (the adjacency matrix of the distance between transmitting nodes and receiving nodes), and H^*_{mat} (the channel gain matrix).

Step 2. The chromosome population required by the GA is generated. The gene segment in the chromosome represents the transmitting power amplification coefficient of a legal node, and a chromosome represents a group of power amplification coefficients, which also represents a transmission power allocation scheme. When the chromosome population is randomly generated, the power amplification coefficient of a legitimate node represented by a gene segment must be an integer multiple of 0.5 and no less than 1, and it can not exceed the preset maximum value. The length of the chromosome depends on the number of legitimate transmitting nodes used for interference.

Step 3. The fitness value of each chromosome in the population is computed. The ultimate goal is to maximize the safe connection probability, so it is taken as the fitness value of the chromosome. The formulas are:

$$\begin{cases} SCP_{d^*} = \\ \quad 1 - exp[\frac{-D^{\alpha}_{u^*e} N_0 |h_{u^*d^*}|^2}{D^{\alpha}_{u^*d^*}(N_0 + \sum_{i=1}^{N} P_{u_i} |h_{u_i d^*}|^2 D^{-\alpha}_{u_i d^*})}] \\ \quad \times \prod_{u_i} [\frac{P_{u_i} |h_{u^*d^*}|^2 D^{\alpha}_{u^*e}}{D^{\alpha}_{u^*d^*} D^{\alpha}_{u^*e} (N_0 + \sum_{i=1}^{N} P_{u_i} |h_{u_i d^*}|^2 D^{-\alpha}_{u_i d^*})}] \\ P_{u^*} \simeq \frac{\gamma_0 D^{\alpha}_{u^*d^*} N_0}{|h_{u^*d^*}|^2} \\ P_{u_i} = A_i \frac{\gamma_0 D^{\alpha}_{u_i d_i} N_0}{|h_{u_i d_i}|^2} \end{cases} \quad (1)$$

where SCP_{d^*} is the secure connection probability of sending node u^* and receiving node d^* [14], α is the path loss coefficient [7], D_{u^*e} is the transmission distance between u^* and eavesdropping node e, N_0 is Gaussian white noise, $|h_{u^*d^*}|^2$ is the channel gain between u^* and d^*, $D_{u^*d^*}$ is the transmission distance between u^* and d^*, N denotes the number of legitimate transmitting nodes used for interference, u_i is the i-th legitimate transmitting node used for interference, d_i is the corresponding receiving node of u_i, P_{u_i} is the transmitting power of u_i, $|h_{u_i d^*}|^2$ is the channel gain between u_i and d^*, $D_{u_i d^*}$ is the transmission distance between u_i and d^*, $|h_{u_i d_i}|^2$ is the channel gain between u_i and d_i [8], $D_{u_i d_i}$ is the transmission distance between u_i and d_i, and A_i is the transmitting power amplification coefficient of u_i.

Step 4. Whether the GA terminates the optimization is determined. The termination conditions include: the number of iterations reaches the preset number or the population fitness converges to stability. If the termination condition is reached, the next step is entered. If not, the genetic operation is performed. Crossover

means exchanging the homogenic fragments of the two chromosomes based on the crossover probability, and mutation means randomly changing the gene fragments in the chromosome based on the mutation probability. In this paper, the random change in accordance with the restrictions is performed on A_i. After the genetic operation, return to Step 3.

Step 5. After the termination of the GA, the transmitting power of each transmitting node that can maximize SCP_{d^*} is obtained. Whether there is d_i whose SNR is smaller than γ_0 under this transmitting power is judged. If not, the transmitting power allocation result of each transmitting node is output. If it exists, the sending node with the smallest $|h_{u_i d_i}|^2 D_{u_i d_i}^{-\alpha}$ in the set of legitimate sending nodes used for interference is deleted, the corresponding receiving node is also deleted, and it returns to Step 2.

3 Simulation Experiment

3.1 Experimental Setup

Simulation experiments were carried out in MATLAB software [11]. The wireless network parameters used for the simulation experiments are shown in Table 1, in which the number of legitimate sending nodes including specific sending nodes was set to 3, 4, 5, 6, 7, 8, and 9 respectively, and the corresponding number of receiving nodes was also the same. The simulation experiments were designed to test the impact of the number of interfering nodes on the protection of wireless communication.

The GA was used to adjust the transmitting power, and the related parameters are as follows. The population size was 15; the first three chromosomes were duplicated as the offspring. The crossover probability was 0.5, the mutation probability was 0.1, and the iteration number was 100.

According to the above conditions, 5,000 random experiments were carried out on the wireless network for each number of legitimate sending nodes. In each experiment, a specific sending node sent 1 MB of data to the corresponding receiving node, and the eavesdropping node tried to receive it. To improve testing efficiency, data transmission in the simulation experiment was not encrypted.

3.2 Test Results

In the random experiments with different numbers of legal sending nodes, a specific sending node sent 1 MB of data to the corresponding receiver node, and in each random experiment, the eavesdropping node eavesdropped on the sent information. The average integrity of the data obtained by the eavesdropping node with or without cooperative interference is shown in Figure 2. It can be seen that in the case of no cooperative interference, the average integrity of the information obtained by the eavesdropping node was maintained at about 90% with the increase in the number of legitimate transmitting nodes because the transmitted information was not encrypted in the experiment. The reason why it failed to reach 100% is that the eavesdropping node was far away from the transmitting node in the random experiment, which reduced the SNR of the eavesdropping node and failed to obtain the transmitted data. In the case of cooperative interference, the average integrity of the information that can be obtained by the eavesdropping node was greatly reduced, and it continued to decrease with the increase in the number of legitimate sending nodes.

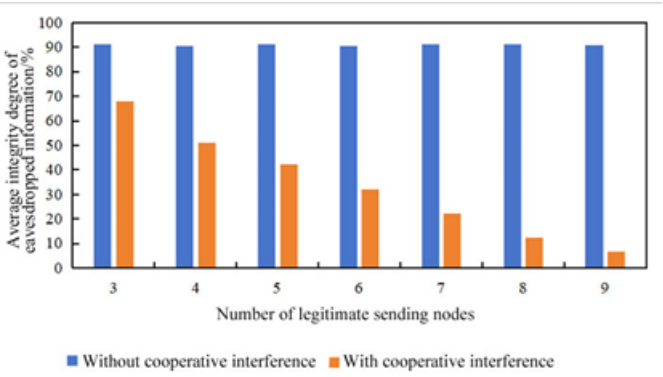


Figure 2: Information integrity of eavesdropping nodes with or without cooperative interference under different numbers of legitimate sending nodes

In the random experiments with different numbers of legitimate sending nodes, the secure connection probability with or without cooperative interference is shown in Figure 3. It can be seen that in the case of no cooperative interference, the secure connection probability in the simulated wireless network did not change significantly and basically stayed at about 71%. In the case of cooperative interference, the secure connection probability in the simulated wireless network first increased and then decreasesd with the increase in the number of legitimate sending nodes. When the number was 6, the secure connection probability was the largest. The reason is that in the case of no cooperative interference, the information of the sending node might be obtained by the eavesdropping node. However, with cooperative interference, the channel of the eavesdropping node was interfered by other nodes, which reduced the probability of being eavesdropped and improved the probability of secure connection. With the increase in the number of legitimate sending nodes, the interference to the eavesdropping node increased, and the probability of a secure connection was also improved. However, when the number of legitimate transmitting nodes was too large, the interference signals generated by them also interfered with the normal receiving nodes.

In the random experiments with different numbers of legitimate transmitter nodes, the system capacity with or

Table 1: Wireless network simulation parameters

Parameter	Setting
Area specification	300 m × 300 m
Number of legitimate sending nodes	4, 5, 6, 7, 8, 9 {3}
The transmission distance of the corresponding sending and receiving nodes	5 m ∼ 15 m
The transmission distance between the sending node and other receiving nodes	20 m ∼ 120 m
The transmission distance between the eavesdropping node and the specific sending node	10 m ∼ 50 m
Channel gain between nodes	Randomly generated, with a mean of 1
α	3
N_0	1
A_i	Its value ranges from 1 to 6 and is a multiple of 0.5
γ_0	5 dB

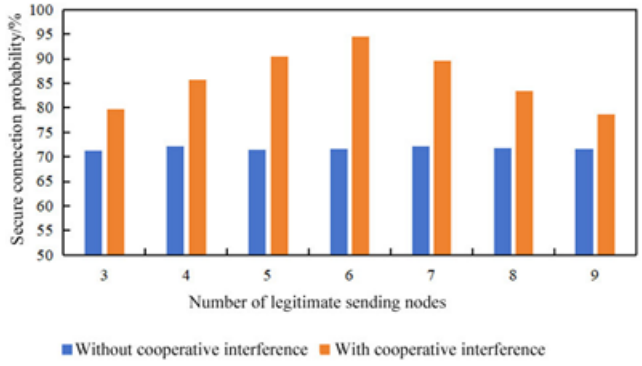


Figure 3: Secure connection probability under different numbers of legitimate sending nodes with or without cooperative interference

without cooperative jamming is shown in Figure 4. The system capacity refers to the transmission rate of wireless communication in a unit frequency band. It can be seen that with the increase in the number of legitimate transmitting nodes, the system capacity in the wireless network without cooperative interference almost did not change. However, the system capacity in the wireless network with cooperative interference first increased and then tended to be stable. The reason is that in the case of no cooperative interference, because eavesdropping nodes stole and interfered with the transmitted information, the system capacity failed to increase after the addition of legitimate transmitting nodes. When there was a cooperative interference, the eavesdropping node was affected, and the probability of a secure connection between the sending node and the receiving node increased, leading to an increased amount of information that could be transmitted, so the system capacity increased. However, when the number of legitimate transmitting nodes was too large, the interference signal affected the receiving node, causing the interference node to fail in correctly demodulating information.

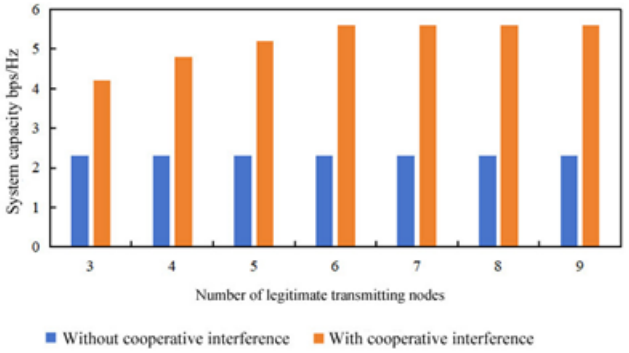


Figure 4: System capacity with and without cooperative interference for different numbers of legitimate transmitting nodes

4 Conclusion

This paper used a GA to allocate the node transmitting power under cooperative interference and then carried out simulation experiments in MATLAB software. In the experiment, the performance of a wireless communication network with or without cooperative interfer-

ence under different numbers of legitimate transmitting nodes was tested. With the increase in the number of legitimate sending nodes, the integrity of information obtained by the eavesdropping nodes in the network without cooperative interference was almost unchanged, and the integrity of information in the network with cooperative interference was not only smaller but also decreased gradually. With the increase in the number of legitimate sending nodes, the secure connection probability of the network without cooperative interference was almost unchanged, while the secure connection probability of the network with cooperative interference first increased and then decreased. It was the highest when the number of sending nodes was 6. With the increase in the number of legitimate sending nodes, the system capacity of the network without cooperative interference remained unchanged, and the system capacity of the network with cooperative interference gradually increased and tended to be flat after the number reached 6.

References

[1] X. Hu, C. Kai, Z. Guo, J. Gao, "A fast forward full-duplex cooperative relay scheme for securing wireless communications," *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 775-779, 2019.

[2] K. Z. Huang, Y. Hong, W. Y. Luo, S. B. Lin, "A method for physical layer security cooperation based on evolutionary game," *Journal of Electronics & Information Technology*, vol. 37, no. 1, pp. 193-199, 2015.

[3] D. H. Ibrahim, E. S. Hassan, S. A. El-Dolil, "Relay and jammer selection schemes for improving physical layer security in two-way cooperative networks," *Computers & Security*, vol. 50, no. may, pp. 47-59, 2015.

[4] N. Kolokotronis, M. Athanasakos, "Improving physical layer security in DF relay networks via two-stage cooperative jamming," in *Signal Processing Conference*, pp. 1173-1177, 2016.

[5] B. Li, J. Zhou, Y. Zou, F. Wang, W. Cao, "Security and reliability trade-off analysis of joint user and jammer selection in the face of co-channel interference," *IET Communications*, vol. 13, no. 6, pp. 2601-2608, 2019.

[6] C. Li, Y. Liu, Q. Xu, Y. Tang, "Self-interference cancellation with frequency offset and nonlinear distortion suppression for cooperative jamming communications," *IEEE Communications Letters*, vol. 23, no. 11, pp. 2091-2094, 2019.

[7] Y. Liu, L. Wang, T. T. Duy, M. Elkashlan, T. Q. Duong, "Relay selection for security enhancement in cognitive relay networks," *IEEE Wireless Communications Letters*, vol. 4, no. 1, pp. 46-49, 2017.

[8] H. Long, W. Xiang, Y. Li, "Precoding and cooperative jamming in multi- antenna two-way relaying wiretap systems without eavesdropper's channel state information," *IEEE Transactions on Information Forensics & Security*, vol. 12, no. 6, pp. 1309-1318, 2017.

[9] Q. Ning, T. Yang, B. Chen, X. Zhou, C. Zhao, X. Yang, "Cooperative transmission of wireless information and energy in anti-eavesdropping UAV relay network," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1283-1292, 2021.

[10] A. E. Shafie, D. Niyato, N. Al-Dhahir, "Security of rechargeable energy-harvesting transmitters in wireless networks," *IEEE Wireless Communications Letters*, vol. 5, no. 4, pp. 384-387, 2016.

[11] H. A. Shah, I. Koo, "Improving physical layer security via cooperative diversity in energy-constrained cognitive radio networks with multiple eavesdroppers," *International Journal of Communication Systems*, vol. 32, no. 14, pp. e4008.1-e4008.18, 2019.

[12] S. Vahidian, S. Aissa, S. Hatamnia, "Relay selection for security-constrained cooperative communication in the presence of eavesdropper's overhearing and interference," *IEEE Wireless Communications Letters*, vol. 4, no. 6, pp. 577-580, 2015.

[13] K. Wang, L. Yuan, T. Miyazaki, D. Zeng, S. Guo, Y. Sun, "Strategic antieavesdropping game for physical layer security in wireless cooperative networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9448-9457, 2017.

[14] W. Yang, L. Xiao, L. Sun, Q. Li, "Cooperative transmission against impersonation attack using inter-session interference in two-hop wireless networks," in *IEEE Trustcom/BigDataSE/ISPA*, pp. 104-110, 2015.

[15] H. Zeng, X. Qin, X. Yuan, Y. Shi, Y. T. Hou, W. Lou, "Cooperative interference neutralization in multi-hop wireless networks," *IEEE Transactions on Communications*, vol. 66, no. 2, pp. 889-903, 2018.

Biography

Liyun Xing, born in September 1983, has obtained a master's degree. She is working at Chongqing Three Gorges Vocational College. She is interested in computer application technology and vocational education.

MDAA: An Unsupervised Anomaly Detection Method for Terminal Traffic in New Power System Based on MDAA

Hao Yang[1], Junfeng Zhang[2], Jia Sun[3], and Xin Xie[4]
(Corresponding author: Xin Xie)

State Grid Jiangxi Electric Power Research Institute[1]
State Grid Jiangxi Electric Power Co., Ltd[2]
Nanchang, 330096, China
State Grid Jilin Power Supply Company[3]
Jilin, 132011, China
East China Jiaotong University[4]
Nanchang, 330096, China
Email: xiexin@ecjtu.edu.cn
(Received July 19, 2023; Revised and Accepted Mar. 4, 2024; First Online Apr. 25, 2024)

Abstract

With the digitalization and intelligence development of new power systems, many IoT intelligent devices with different rates, diverse types, and protocols continue to be connected. It poses a severe challenge to obtaining effective traffic features of terminal devices and achieving accurate intrusion detection. Existing anomaly detection methods are facing severe challenges in this regard. This paper proposes an unsupervised anomaly detection model called MDAA (Multidimensional Attention Autoencoder). First, the damping incremental statistical information algorithm is utilized to perform multidimensional feature extraction on the original traffic packet to enhance the information expression of samples. Meanwhile, the random forest is combined with the traditional attention model to reasonably allocate attention weights for different information so that the latent vectors generated by the encoder can fully express the traffic features of terminal devices, further enhancing the reconstruction ability of the decoder and improving the unsupervised anomaly detection accuracy of the system. Finally, experiments were conducted on real IoT datasets. The results demonstrate that compared with the traditional method and the current mainstream unsupervised model, the detection accuracy of the proposed model is greatly improved, the maximum precision reaches 95.88%, and compared with the current mainstream unsupervised models, the proposed model has better comprehensive performance.

Keywords: Anomaly Detection; Internet of Things; MDAA

1 Introduction

With the transformation and upgrading of traditional power systems to new power systems, as well as the continuous advancement of digitalization and intelligent construction, a large number of IoT terminals sensing, transmission, and control devices are connected to the network, leading to an increasing number of grid assets and increasingly complex device topology structures. Due to the openness of new power systems, many IoT devices are connected to the Internet, resulting in continuously increasing physical threats and network-to-attack risks, which bring new difficulties for the electricity grid's safe and steady operation. Attackers can cause network congestion and paralysis by sending malicious packets, exploiting known vulnerabilities, constructing botnets, launching DDoS, APT attacks, etc. [28]. For example, in 2015, the Russian hacker group "SandWorm" launched a malicious attack on the Ukrainian national power grid, causing a large-scale blackout in Ukraine and drawing high attention and vigilance from home and abroad on the security of power systems and their infrastructure networks. Therefore, it is necessary to build a new power system security defense system, enhance its resilience, elasticity, and self-healing ability, and ensure its security and controllability [29].

For new power systems transitioning to digitalization and intelligence, the large number of IoT terminals means the harm caused by attacks will be greatly magnified. Therefore, anomaly detection for terminal flow in new power systems is extremely important. Anomaly detection effectively detects network intrusion by analyzing data patterns that do not conform to expected behaviors.

Network traffics anomaly detection [16] collects network traffic information on operational networks and analyzes it to determine any attack behavior in the network [23]. Its algorithm requires obtaining traffic features and combining them to detect different network attack behaviors. However, in new power systems, many terminal devices have different rates, diverse types, large network traffic, and different data formats, making it a highly challenging task to obtain effective traffic features for terminal devices.

Machine learning-based intrusion detection techniques have made significant progress in recent years [3, 5, 11, 24, 26]. The most popular method, which is data-driven, is to use a neural network for traffic detection [9]; after experts collect traffic data, abnormal traffic is marked. The difference between normal and attack traffic is learned and classified via the algorithm model, based on which new traffic is detected.

Therefore, an unsupervised learning model that does not require marked abnormal data samples for training, and only uses an algorithm to fit the model to the detection results of the input sample data was selected for use in this study [14].

Traditional unsupervised models, such as the autoencoder (AE), use the reconstruction errors of normal and abnormal samples to judge abnormalities [2]. With the increase in the volume and complexity of device data, encoders have limited ability to learn features; therefore, extracting the effective features of original data has become an important challenge. Furthermore, a fixed latent vector also limits the decoder's ability to do its job, partly due to the inability to express the important features of the input sample in the latent vector. The impact of anomaly detection will be significantly reduced in these circumstances. This research suggests an unsupervised anomaly detection model for terminal traffic based on an attention mechanism and AE in light of the current issues. The proposed model consists of two main components: feature extraction and anomaly detection, and does not require labeled anomalous data samples for training but rather matches the detection results of the model with input sample data using an algorithm. Experiments were conducted on a real-world IoT traffic dataset collected by Ayyoob *et al.* [1], demonstrating the proposed model's advanced performance and applicability.

The contributions of this paper are as follows:

1) A feature extraction scheme that can fully extract the features of the original traffic data packet and effectively express the sample information is proposed.

2) A novel attention mechanism-based unsupervised anomaly detection approach is developed, combining the random forest with the conventional attention model.

3) The proposed MDAA (Multidimensional Attention Autoencoder) was tested on multiple real IoT datasets. The results of the experiment prove that the speed of feature extraction was faster, and the anomaly detection effect was better (the precision reached 95.88%) than those of other methods.

The rest of this essay is structured as follows. The related work and theoretical foundation of this research are described in Section 2. The MDAA model is introduced in Section 3, along with its structural layout. The effectiveness of the suggested strategy is experimentally verified in Section 4. Finally, Section 5 offers the principal conclusions, key discoveries, and future research directions.

2 Related Work

2.1 Unsupervised Learning

While deep learning has many advantages in anomaly detection, the model cannot be trained well with massive labeled traffic.; therefore, substantial research has been carried out on unsupervised learning [6]. Yisroel *et al.* [17] proposed an unsupervised learning model called Kitsune, which uses damped incremental statistics to quickly extract time series of data and detect real-time anomalous traffic from dynamic data streams using the KitNet core algorithm [30].

The AE is an algorithm for unsupervised learning that aims to construct a mapping relationship between the potential space vector z and the original (input) sample space variable x; the encoder and decoder make up the structure, illustrated in Figure 1.

The encoders function is to compress the vectors in the input feature space into the latent feature space to get the latent vector z. In the encoding step, the high dimensional data is mapped into low dimensional data to decrease the amount of data. To achieve the reproduction of the input data, the decoder returns the representation of the latent layer feature space to the original input space. For any input , the latent vector z can be obtained by the encoding function. The encoding function can be expressed as:

$$z = f(x) = s(W_1 x + p) \tag{1}$$

where s is the activation function, and W_1 denotes the weight matrix between the input and latent layers, and p is the bias term.

The decoding function used in the decoder stage is defined as follows.

$$x' = g(z) = \sigma(W_2 z + q) \tag{2}$$

where W_2 is the weight matrix for decoding and q is the bias term.

The goal of the AE is to make the reconstruction error between x and $x^{'}$ as small as possible, and its objective function is defined as follows.

$$L(x, x^{'}) = \|x - x'\|^2 = \|x - \sigma(W_2(s(W_1 x + p)) + q)\|^2 \tag{3}$$

At present, AEs are widely used for anomaly detection. Xie *et al.* [27] designed a model called MHMA. It uses

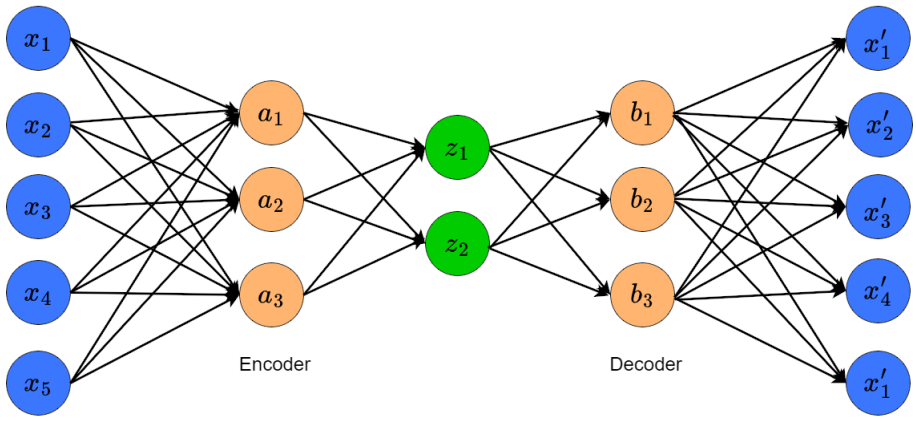


Figure 1: The structure diagram of the traditional autoencoder

the encoding function to retrieve the most relevant items in the memory and aggregates them before passing them to the decoder for reconstruction. However, when normal and abnormal samples are very similar, there are still some instances in the model that cannot be fully distinguished. Dong *et al.* [7] developed a model called MemAE, which uses the encoding function as a query to retrieve the most relevant items in the memory, and these items are then aggregated and passed to the decoder. But the detection accuracy of the method suffers when the images in the dataset have more complex content and show greater intra-class variance over multiple classes. Swee *et al.* [15] used adversarial AEs to convert high-dimensional multimodal data to low-dimensional single-mode potential distribution data with clear tail probabilities, simultaneously optimized the parameters for deep AEs and mixed models in an end-to-end manner, and used an independent estimation network to promote the update of the parameters; the resulting model can better deal with complex real-world traffic data. And this method only uses a plain multivariate Gaussian prior with a fixed mean and covariance for the experiments. Bo *et al.* [31] designed a deep AE Gaussian mixture model, which uses an end-to-end approach to jointly optimize the parameters of the deep AE and the hybrid model, and employs an independent estimation network to promote the parameter learning of the hybrid model to reduce reconstruction error. But the average precision and recall of this method compared to the baseline method needs to be further improved. The current challenge is designing an AE to extract effective features, obtain a reasonable reconstruction loss, and make the reconstructed data distribution close to the real traffic data [21]. To address these problems, this paper introduces the attention mechanism, which will be further explained in Section 2.2.

2.2 Attention Mechanism

The attention mechanism enables a deep network model to suppress other areas of less significance and pay greater attention to a particular small area of interest to the user [20]. Attention models have seen significant use recently in several deep learning applications, including image processing [10], speech recognition [8], and natural language processing [18], among others. Long input sequences make it difficult to retain all the relevant information (due to the large amount of information and fixed length). The attention mechanism model introduces attention weights and gives the decoder access to the encoded input sequence. Learning the attention weights is the responsibility of the network structure's attention module, which is also in charge of creating a new vector that the decoder utilizes as input. The vector is the weighted sum of the encoder's latent states and related attention weights. As shown in Equation (4).

$$\tilde{x}_j = \sum_{j=i}^{T} a_{ij} h_j \tag{4}$$

The traditional weighted summation method cannot effectively score the importance of the original sample features. Therefore, the calculation method of the attention weight is improved in this article. The random forest model is combined with the attention layer to improve the sensitivity of the encoder to important features; as a result, the hidden vector can reasonably express the important information of the original sample.

3 MDAA Anomaly Detection Model

This paper uses the most typical volumetric attack sample [22] as the main positive sample. Firstly, features are extracted from the original data, and an improved attention mechanism is introduced to enhance the latent vector representation of important features, thereby strengthening the decoder's ability to reconstruct the original sample.

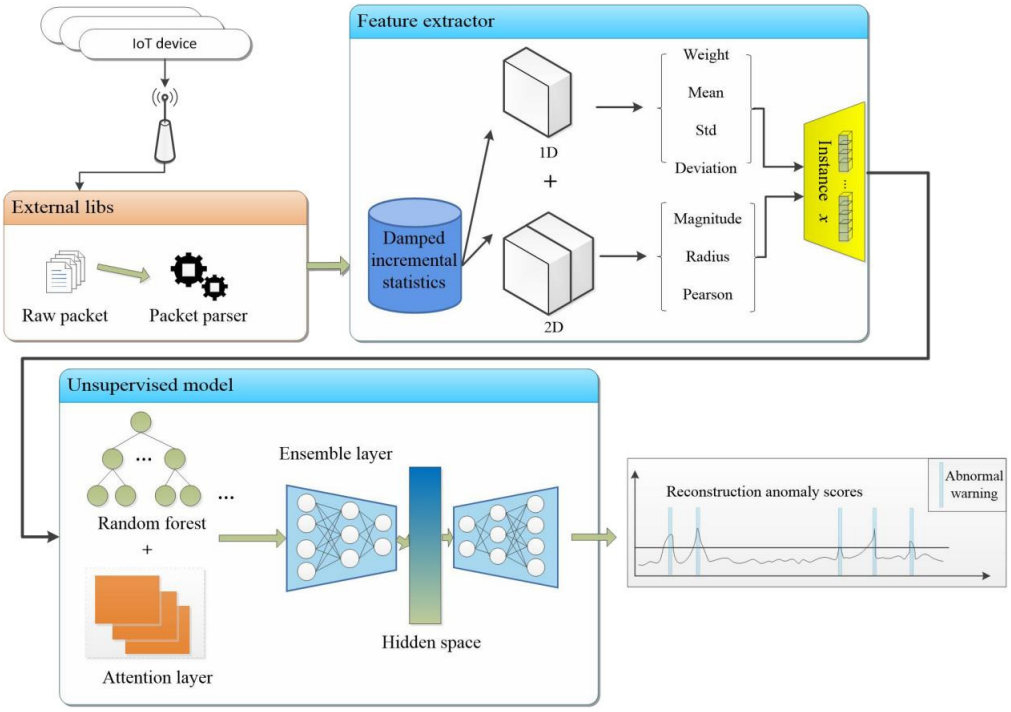


Figure 2: Overview of the MDAA

The structural design of the proposed model is introduced in this section, and the overall concept of the model is shown in Figure 2. Only normal data are utilized to train the model during the training phase. It should be noted that the input data are not the original traffic data but the data output by the designed feature extraction module. Therefore, additional convolutional layers are optional to extract high-level features, reducing convolution operations and the model training time. To achieve the adaptive extraction of the dimensional importance of each moment, increase the expressiveness of the encoder, and strengthen the decoder's capacity to reconstruct hidden vectors, a mechanism for attention is included in the encoder period, and the calculation method of the attention weights is modified. Finally, the 3sigma principle [4] is used to process the reconstruction error of the input sample. A large reconstruction error means that the input sample is anomalous since it deviates from the expected data distribution; otherwise, it is a normal sample.

3.1 Improved Feature Extraction Scheme

First, given an unbounded data stream $X = \{x_1, x_2, x_3, \cdots\}$, the mean, standard deviation, and deviation of the data stream X are updated incrementally by maintaining tuples $IS := (N, S, SS)$, in which N, S, and SS respectively represent the number, linear sum, and sum of squares of instances at that time. Specifically, the update process for the insertion of x_i into the IS is $IS \leftarrow (N+1, S+x_i, SS+x_i^2)$.

Table 1 reports a list of statistics calculated using the damped incremental statistics algorithm [1]. The statistics that contain one and two incremental statistics are called one-dimensional and two-dimensional statistics. To extract the current behavior of the data stream, the old instance must be forgotten. The decay function can be expressed as follows:

$$d_\lambda(t) = 2^{-\lambda t} \tag{5}$$

where λ is the decay factor and t is the time elapsed from the inflow of data stream x_i to the last observed time. The tuple of the damped incremental statistics is defined as $IS_{i,\lambda} := (w, S, SS, RP_{ij}, T_{last})$, where w is the number of current instances,T_{last} is the last updated timestamp of the tuple $IS_{i,\lambda}$, and RP_{ij} is the residual product between data stream I and data stream J. Algorithm 1 describes how a new value is updated with x_{cur} at time t_{cur}.

When a packet arrives, 110 traffic statistics of the packet are extracted, including the source MAC and IP address from the packet (represented as SrcMAC-IP), the source IP from the packet (represented as SrcIP), the channel sent between the source IP and the destination IP of the packet, and the socket sent between the source TCP/UDP socket and the destination TCP/UDP socket of the packet. Table 2 reports the 22 features extracted from a time window. This feature extraction module extracts the same features from five-time windows, resulting in 110 features. In addition, the feature mapping module after the feature extraction module in Kitsune is removed; therefore, the 110 features obtained after the feature extraction module are used as the input data for subsequent

anomaly detection.

Algorithm 1 The algorithm for inserting a new value into a damped incremental statistic.

1: $\gamma \leftarrow d_\lambda(t_{cur} - t_{last})$ Computer decay factor
2: $IS_{i,\lambda} \leftarrow (\gamma w, \gamma S, \gamma SS, \gamma RP, T_{cur})$ Process decay
3: $IS_{i,\lambda} \leftarrow (w+1, S + x_{cur}, SS + x_i^2, RP_{ij} + r_i r_j, T_{cur})$ Insert value
4: **return** $IS_{i,\lambda}$;

3.2 Unsupervised Model Based on an Improved Attention Mechanism

Figure 3 depicts the precise configuration of the unsupervised model that this paper proposes.

An attention mechanism is introduced in the proposed model. It assigns higher weights to important features by scoring the importance of the features of input samples, thereby allowing the encoder to retain important information during the encoding process. In short, for each input x_t, a different attention weight β_t^k is assigned to each feature, and the attention weight measures the importance of the k-th feature of the sample at time t. Determining how to find a reasonable value that can fully represent the importance of features is particularly important. In this regard, the random forest model [25] is combined with an attention mechanism for the first time, thereby improving the traditional calculation method of weighted summation and the rationality of β_t^k. Random forest is an ensemble learning method that selects more valuable feature attributes by calculating the Gini coefficient. Then this paper uses multiple CART decision trees to complete the classification task. In the proposed method, the Gini index calculates the characteristic contribution, also known as the Gini impurity.

$$GI = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} {p_k}^2 \quad (6)$$

$$S_{jm} = GI_m - GI_i - GI_r \quad (7)$$

where GI_i and GI_r denote the Gini indices for the two new nodes after branching.

Assuming that there is a total of n decision trees within the random forest and for feature x_j the importance score is given by Equation (8). The generated significance scores are all finally normalized.

$$S_j = \sum_{i=1}^{n} S_{ij} \quad (8)$$

$$S_j = \frac{S_j}{\sum_{i=1}^{m} S_j} \quad (9)$$

Where S_j is the importance score of the j-th feature based on the importance of features obtained by the random forest.

Inspired by Yao [19], a fixed attention model, such as the multi-layer perceptron, is adopted, and the attention weights are further calculated as follows:

$$e_t^k = v_e^T tanh(W\delta_{t-1} + Ux^k) \quad (10)$$

$$\varphi_t^k = \frac{exp(e_t^k)}{\sum_{i=1}^{n} exp(e_t^i)} \quad (11)$$

Where v_e^T, W , and U are the learned parameters, and φ_t^k is the attention weight of the importance of the k-th feature at time t. In an effort to guarantee that the total attention weights equal 1, the softmax function is then applied to e_t^k. The final attention weight in this study is obtained via the combination of the random forest and attention model:

$$\beta_t^k = a\frac{(\varphi_t^k)}{(\varphi_t^k + S_T^k)} + b\frac{(S_t^k)}{(\varphi_t^k + S_t^k)} \quad (12)$$

where a and b are hyperparameters set based on experience. In this way, the input sequence x can be updated to $\tilde{x}$ via Equation (13), and the updated value of $\tilde{x}$ is used as the input of the encoder.

$$\tilde{x} = (\beta_t^1 x_t^1, \beta_t^2 x_t^2, \beta_t^3 x_t^3, ..., \beta_t^k x_t^k) \quad (13)$$

This attention mechanism effectively combines the advantages of both the random forest and attention mechanism, and it can assign higher weights to the important features of the input samples. Therefore, more important features can be preserved in the hidden vector, which is helpful for the decoder to reconstruct the sample.

4 Experimental Study

The configuration of the experimental hardware was an i5-9400H CPU with 16G memory and an NVIDIA GTX1660Ti graphics card with 6G video memory. The overall implementation framework of the model was the Keras framework [12], and the Adam optimizer [13] with a learning rate of 0.001 was used to optimize the model.

4.1 Dataset

1) Commonly used IoT device datasets:

These datasets originally consisted of real IoT data collected by Ayyoob *et al.* [1], who collected data packets from the test platform, including both benign and attack traffic. In this study, the original traffic collected was re-divided into four representative datasets according to the type of volumetric attack. The Samsung Camera and Netatmo Camera datasets contain only TCP flooding attack data, the WeMo Switch dataset contains only ping of death attack data, and the Chrome Cast dataset contains only simple service discovery protocol (SSDP) attack data.

Table 1: List of incremental statistics that can be calculated from streams S_i and S_j

Type	Statistic	Notation	Calculation
1D	Weight	w	w
	Mean	μ_{X_i}	S/w
	Standard	σ_{X_i}	$\sqrt{\lvert SS/w - (S/w)^2 \rvert}$
	Deviation	η_{X_i}	$x_i - S/w$
2D	Magnitude	M_{X_i,X_j}	$\sqrt{\mu_{X_i}^2 + \mu_{X_j}^2}$
	Radius	R_{X_i,X_j}	$\sqrt{(\sigma_{X_i}^2)^2 + (\sigma_{X_j}^2)^2}$
	Person	P_{X_i,X_j}	$\frac{RP_{ij}}{(w_i+w_j)\sigma_{X_i}\sigma_{X_j}}$

Table 2: Statistical features extracted from each time window λ when a packet arrives

Statistics	Aggregated by	Number of features
μ_i, σ_i, η_i	SrcMAC-IP, Channel, Socket	9
$M_{i,j}, R_{X_i,X_j}, P_{X_i,X_j}$	Channel, Socket	6
w_i	SrcMAC-IP, SrcIP, Channel, Socke	4
w_i, μ_i, σ_i	Channel	3

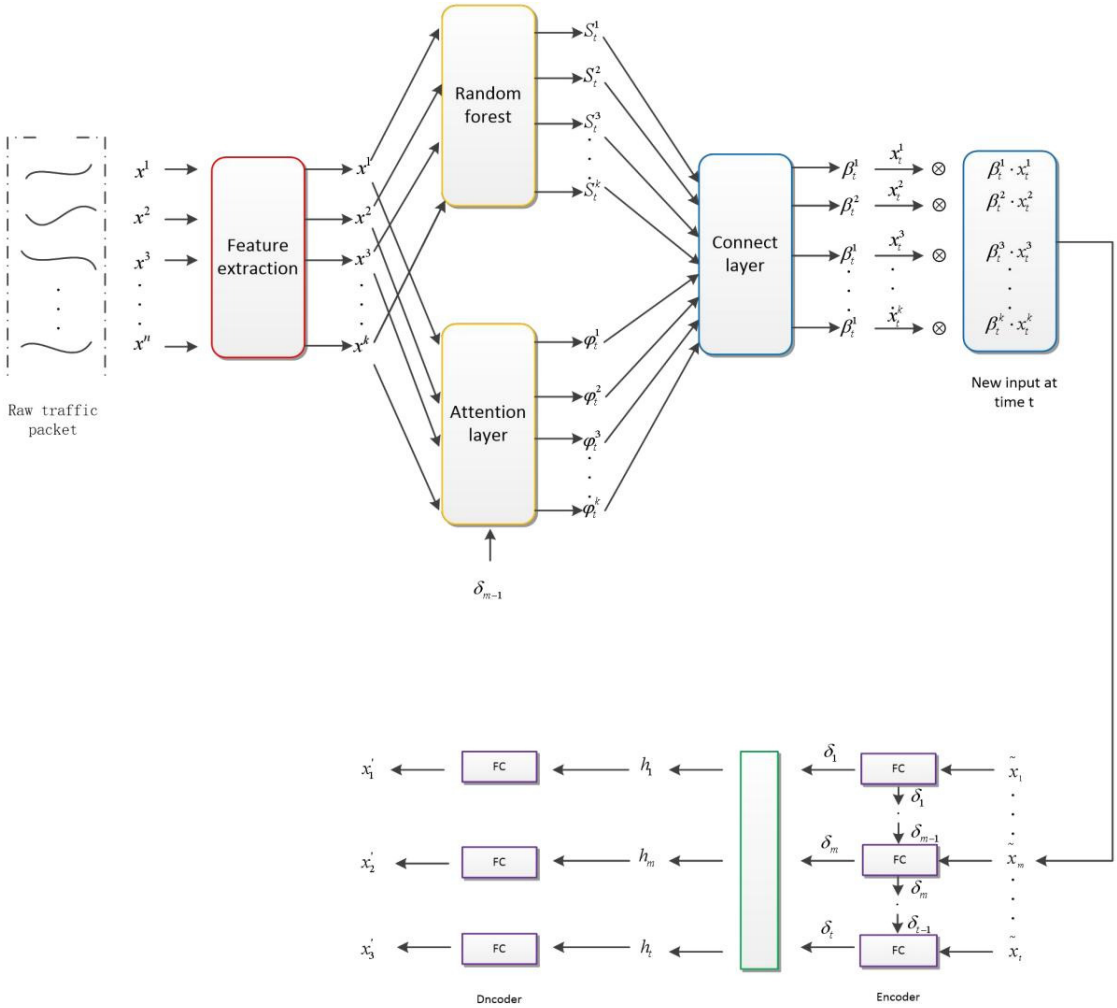


Figure 3: The structure of MDAA

2) Multiple attack data:

The Multiple attack dataset was constructed based on the traffic capture of IP webcam devices deployed in laboratory environments by Yisroel *et al.* [17]. It includes data on a variety of attacks, including OS scan, fuzzing, video injection, APR MitM, active wiretap, SSDP flood, SYN Dos, SSL renegotiation, and Mirai attacks.

3) CIC-IDS-2017:

The CIC-IDS-2017 dataset is an intrusion detection dataset released by the Information Security Excellence Center (ISCX) of the University of New Brunswick (UNB) in 2017. It contains benign and up-to-date common attack data collected over a period of five days.

Detailed information about the datasets is reported in Table 3.

Table 3: Statistics of the datasets

Dataset	Instances	Anomaly ratio
Samsung camera	235531	0.15
WeMo switch	241015	0.05
Chrome-cast	256200	0.2
Netatmo camera	222015	0.1
Multiple attack data	27000	0.3
CIC-IDS-2017	90000	0.2

4.2 Metrics

In this paper, the samples with attack traffic are referred to as positive samples, while the normal samples are referred to as negative samples. The anomaly detection performance of the proposed model was evaluated based on the precision, recall, and F1 score. Accuracy and recall are used to evaluate the quality of results, while F1 score is used to comprehensively reflect the overall metrics. This work mainly focuses on device data, and focus was placed on whether the false alarm rate could be effectively reduced while detecting real negative samples. Therefore, the F1 indicator was mainly used to measure the performance of the proposed anomaly detection method.

4.3 Result

Experiment 1: Comparison of anomaly detection performance on typical datasets

This experiment validates the performance of the model using three single-attack datasets and two multiple-attack datasets. Only negative samples are used in training process, and the training set and validation set are divided in a ratio of about 8:2. The validation set was used to adjust the hyperparameters of the model and preliminarily evaluate the ability of the model. The ratio of abnormal samples to normal samples in the test set was approximately 1:1.

In the experiment, the fully connected neural network was used as the basic structure of the encoder and decoder. In the outlier detection stage, the reconstructed mean absolute errors (MAEs) of the normal and abnormal samples, namely the mean value of the absolute errors between the observed and true values, were compared. The 3sigma principle was used to handle the outliers and finally determine the abnormal samples. The proposed MDAA model was compared with the classical and most advanced algorithms used in the unsupervised learning field, and the results are reported in Table 4.

As can be seen from Table 4, the proposed MDAA model achieved good performance on the six datasets, which demonstrates that MDAA can exhibit superior detection performance in the face of both a single attack and multiple attacks. It is also worth noting that CIC-IDS-2017 does not contain data from IoT devices, but MDAA still achieved outstanding results on this dataset, indicating that MDAA has good generalization performance in different types of traffic data detection, and therefore has certain applicability.

Experiment 2: The effectiveness of the feature extraction scheme

The Samsung Camera dataset was used for this experiment. The experimental results are shown in Table 5.It can be seen that the model achieves the best accuracy, F1 score, and recall when the number of features is 110. Therefore, under certain conditions, the increase of the number of features will provide increasingly effective information for the model, and will help the model to make accurate judgments. Moreover, it can be concluded that in the proposed feature extraction scheme, reducing the time overhead will inevitably result in the loss of model performance. Therefore, under the condition of ensuring similar accuracy, the time overhead can be reduced by compressing the number of features. After the number of features yields the optimal situation, it must be considered how the proposed feature extraction scheme is superior to other models.

In Table 6, Kitsune represents the physical sign extraction scheme in the NIDS proposed by Yisroel *et al.* [17], which extracts features from the original data and ultimately obtains 115 features. In Kitsune*, the feature extraction module in Kitsune was replaced with the proposed feature extraction scheme, and 110 features are included. It can be seen that Kitsune* achieved the highest precision on all four data sets and the highest recall and F1 values on two datasets, which demonstrates that the proposed feature extraction scheme has a significant effect on most real IoT traffic. In addition, the proposed feature extraction scheme was found to reduce the time consumption, which means that although 110 features were extracted, the calculation of the model was simpler and the computational time was reduced.

Table 4: Anomaly detection results on six datasets

Method	Samsung Camera			WeMo Switch		
	Pre	Recall	F1	Pre	Recall	F1
OC-SVM	0.8239	0.6642	0.7128	0.5187	0.4357	0.4705
AE	0.7292	0.8161	0.7215	0.3935	0.4645	0.4303
DCN	0.7363	0.7477	0.7021	0.5230	0.4987	0.4734
DAGMM	0.5565	0.7821	0.7152	0.2759	0.4024	0.4327
TadGAN	0.6221	0.5327	0.5756	0.1762	0.2421	0.1988
Kitsune	0.9261	**0.9922**	0.7204	0.4188	0.5863	0.4902
MDAA	**0.9422**	0.8151	**0.8361**	**0.6421**	**0.7452**	**0.6972**
Method	Chrome-Cast			Netatmo Camera		
	Pre	Recall	F1	Pre	Recall	F1
OC-SVM	0.8424	0.9793	0.9099	0.7119	0.8058	0.7592
AE	0.7836	0.9211	0.8186	0.7422	0.8231	0.7662
DCN	0.7329	0.8943	0.8187	0.7182	0.7891	0.7452
DAGMM	0.7980	0.7726	0.8673	0.5903	0.8952	0.7127
TadGAN	0.6482	0.5988	0.6467	0.6744	0.6987	0.6222
Kitsune	0.6141	0.8575	0.8215	0.6916	**0.9418**	0.7901
MDAA	**0.9186**	**0.9543**	**0.9362**	**0.9080**	0.8522	**0.8638**
Method	CIC-IDS-2017			Multiple attack data		
	Pre	Recall	F1	Pre	Recall	F1
OC-SVM	0.5122	0.6782	0.5901	0.6792	0.7782	0.7292
AE	0.4278	0.6922	0.5481	0.6683	0.7801	0.7431
DCN	0.6002	0.7133	0.6233	0.6776	0.7172	0.6654
DAGMM	0.8297	0.8441	0.8699	0.6322	0.6871	0.7089
TadGAN	0.4772	0.6021	0.5042	0.4322	0.4982	0.4790
Kitsune	0.5956	0.8839	0.7116	0.7083	0.7104	0.7185
MDAA	**0.9588**	**0.9462**	**0.9589**	**0.7979**	**0.9921**	**0.8844**

Table 5: Features comparison

Features	Pre	Recall	F1	Time
132	0.922	0.772	0.780	271.1s
110	**0.942**	**0.815**	**0.836**	273.5s
88	0.893	0.749	0.752	271.5s
66	0.662	0.723	0.673	251.5s
44	0.677	0.737	0.693	256.5s
22	0.682	0.733	0.682	237.1s

Experiment 3:Comparison of reconstruction effects

Unsupervised AE-based models largely rely on the reconstruction errors of samples to judge anomalies. Therefore, whether the reconstruction errors of normal and abnormal samples can be effectively distinguished is an important standard by which to measure the performance of the model. To verify the effectiveness of the attention mechanism, the improved attention was compared with unmodified attention, and the same hyperparameter configuration was used for training and testing on the same dataset. Finally, the reconstruction error was calculated, and the results are shown in Figure 4.

Where (a) and (b) are trained based on Multiple attack data and (c) and (d) are trained based on CIC-IDS-2017. And Figures (a) and (c) represent the reconstruction error of the normal model, and Figures (b) and (d) represent the reconstruction error of the model after adding the improved attention mechanism. In Figure 4, blue dots and orange triangles respectively represent the reconstruction errors of normal and abnormal samples. The graphs on the left present the reconstruction error results of the original model, while the graphs on the right present the model reconstruction error results after the addition of the improved attention mechanism. It can be found that the concentration of blue dots in the right image is higher than that in the left image, and the coincidence degree of the blue dots and orange triangles in the right image is low. Therefore, the normal sample yielded a lower reconstruction error, and the abnormal sample yielded a higher reconstruction error. This demonstrates that the proposed model can perform satisfactory reconstruction for most of the data of normal samples. It also means that the improved attention mechanism can fully extract the key information of normal samples in the training stage, thereby effectively helping the AE reconstruct normal samples. It is worth noting that MDAA is an unsupervised model and uses only normal samples for training. Therefore, in the test stage, any attack sample is consid-

Table 6: Comparison of feature extraction schemes

Dataset	Method	Pre	Rec	F1	Time
Samsung Camera	Kitsune	0.926	**0.992**	0.720	842.149s
	Kitsune*	**0.941**	0.943	**0.737**	272.412s
WeMo Switch	Kitsune	0.408	0.486	0.444	845.848s
	Kitsune*	**0.495**	**0.668**	**0.699**	286.602s
Multiple attack data	Kitsune	0.711	0.684	**0.712**	1034.367s
	Kitsune*	**0.736**	**0.742**	0.701	332.367s

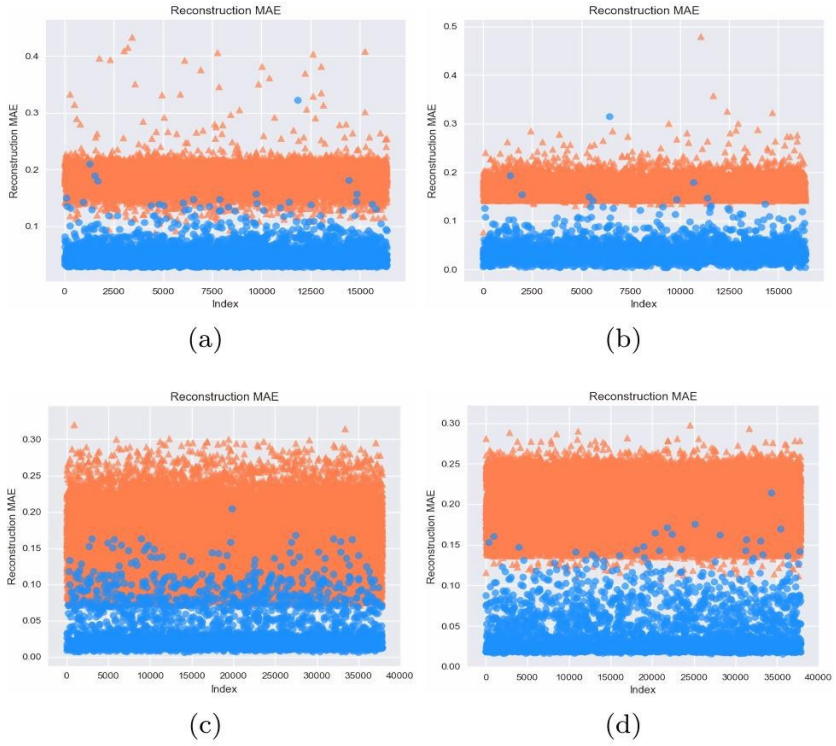


Figure 4: The comparison of the reconstruction error

ered an unknown attack.

5 Conclusions and Future Work

Regarding some of the issues currently present in anomaly detection of traffic in IoT terminal devices, this paper proposed an AE-based anomaly detection method called MDAA, for which an improved feature extraction scheme and a novel attention mechanism were constructed. Via the improved feature extraction scheme, the effective features of original traffic data can be obtained to the greatest extent, and the sample information can be fully expressed. Moreover, the random forest was combined with the traditional attention model for the first time, which optimizes the calculation method and enhances the rationality of the attention weights. The latent vector generated by the encoder can effectively express the importance of the features, and further improves the reconstruction ability of the decoder. Moreover, the proposed model yields obvious differences between the reconstruction errors of normal and abnormal samples. Experiments were conducted on data collected for different IoT devices, and the experimental results demonstrate that the proposed model exhibits a certain degree of versatility while improving the anomaly detection effect. However, the model results in similar reconstruction errors when the normal samples are very similar to the abnormal samples. Therefore, it is easy to incorrectly classify the abnormal samples. Moreover, in the dataset containing the ping of death attack data, although the model achieved the best results by comparison, the overall results were not ideal. Therefore, in future work, more fine-grained learning of samples will be implemented to overcome this problem.

In the new power system, there are a large number of

IoT terminals. In order to improve the digital transformation of the new power system and enhance the demand for terminal security of electronic devices, anomaly detection methods can be applied to IoT terminals such as substation monitoring cameras and smart circuit breakers for routine self-inspection. Proactive defense measures can then be taken, transforming from "remedial action after the fact" to "prevention and control beforehand". Applying deep learning to network security in the new power system will better support network information security, ensure the safe and efficient operation of the new power system, and provide strong support for the long-term development of the power industry.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China, under Grant No. 62162026, and the Science and Technology Project supported by Education Department of Jiangxi Province, under Grant No.GJJ210611.

References

[1] H. Ayyoob, Benson T. A. Gharakheili, H. H., and V. Sivaraman, "Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity," in *Proceedings of the 2019 ACM Symposium on SDN Research*, pp. 36–48, 2019.

[2] Y. Bengio, P. Lamblin, P. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in Neural Information Processing Systems 19, MITPress, Cambridge, MA*, 2007.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[4] P. F. Cao, F. Dai, G. Z. Liu, J. M. Yang, and B. Huang, "A survey of traffic prediction based on deep neural network: Data, methods and challenges," in *International Conference on Cloud Computing*, pp. 17–29. Springer, 2021.

[5] H. W. Deng and X. W. Li, "Network traffic anomaly recognition and detection based on deep learning," *Computer System Applications*, vol. 32, no. 02, pp. 274–280, 2023.

[6] A. de Mello Koch, E. de Mello Koch, R. de Mello Koch, "Why unsupervised deep networks generalize," *arXiv e-prints*, pp. arXiv–2012, 2020.

[7] D. Gong, L. Q. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. V. D. Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1705–1714, 2019.

[8] M. H. Guo, T. X. Xu, J. J. Liu, Z. N. Liu, P. T. Jiang, T. J. Mu, S. H. Zhang, R. Martin, M. M. Cheng, and S. M. Hu, "Attention mechanisms in computer vision: A survey," *Computational visual media*, vol. 8, no. 3, pp. 331–368, 2022.

[9] E. Hodo, X. Bellekens, A. Hamilton, P. L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of iot networks using artificial neural network intrusion detection system," in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–6. IEEE, 2016.

[10] Q. B. Hou, D. Q. Zhou, and J. S. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13713–13722, 2021.

[11] W. X. Jia and P. H. Zhang, "Nonlinear network traffic anomaly detection method based on multi-feature recognition," *Journal of Hubei University of Science and Technology*, vol. 43, no. 02, pp. 145–150, 2023.

[12] Keras, "Keras, github," Apr. 4, 2024. (http://github.com/keras-team/keras)

[13] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization," 2014. (https://arxiv.org/abs/1412.6980)

[14] M. J. Li, H. Z. Huang, L. Ma, W. Liu, T. Zhang, and Y. G. Jiang, "Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 184–199, 2018.

[15] S. Lim, Y. Loo, N. T. Tran, N. M. Cheung, G. Roig, and Y. Elovici, "Doping: Generative data augmentation for unsupervised anomaly detection with gan," in *2018 IEEE international conference on data mining (ICDM)*, pp. 1122–1127. IEEE, 2018.

[16] J. Y. Liu, D. S. Yang, M. J. Lian, and M. S. Li, "Research on classification of intrusion detection in internet of things network layer based on machine learning," in *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, pp. 106–110. IEEE, 2021.

[17] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," *machine learning*, vol. 5, p. 2, 2018.

[18] C. Pierre, C. Chloe, P. Pablo, "Infolm: A new metric to evaluate summarization & data2text generation," vol. 36, no. 10, pp. 10554–10562, 2022.

[19] Y. Qin, D. J. Song, H. F. Chen, W. Cheng, G. F. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.

[20] S. Qureshi, J. He, S. Tunio, N. Zhu, F. Ullah, A. Nazir, A. Wajahat, "An adaptive multi-layer architecture for iot based idps for attacks using deep learning method," *International Journal of Network Security.*, vol. 24, no. 5, pp. 815–827, 2022.

[21] T. Rainforth, A. Kosiorek, T. A. Le, C. Maddison, M. Igl, F. Wood, and Y. W. Teh, "Tighter variational bounds are not necessarily better," in *International Conference on Machine Learning*, pp. 4277–4285. PMLR, 2018.

[22] S. Ramanathan, J. Mirkovic, M. L. Yu, and Y. Zhang, "Senss against volumetric ddos attacks," in *Proceedings of the 34th Annual Computer Security Applications Conference*, pp. 266–277, 2018.

[23] A. Shah, S. Clachar, M. Minimair, and D. Cook, "Building multiclass classification baselines for anomaly-based network intrusion detection systems," in *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 759–760. IEEE, 2020.

[24] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE transactions on emerging topics in computational intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[25] T. M. Tomita, J. Browne, C. C. Shen, J. Chung, J. L. Patsolic, B.n Falk, C. E. Priebe, J. Yim, R. Burns, and M. Maggioni, "Sparse projection oblique randomer forests," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 4193–4231, 2020.

[26] W. B. Wang, H. Liu, H. Lin, P. C. Du, and J. L. Jiang, "Power industry control flow application layer message anomaly detection based on deep learning," *Power System Automation*, pp. 1–12, 2023.

[27] X. Xie, X. L. Li, B. Wang, T. C. Wan, L. Xu, and H. P. Li, "Unsupervised abnormal detection using vae with memory," *Soft Computing*, vol. 26, no. 13, pp. 6219–6231, 2022.

[28] X. Xie, X. Y. Zhang, and J. L. Yang, "Decision tree algorithm combining information gain and gini index," *Computer Engineering and Applications*, vol. 58, no. 10, pp. 139–144, 2022.

[29] J. Zhang, "Detection of network protection security vulnerability intrusion based on data mining," *International Journal of Network Security.*, vol. 21, no. 6, pp. 979–984, 2019.

[30] Q. Y. Zhang, X. W. Hu, Z. Wang, "High quality image steganography model based on encoder-decoder networks and 2d logistic chaotic encryption," *International Journal of Network Security.*, vol. 25, no. 3, pp. 394–408, 2023.

[31] B. Zong, Q. Song, W. Cheng, C. Lumezanu, D. Cho, and H. F. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018.

Biography

Hao Yang. He received his master's degree in computer software and theory from Huazhong University of Science and Technology in 2005.He's currently employed at State Grid Jiangxi Electric Power Research Institute.His research interests include software theory, network and information security.

Junfeng Zhang. He received his master's degree in power systems and automation from Huazhong University of Science and Technology in 2008.He is employed by State Grid Jiangxi Electric Power Co. Ltd.His research interests include network and information security,power system automation, new digital technology.

Jia Sun. She obtained her master's degree in power system and its automation from Nanchang University in 2015. She is currently employed by Jilin Power Supply Company of State Grid Jilin Electric Power Co., Ltd., engaged in the Power System Dispatching Automation and Artificial Intelligence Technology in Power System and its Application in Power System.

Xin Xie. He received his master's degree in Control Theory and Control Engineering of Nanchang University in 2001.He is employed by East China Jiaotong University.His research interests include computer vision, computer network and information security.

Taking LM as the Brain: A Novel Approach Integrating Language Model and Generative Agent for Intelligent Decision Systems

Jingkang Yang[1], Xiaodong Cai[1], Yining Liu[2], Mingyao Chen[3], and Chin-Chen Chang[4]
(Corresponding author: Xiaodong Cai)

School of Information and Communication, Guilin University of Electronic Technology[1]
School of Computer Science and Information Security, Guilin University of Electronic Technology[2]
Research and Development Department, Guilin Topintelligent Communication Technology Company Limited[3]
Guilin 541004, China
Department of Information Engineering and Computer Science, Feng Chia University[4]
Taichung 407102, Taiwan
Email: 335028222@qq.com
(Received Feb. 1, 2024; Revised and Accepted Apr. 3, 2024; First Online Apr. 25, 2024)

Abstract

The Large Language Model (LLM) has demonstrated significant capabilities in intelligent robotics and Autonomous Driving(AD). Compared to traditional end-to-end models, decision reasoning in the form of language exhibits enhanced generalization and interpretability. To harness the inferential decision-making capabilities of large models more effectively, we propose a generative intelligent agent capable of translating various sensor signals into language descriptions, thereby utilizing Language Models (LMs) for decision-making. Furthermore, to address the practical deployment challenges of large models, we fine-tuned a GPT-2 model on an intelligent traffic corpus. The fine-tuned model performs competitively with LLMs in the domain of traffic decision-making. This approach can potentially deploy LMs directly on computationally constrained in-vehicle platforms to reduce communication latency. We evaluated the proposed method on a simulated dataset, and the results indicate that the pure language decision model surpasses the vector-based and fusion models. Moreover, our method exhibits significantly higher efficiency than LLMs.

Keywords: Autonomous Driving; Fine-tuning; Generative Agent; Intelligent Decision Systems; Natural Language Generation

1 Introduction

The success of LLMs has led to development of numerous industries. Tasks that were previously exclusive to human intervention, such as medical consultations [20], programming [22], and writing [26], have demonstrated the potential for accomplishment by LLMs. Recently, there has been a notable surge in research exploring the integration of LLMs into decision-making contexts [3, 8]. In contrast to conventional end-to-end models, employing language-based decision reasoning offers heightened generalization and interpretability [9, 10].

A significant challenge facing modern AD systems is their limited interpretability, which hampers the analysis of accidents. This issue originates from the 'black-box' nature of neural networks, making it difficult for researchers to identify the specific reasons behind these networks' decisions. Although there have been efforts to improve interpretability, current research has yet to fully overcome the challenges posed by real-world driving scenarios.

Different from conventional end-to-end neural networks or multi-stage intelligent decision systems, LLMs exhibit robust chain-reasoning capabilities. Crucially, their reasoning processes can be expressed in natural language, providing a heightened level of interpretability. Initial success has been observed in approaches that amalgamate LLMs with traditional neural networks [1, 4], offering promising prospects for addressing interpretability concerns in AD and intelligent decision systems.

However, prevailing fusion methodologies still exhibit partial reliance on vector representations, constraining the full realization of model interpretability. Moreover, their decision-making relies on textual reasoning provided by LLMs, presenting numerous limitations during practical deployment, such as real-time issues and privacy concerns [25]. LLMs, characterized by a multitude of parameters and substantial computational demands, cannot be directly deployed on in-vehicle platforms or edge nodes. Instead, they necessitate invocation through remote communication. Nevertheless, this approach imposes rigorous

requirements on network communication, with even slight network delays potentially rendering the system incapable of timely response—rendering it unsuitable for real-time scenarios with stringent demands, such as AD.

In order to address these two challenges, we propose a novel LM-based intelligent decision system. Firstly, we employ a generative agent to take over the feature extraction task. This generative agent is capable of transforming various sensor signals into language tokens. While these language tokens may lack the semantic richness of natural language, they are suitable for conducting question-and-answer interactions with LMs. Simultaneously, the generative agent has the capacity to inductively capture historical states, extracting more detailed state descriptions through LMs, thereby further enhancing the interpretability of the model.

Secondly, we substitute the large LM with a GPT-2 model fine-tuned on a textual driving dataset. The fine-tuned model is tailored to the specific application domain, enabling it to achieve comparable performance with a reduced parameter count. Benefiting from the reduction in both parameters count and computational demands, our model can be directly deployed on in-vehicle platforms and edge nodes, effectively resolving the issue of communication latency.

Our contributions can be summarized as follows:

1) We introduce a novel intelligent decision system exclusively based on language models. Employing a generative agent, we manage and preserve the operational state and historical information of vehicles, leveraging language models for inference and decision-making. This architecture maximizes interpretability, facilitating the performance assessment and accident investigation of the system.

2) We substitute the LLM with a fine-tuned GPT-2 as the core model. Compared to LLMs, the fine-tuned model accomplishes decision-making tasks in specific domains with fewer parameters, enabling the core model's direct deployment on in-vehicle platforms, fundamentally resolving the issue of network latency.

3) We evaluate the proposed approach on a simulated dataset, with results demonstrating that our method slightly lags behind fusion methods in terms of performance but significantly outperforms them in terms of inference speed.

The remaining sections of this paper are organized as follows: Section 2 provides a comprehensive review of the application research of large language models in the domains of intelligent decision-making and AD. Section 3 presents a detailed description of the proposed methodology. Section 4 showcases our experimental approach and evaluation results. Finally, we conclude our work in Section 5.

2 Related Work

Due to the rapid development of deep learning, autonomous systems have achieved significant success in recent years [11]. Many autonomous systems are based on end-to-end models to bring about more performance improvements [21]. However, interpretability has been a challenge in deep learning, particularly for end-to-end models [7, 12], posing significant difficulties in accident analysis. In the field of AD, understanding the specific reasons behind intelligent decisions is crucial. It helps establish trust between humans and artificial intelligence, promotes human-AI collaboration, and ensures driving safety [23].

In response to this issue, many scholars have conducted research on the interpretability of neural networks, providing explanations for the intrinsic decisions of deep learning from various perspectives. Ribeiro *et al.* [17] proposed a method to explain the predictions of classification models. Selvaraju *et al.* [18] achieved interpretability by gradient-based localization of important regions predicted by the model. Kim *et al.* [5] used visual attention maps to identify regions of interest to explain the decision-making process of autonomous driving systems. Although these methods have made some progress, they require a significant amount of expertise and technology to implement, making them challenging to popularize in engineering fields. Rohrbach *et al.* [6] introduced natural language into autonomous driving systems, leveraging the easily understandable nature of natural language to explain the intelligent decision-making process.

With the success of LLMs in various domains, an increasing focus has been on combining LLMs with neural networks using vector representations to enhance model interpretability. Fu *et al.* [2] constructed a closed-loop system based on an LLM, demonstrating its capability for environmental understanding and interaction in AD. Xu *et al.* [24] integrated language tokens and visual information, using an LLM to predict vehicle control signals. Mao *et al.* [10] employed an LLM to generate language descriptions of driving trajectories, providing better prompts for LLM-based autonomous driving models. Chen *et al.* [1] proposed a fusion model that combines vector representations and language prompts, enabling an understanding of driving scenarios. Sha *et al.* [19] guided LLM reasoning by designing cognitive paths, leveraging LLM as a decision component for autonomous driving.

Inspired by above researches, we propose a fully LM-based intelligent decision-making system for AD. We use a generative agent to record and manage the vehicle's state and events, facilitating better queries to the LM. To deploy the LM on an in-vehicle platform, we fine-tune a lightweight LM on a text-based driving dataset, replacing LLM as the core of intelligent decision-making.

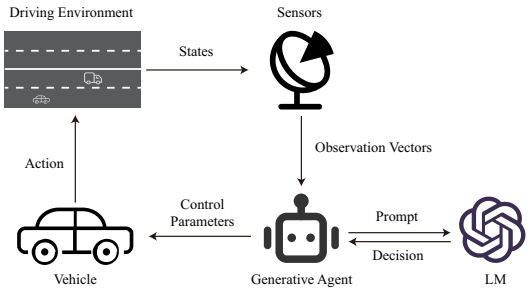


Figure 1: The Complete Framework of the Proposed System

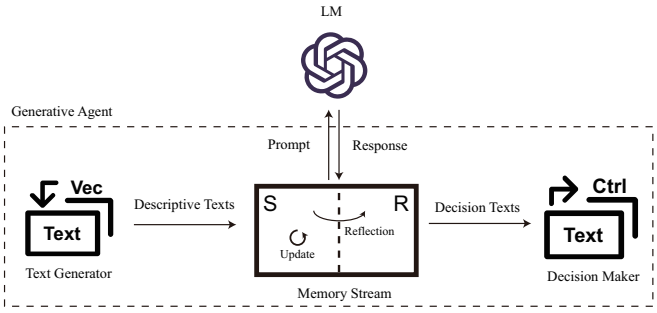


Figure 2: The Structure of the Generative Agent

3 Method

In this paper, we propose a novel framework solely based on an LM, which shown in Figure 1. The generative agent transforms sensor data into language tokens and stores them in a memory stream. It selects appropriate memories to pose questions to the LM. The LM functions as the cognitive center of the system, providing relevant responses based on the inquiries posed. Subsequently, the generative agent executes decisions based on the responses obtained from the LM. Concurrently, the generative agent periodically invokes the LM for "reflection" to obtain more detailed language descriptions of the vehicle's state, further enhancing the interpretability of the system.

3.1 Generative Agent

3.1.1 Functional Modules

The generative agent comprises three modules: a text generator, a memory stream, and a decision converter, as illustrated in Figure 2.

Text Generator. In our approach, we take a structured language generator as the text generator to produce descriptive texts based on sensor data, which is proposed in [1]. These descriptive texts characterize the current state of the vehicle and are subsequently stored in the memory stream. Table 1 shows an example of state information represented by vectors, and the descriptive text generated from it is:

I'm observing 1 car(s) and 3 pedestrian(s).
A moving car; Angle in degrees: 16.19; Distance: 33.96m; Direction of travel: from right to left; My attention: 100%
A pedestrian; Angle in degrees: -6.58; Distance: 11.98m; Direction of travel: from left to right; Crossing: True; My attention: 100%
A pedestrian; Angle in degrees: -27.19; Distance: 7.66m; Direction of travel: same direction as me; Crossing: False; My attention: 1%
A pedestrian; Angle in degrees: 54.13; Distance: 10.44m; Direction of travel: opposite direction from me; Crossing: False; My attention: 0%
The distance to the closest intersection is 22.35m
There is no traffic light(s).
My car -0.12m from the lane center and 0.77 degrees left off center.
My current speed is 7.07 mph
Steering wheel is 2.93% right.
I need to go straight for at least 60.34m.

Memory Stream. The memory stream is a pivotal module within the generative agent, responsible for recording and managing all states and reasoning outcomes of the vehicle during its operation. The memory stream consists of three sections: the status page and the reflection page. The status page logs the state records of the vehicle at each moment during operation, along with corresponding decisions. Meanwhile, the reflection page records more specific descriptions derived from the vehicle state reasoning and higher-level conclusions.

Decision Converter. The decision converter extracts necessary operational information from the responses provided by the LM and translates it into vehicle control signals. In this paper, due to imposed constraints on the format of LM responses, control parameter extraction is implemented using regular expressions. When the intelligent decision system is not authorized to assume direct control over the vehicle, the decision converter merely issues alerts to the driver, foregoing the conversion of LM decisions into control signals.

3.1.2 Decision Making Process

The core task of generative agents lies in managing and updating the memory stream, as well as posing questions to the LM based on the content of the memory stream to obtain decision advice. The complete process includes four steps: evaluation, retrieval, decision, and reflection.

Evaluation. Each time the generative agent acquires a description of the observed states, it assesses the importance of each record within it. In this step, the agent calls the LM, distinguishing the significance of

Table 1: Vector representation of observation states

Ego States	Pedestrian States	Route States	Vehicle States
$[-2.0240, 0.6324, \cdots]$	$[[1.0000, 1.4813, \cdots], [1.0000, 1.6613, \cdots], \ldots]$	$[[0.2354, -0.0119, \cdots], [0.4354, -0.0114, \cdots], \ldots]$	$[[1.0000, 1.0000, \cdots], [1.0000, 1.0000, \cdots], \ldots]$

events by assigning higher scores to those deemed important by the LM. We follow the method proposed in [13], directly questioning the LM and requesting its evaluation. An example prompt is as follows:

On the scale of 1 to 10, where 1 is purely inessential in driving and 10 is extremely important in driving, rate the likely significance of the following piece of memory.
Memory: The traffic light turned red
Rating: <fill in>

Subsequently, the LM provides an importance score of 8 for this event, and this score, along with the corresponding event description, is recorded in the state section of the memory stream. The agent also records the time of each observational state, and this information is utilized in the subsequent retrieval phase.

Retrieval. After recording each state, the generative agent conducts retrieval of state information to select the most appropriate events for the current decision. This is achieved by computing the retrieval score S of each record:

$$S = \alpha S_r + \beta S_i \tag{1}$$

where S_r represents recency, with the proximity to the retrieval time increasing S_r. S_i represents importance, where more important event has a greater S_i. α and β are the coefficients for S_r and S_i, respectively. The calculation methods for S_r and S_i are defined in Equation (2) and Equation (3):

$$S_r = \frac{1}{1 + d \times t} \tag{2}$$

$$S_i = \frac{R}{10} \tag{3}$$

where d is the decay rate and t is the from the observed event to the time of retrieval, R is the importance score provided by LM.

In this paper, we employ the Top-p method to select records with high retrieval scores. The generative agent initially sorts records based on their retrieval scores and subsequently selects a subset of records in descending order where the cumulative retrieval score is greater than or equal to the set threshold p.

Decision. After retrieval, the generative agent organizes the selected subset of records and queries the LM. An example prompt is as follows:

Three seconds ago, My car is passing an intersection.
My actions are:
Throttle (Accelerator) pedal: 40%
Brake pedal: 0%
Steering: Maintain current direction (0% turn)
Now a pedestrian is crossing the road from left to right; Angle in degrees: 5.43; Distance: 28.22m; Direction of travel: from left to right; My attention: 15%
My car is passing an intersection.
There is no traffic lights.
My current speed is 9.56 mph.
Steering wheel is 3.91% left.
The next I need to turn around in 29.43m.
Please act as a professional driver and help me decide what to do in the given situation. The decision consists of three items: the throttle percentage, the brake percentage, and the percentage of left or right turn Angle. (For example: Here are my actions:
- Accelerator pedal 81%
- Brake pedal 0%
- Going to steer 1% to the left.)

Subsequently, the LM provides suggested decisions and their rationale, and this information is recorded in the memory stream. Due to this prompt being generated by a non-intelligent program rather than a language model, there may be some grammatical errors in it. Fortunately, most language models exhibit robustness, and their judgment is not significantly affected by these minor grammar errors.

Reflection. After decision, the generative agent requests the LM to summarize the current driving state and provide a more detailed description to check whether the LM truly comprehends the driving process. An example prompt is as follows:

Three seconds ago, My car is passing an intersection.
My actions are:
Throttle (Accelerator) pedal: 40%
Brake pedal: 0%
Steering: Maintain current direction (0% turn)
Now I'm observing 1 cars and 2 pedestrians.
A moving car; Angle in degrees: -17.38; Distance: 37.92m; Direction of travel: from left to right; My attention: 82%
A pedestrian; Angle in degrees: -27.82; Distance: 6.25m; Direction of travel: same direction as me; Crossing: False; My attention: 100%
A pedestrian; Angle in degrees: -31.59; Distance:

24.59m; Direction of travel: same direction as me; Crossing: False; My attention: 48%
My car is passing an intersection.
There is no traffic lights.
My current speed is 9.56 mph.
Steering wheel is 3.91% left.
The next I need to turn around in 29.43m.
Here are my actions:
Throttle (Accelerator) pedal: 20%
Brake pedal: 0%
Steering: Increase to 8% to the left
Summarize the current driving scenario

Subsequently, the LM provides a summary about the current driving state:

Your car is still passing an intersection with no traffic lights, and your current speed is 9.56 mph. The steering wheel is turned 3.91% to the left. The next action you've planned is:
Throttle (Accelerator) pedal: 20%
Brake pedal: 0%
Steering: Increase to 8% to the left.
You anticipate the need to turn around in the next 29.43 meters.

These question-answer sessions will be recorded in the reflective page of the memory stream for analysis and accident investigation purposes.

3.2 Lightweight Language Model

Previous works utilized LLMs as the decision core for AD due to their powerful reasoning capabilities [2, 19]. However, the high computational demands of LLMs necessitate remote communication with the AD agent, resulting in significant communication delays and challenging real-time requirements for AD. In this subsection, we propose a lightweight approach by fine-tuning a specialized small language model for AD. This model can be directly deployed on in-vehicle platforms, fundamentally addressing the issue of communication delays.

Fine-tuning is a commonly used method to specialize a model, where new datasets are used to further optimize parameters based on a pre-trained model, enabling the model to perform effectively on new tasks. We utilized GPT2-small as the pre-trained model, which has undergone extensive training on a large-scale corpus. Building upon the LLM-Driver Dataset provided by Chen *et al.* [1], we supplemented it with some question-answer pairs generated by ChatGPT-3.5, creating a fine-tuning dataset. We then conducted full-parameter fine-tuning of GPT-2 using this dataset.

4 Experiments

4.1 Experimental Setup

In this subsection, we provide details of the setup, including the experimental platforms, dataset and training hyper-parameters.

4.1.1 Experimental Platform

In this work, we trained and evaluated the models on a high-performance server and conducted efficiency assessments on a mobile device. The purpose of this setup is to simulate the operating conditions of in-vehicle platforms as closely as possible during efficiency assessments. We provide the details for both devices.

1) High-performance server:
OS: Ubuntu 18.04
Intel(R) Core(TM) i7-10700F CPU @ 2.90GHz
RAM: 64GB
GPU: NVIDIA GeForce RTX 3090
VRAM: 24GB

2) Mobile device:
OS: Windows 11 Professional
CPU: Intel(R) Core(TM) i9-13900HX @ 2.20 GHz
RAM: 16GB
GPU: NVIDIA GeForce RTX 4060M
VRAM: 8GB

4.1.2 Dataset and Training

The dataset we utilized primarily originates from LLM-Driver Dataset provided by Chen *et al.* [1], encompassing language descriptions for 10k driving scenarios and corresponding 160k question-answer dialogues. As the proposed generative agent requires additional information, we augmented the training set with 1k question-answer dialogues generated by Chat-GPT3.5, collectively forming a fine-tuning dataset for this work. We evaluate the proposed method using the test set from LLM-Driver Dataset, comprising language descriptions and question-answer dialogues for 1k driving scenarios.

We conducted a 20-epoch full-parameter fine-tuning of GPT-2 on the fine-tuning dataset to enable it to answer various questions related to driving scenarios. The fine-tuning utilized the Adam optimizer with an initial learning rate of $5e-5$ and a batch size of 12.

4.2 Performance Evaluation

We employed Perceiver BC [14], LLM-Driver [1], RWKV-7B [15] and ChatGPT-3.5 as baseline models. Perceiver BC is a comprehensive model capable of handling multiple data types. In this paper, we used Perceiver BC to receive observation vector data as input and directly predict control parameters. LLM-Driver is an integrated model that combines features from vector representations and language prompts for prediction. Additionally, we constructed two pure LM baselines, which utilizes the generative agent to invoke RWKV-7B and ChatGPT-3.5 for prediction.

Performance evaluation is conducted from two perspectives. On one hand, we assess the control parameter errors given by the models. On the other hand, we evaluate the models' question-answering capabilities. For the former, we employ the normalized mean absolute error metric, evaluating acceleration/brake pressure error E_1 and steering wheel angle error E_2 separately. For the latter, we use ChatGPT-3.5 to evaluate the models' responses, a method gradually applied to open-ended text generation tasks [1, 16].

Table 2: The result of performance evaluation

	E_1	E_2	GPT Grading
Perceiver-BC	0.2062	0.1048	-
LLM-Driver	0.1016	0.0115	7.82
RWKV-7B	0.1432	0.0268	6.16
ChatGPT-3.5	**0.0592**	**0.0021**	**9.26**
Ours	0.0644	0.0027	8.42

The evaluation results are shown in Table 2. In all evaluation metrics, the method using ChatGPT-3.5 is the best, due to the LLM's extensive knowledge and powerful reasoning capabilities. The performance of our model is only slightly inferior to that of the LLM. However, our model has significantly fewer parameters than LLMs, giving it an advantage in practical deployment.

4.3 Efficiency Evaluation

To validate the practicability of our proposed model on in-vehicle platforms, we conducted an efficiency assessment of the proposed method on mobile devices and compared it with the approach using RWKV-7B and ChatGPT-3.5. Considering that the method of invoking ChatGPT-3.5 involves remote communication in practical deployment, we also include the communication latency in our assessment. The experimental results shown in Table 3 indicate that our solution achieves high generation speeds on mobile devices, meeting the real-time requirements of AD.

Table 3: Comparison of text generation speed between the proposed method and ChatGPT-3.5 (Including communication delays)

Method	Generation Speed(Tokens/s)
RWKV-7B	6.1
ChatGPT-3.5	13.9
Ours	**98141.2**

5 Conclusions

In summary, our research introduces a pioneering approach to intelligent decision-making in the domain of AD by integrating the LM into the generative agent. This integration not only enhances the interpretability of decision reasoning but also effectively manages and preserves critical information about the operational state and historical context of vehicles. A key aspect of our work involves addressing the challenges associated with the practical deployment of LLMs, such as their computational demands and communication latency issues. To overcome these challenges, we present a fine-tuned GPT-2 model specifically tailored to the AD domain. The fine-tuned model not only achieves competitive performance but also operates with a reduced parameter count, enabling its direct deployment on in-vehicle platforms and edge nodes. This resolution significantly alleviates the concerns related to network latency, making our approach suitable for real-time scenarios with stringent demands, such as AD. The utilization of the fine-tuned LM in decision-making processes, as demonstrated in experimental results, offers a unique advantage over compared fusion models, providing superior performance and interpretability.

Acknowledgments

This study was supported by the Guangxi Science and Technology Major Program under Grant No. AA20302001. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] L. Chen, O. Sinavski, J. Hünermann, A. Karnsund, A. J. Willmott, D. Birch, D. Maund, and J. Shotton, "Driving with llms: Fusing object-level vector modality for explainable autonomous driving," *arXiv preprint arXiv:2310.01957*, 2023.

[2] D. Fu, X. Li, L. Wen, M. Dou, P. Cai, B. Shi, and Y. Qiao, "Drive like a human: Rethinking autonomous driving with large language models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 910–919.

[3] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[4] A. Keysan, A. Look, E. Kosman, G. Gürsun, J. Wagner, Y. Yu, and B. Rakitsch, "Can you text what is happening? integrating pre-trained language encoders into trajectory prediction models for autonomous driving," *arXiv preprint arXiv:2309.05282*, 2023.

[5] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in

Proceedings of the IEEE international conference on computer vision, 2017, pp. 2942–2950.

[6] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata, "Textual explanations for self-driving vehicles," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 563–578.

[7] J. Li and L. Feng, "Prediction of covid-19 transmission risk in universities based on seir and multi-hidden layer back-propagation neural network model," *IJLAI Transactions on Science and Engineering*, vol. 2, no. 1, pp. 1–7, 2024.

[8] J. Mai, J. Chen, B. Li, G. Qian, M. Elhoseiny, and B. Ghanem, "Llm as a robotic brain: Unifying egocentric memory and control," *arXiv preprint arXiv:2304.09349*, 2023.

[9] S. Malla, C. Choi, I. Dwivedi, J. H. Choi, and J. Li, "Drama: Joint risk localization and captioning in driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 1043–1052.

[10] J. Mao, Y. Qian, J. Ye, H. Zhao, and Y. Wang, "Gpt-driver: Learning to drive with gpt," in *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

[11] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. de Albuquerque, "Deep learning for safe autonomous driving: Current challenges and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4316–4336, 2020.

[12] D. Omeiza, H. Webb, M. Jirotka, and L. Kunze, "Explanations in autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 142–10 162, 2021.

[13] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–22.

[14] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–22.

[15] B. Peng, E. Alcaide, Q. Anthony, A. Albalak, S. Arcadinho, H. Cao, X. Cheng, M. Chung, M. Grella, K. K. GV *et al.*, "Rwkv: Reinventing rnns for the transformer era," *arXiv preprint arXiv:2305.13048*, 2023.

[16] E. Perez, S. Ringer, K. Lukošiūtė, K. Nguyen, E. Chen, S. Heiner, C. Pettit, C. Olsson, S. Kundu, S. Kadavath *et al.*, "Discovering language model behaviors with model-written evaluations," *arXiv preprint arXiv:2212.09251*, 2022.

[17] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[19] H. Sha, Y. Mu, Y. Jiang, L. Chen, C. Xu, P. Luo, S. E. Li, M. Tomizuka, W. Zhan, and M. Ding, "Languagempc: Large language models as decision makers for autonomous driving," *arXiv preprint arXiv:2310.03026*, 2023.

[20] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," *Nature medicine*, vol. 29, no. 8, pp. 1930–1940, 2023.

[21] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 537–547, 2020.

[22] F. F. Xu, U. Alon, G. Neubig, and V. J. Hellendoorn, "A systematic evaluation of large language models of code," in *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, 2022, pp. 1–10.

[23] W. Xu, "From automation to autonomy and autonomous vehicles: Challenges and opportunities for human-computer interaction," *Interactions*, vol. 28, no. 1, pp. 48–53, 2020.

[24] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," *arXiv preprint arXiv:2310.01412*, 2023.

[25] S. Yin, H. Li, L. Teng, A. A. Laghari, and V. V. Estrela, "Attribute-based multiparty searchable encryption model for privacy protection of text data," *Multimedia Tools and Applications*, pp. 1–22, 2023.

[26] A. Yuan, A. Coenen, E. Reif, and D. Ippolito, "Wordcraft: story writing with large language models," in *27th International Conference on Intelligent User Interfaces*, 2022, pp. 841–852.

Biography

Jingkang Yang received the B.E. degree in mechanical design-manufacture and automation from Lanzhou Jiaotong University, Lanzhou, China, in 2018. He is currently working toward the Ph.D. degree with the School of Information and Communication, Guilin University of Electronic Technology, Guilin, China. His research interests include data privacy and machine learning.

Xiaodong Cai received the B. E. degree in precision instrumentation from Shanghai Jiao Tong University, Shanghai, China, in 1994, the M. E. degree in digital electronics engineering from University of Sussex, Brighton, UK, in 2001, and the Ph.D. degree in video processing

and communications from University of Sussex, Brighton, UK, in 2008. He is currently a Professor with the Guilin University of Electronic Technology, Guilin, China. His research interests include image processing and video communication.

Yining Liu (Senior Member, IEEE) received the B.S. degree in applied mathematics from Information Engineering University, Zhengzhou, China, in 1995, the M.S. degree in computer software and theory from the Huazhong University of Science and Technology, Wuhan, China, in 2003, and the Ph.D. degree in mathematics from Hubei University, Wuhan, China, in 2007. He is currently a Professor with the Guilin University of Electronic Technology, Guilin, China. His research interests include data privacy, security and privacy in VANETs, image security, and machine learning.

Mingyao Chen received his B.E. degree in Computer Science and Technology from the Guilin University of Electronic Technology, Guilin, China, in 2010. He currently serves as the Deputy General Manager at Guilin Topintelligent Communication Technology Co., Ltd., located in Guilin, China. His primary research interests lie in the field of artificial intelligence.

Chin-Chen Chang (Fellow, IEEE) received the Ph.D. degree in computer engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1987.On numerous occasions, he was invited to serve as a visiting professor, the chair professor, an honorary professor, an honorary director, an honorary Chairperson, a distinguished alumnus, a distinguished researcher, and a research fellow by universities and research institutes. He has been the Chair Professor with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan, since February 2005. His current research interests include database design, computer cryptography, image compression, and data structures.

Monitoring and Management of Sudden Online Public Opinion under Big Data from a Legal Perspective

Fan Tu

(Corresponding author: Fan Tu)

School of Law, Northwest Normal University
No. 967, Anning East Road, Anning District, Lanzhou City, Gansu 730000, China
Email: tu1988f@outlook.com

(Received Jan. 19, 2023; Revised and Accepted Oct. 21, 2023; First Online Apr. 25, 2024)

Abstract

From a legal perspective, there are still many shortcomings in managing sudden online public opinion, which needs to be reliably monitored. In this paper, based on public opinion big data and taking Weibo as an example, a classification method of text sentiment orientation based on bidirectional long short-term memory (BiLSTM)-attention was proposed, and then indicators such as public opinion hotness were used as monitoring indicators. The correlation value was obtained through gray correlation analysis (GRA). The k-means clustering algorithm was used to grade the risk degree of public opinion. The improved sparrow search algorithm-back-propagation neural network (ISSA-BPNN) method was developed to monitor sudden online public opinion. The experimental analysis found that the method based on BiLSTM-attention realized accurate classification of text sentiment tendency, and the F1 value reached 0.836, 0.789, and 0.812, which was better than support vector machine and other methods. The method based on ISSA-BPNN also realized accurate judgment of the risk level of sudden online public opinion, and the monitoring accuracy rate of 20 events reached 95%. The results prove the reliability of the proposed sudden online public opinion monitoring method, which can support legal management.

Keywords: Big Data; Emergency; Monitoring and Management; Online Public Opinion

1 Introduction

Under the background of big data, as Internet technology advances rapidly [17], an increasing number of netizens can freely share their views on social events, and the intense information interaction can easily lead to the emergence of sudden online public opinion [3]. Online public opinion information refers to the views and discussions of netizens on events and topics published on the Internet, with fast dissemination speed, wide range, and complex and diverse content. Discussions of emergencies, when containing many negative emotions, will hurt the society and continue to deteriorate, and if not managed and intervened promptly, it may lead to deeper crises, affecting the stable development of the society. At present, there is a lack of special legal regulations on sudden online public opinion, and political and legal organs are less capable of responding to accountability arising from public opinion. Coupled with the lack of legal awareness of netizens, when faced with sudden online public opinion, they may disseminate it randomly without identifying it, leading to indiscriminate fermentation of public opinion, and there is no clear law to characterize such behavior.

To realize the management of public opinion in the legal perspetive, legislative efforts are still needed, and therefore, it is particularly important to monitor the sudden online public opinion through the means of big data [10]. Barachi et al. [2] developed a complex sentiment analysis framework to monitor people's attitudes towards climate change in online social media and found that the method achieved a recognition rate of 88.41%. Xie et al. [14] designed a model based on the bald eagle algorithm-optimized radial basis function neural network for forecasting online opinion and found through experiments that the method had stable prediction effects. Wang et al. [13] put forward an approach to monitor the drift of opinion leaders in microblog posts. Through the analysis of actual data, they found that the method revealed the drift of opinion leaders and effectively monitored public opinion crises.

Zhang et al. [16] combined machine learning algorithms and an ontology structure to construct a prediction model of public opinion warning level for dispute cases, which realized public opinion warning for medical dispute cases and improved judicial credibility. This paper designed a monitoring method for sudden online public opinion by

classifying the sengtiment orientation of public opinion information and conducted experiments with actual Weibo data as an example to validate the designed method. Some proposals were presented regarding the legal-based management of sudden online public opinion under the legal perspective.

2 Monitoring of Sudden Online Public Opinion Under Big Data

2.1 Sudden Online Public Opinion Data

Weibo, as a widely accessible social platform, has emerged as a significant avenue for individuals to stay informed about current affairs and engage with entertainment news. An increasing number of individuals are utilizing Weibo as a means to actively participate in discussions surrounding trending events, expressing their subjective viewpoints through posts and comments. Such user-generated information constitutes valuable public opinion data, which, if left unattended, can potentially give rise to unforeseen online public opinion events and subsequent storms.

This paper took Weibo as an example, crawled topic data through crawler data, performed cleaning and word segmentation on the data, established monitoring indicators based on the sentiment classification of texts to monitor the sudden online public opinion.

2.2 Methods for Categorizing Text Sentiment Orientation

The purpose of text sentiment orientation classification is to determine whether the sentiment of the text is positive or negative and to understand the implicit sentiment attitude of the speaker. Before classification, it is first necessary to convert the text into a form that can be understood by the computer. This paper uses the skip-gram model in word2vec [1], which can realize the speculation of multiple words in context based on the current word $w(t)$. The objective function can be written as:

$$L = \sum_{w \in C} \log p(context(w)|w),$$

where $p(context(w)|w)$ is the conditional probability.

After using the skip-gram model to get a 128-dimensional word vector, the sentiment classification of text can be performed. In the selection of classification approaches, this paper uses the bidirectional long and short-term memory (BiLSTM) network. The Weibo text data contains certain temporal relationships, and the use of LSTM can effectively capture the long-term dependencies in the sequence [6], and the two-directional LSTM can realize the extraction of the contextual information of the Weibo text, so BiLSTM is more suitable for the text sentiment classification task in this paper.

In LSTM, the forgetting gate is used to select information that needs to be forgotten, and the output f_t is:

$$f_t = \sigma(W_f x_t + U_f h_(t-1) + b_f).$$

The input gate is used to select information that needs to be added, and the output i_t is:

$$i_t = \sigma(W_i x_t + U_i h_(t-1) + b_i).$$

Then, based on f_t and i_t, the new cell state is obtained:

$$\begin{aligned} \widetilde{c_t} &= \tanh(W_c x_t + U_c h_(t-1) + b_c), \\ c_t &= i_t \times \widetilde{c_t} + f_t \times c_(t-1). \end{aligned}$$

The output gate is used to select information that needs to be output:

$$\begin{aligned} o_t &= \sigma(W_o x_t + U_o h_(t-1) + b_o) \\ h_t &= o_t \times \tanh(c_t) \end{aligned}$$

The parameters involved in LSTM and their meanings are given in Table 1.

Table 1: LSTM parameters

Parameter	Hidden meaning
h_{t-1}	The hidden state of the previous moment
x_t	The input at time t
f_t	The output of the forgetting gate
σ	Sigmoid function
W	Weighting matrix
b	Bias
i_t	The output of the input gate
$\widetilde{c_t}$	The candidate cell state at time t
c_t	The cellular state at time t
o_t	The output of the output gate
h_t	The output of the hidden layer

The output state of the BiLSTM is h_t at time t, including outputs in both directions:

$$\begin{aligned} \overrightarrow{h_t} &= LSTM(x_t, \overrightarrow{h_{t-1}}), \\ \overleftarrow{h_t} &= LSTM(x_t, \overleftarrow{h_{t-1}}), \\ h_t &= [\overrightarrow{h_t}, \overleftarrow{h_t}]. \end{aligned}$$

To further realize the classification of key features, this paper combines the attention mechanism [12] with BiLSTM to assign higher weights to words that express richer emotions. The attention mechanism determines the corresponding weight coefficient a_i by calculating the similarity between query and key and gets the target attention value. The process is:

$$\begin{aligned} F(Q,K) &= Q \cdot K, \\ a_i &= softmax(F(Q,K)), \\ Attention(Q,K,V) &= \sum_{i=1}^{L_x} a_i \cdot value, \end{aligned}$$

where L_x is the length of the key-value pair.

For the output state h_t of the BiLSTM layer, it is multiplied with the corresponding weights to get the final output of the BiLSTM-attention method:

$$s = \sum_{t=1}^{T} a_t h_t.$$

The output is input to the softmax layer for classification:

$$y_i = softmax(ws + b),$$

where w and b are the weight and bias, y_i is the probability distribution that the text belongs to category i. Finally, the steps of the BiLSTM-attention method are shown below.

1) The raw data is collected and cleaned, followed by word segmentation and stop word elimination.
2) A 128-dimensional word vector is obtained by training using the Skip-gram model.
3) Deep features are extracted using the BiLSTM-attention method.
4) The softmax classifier is used to get the final classification result.

2.3 Monitoring of Sudden Online Public Opinion

Based on the categories of the sentiment orientation of Weibo text, the monitoring indicator system of sudden online public opinion was established, as presented in Table 2.

Table 2: Indicator system

Primary indicator	Secondary indicator
Public opinion heat	Topic reading volume
	Topic discussion volume
	Topic interaction volume
	Topic originality volume
The subject of public opinion	Official response
	The sentiment orientation of Internet users
Public opinion event	Percentage of images propagated
	Percentage of videos propagated
	Peak propagation speed

According to Table 2, this paper mainly considers aspects of public opinion heat and subject, and the specific interpretation of the indicators is as follows.

Public opinion heat: It refers to the mechanism of Weibo topics, and the relevant four indicators are as follows:

1) Topic reading volume: the total number of times the content under a topic has been read;
2) Topic discussion volume: the total volume of posts under the topic (including original posts and reposted posts);
3) Topic interaction volume: the total number of interactions under the topic, including reposts, comments, and likes;
4) Topic originality volume: the total number of original posts under the topic.

Subjects of public opinion: The subjects include officials and netizens, and the details are as follows:

1) Official response: whether the government officially responded to the incident, yes = 1, no = 0;
2) Netizens' sentiment orientation: it refers to the sentiment orientation of netizens contained in the text of Weibo posts, which is obtained by the BiLSTM-attention method.

Public opinion events: The faster an event spreads, the faster it develops. The more pictures and videos included in the dissemination of the event, the clearer the description of the event, and the more it will help the public understand the truth. The details are:

1) Percentage of images: it is quantified by the proportion of images to original posts;
2) Percentage of videos disseminated: it is quantified by the proportion of videos to original posts;
3) Peak propagation rate: it refers to the maximum amount of posts posted per hour.

For all of the above indicators, the entropy weight approach [4] was used to determine the indicator weights, and then the risk level of public opinion events was calculated based on the gray correlation analysis (GRA) [11]. According to the nine indicator values in Table 2, the indicator matrix is obtained:

$$X = \begin{bmatrix} x_{11} & \cdots & x_{19} \\ \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{n9} \end{bmatrix}$$

Then, standard data array X_c needs to be determined using the optimal value of the i-th indicator in the indicator matrix:

$$X_c = (x_c(1), x_c(2), \cdots, x_c(9)).$$

$x_c(1)$ is the optimal value of the first indicator, and so on.

The indicator values are normalized to obtain a standardized matrix $\underline{X}$:

$$\overline{X} = \begin{bmatrix} \overline{x_{11}} & \cdots & \overline{x_{19}} \\ \cdots & \cdots & \cdots \\ \overline{x_{n1}} & \cdots & \overline{x_{n9}} \end{bmatrix}$$

The correlation factor of comparative data arrays X_i and X_c is calculated:

$$\delta_i(k) = \frac{\min_i \max_k |X_c(k) - X_i(k)| + \gamma \min_i \max_k |X_c(k) - X_i(k)|}{|X_c(k) - X_i(k)| + \gamma \min_i \max_k |X_c(k) - X_i(k)|}$$

where γ is the resolution, generally taken as 0.5.

Finally, the correlation is calculated:

$$\varphi_i = \sum_{k=1}^{9} w_k \delta_i(k),$$

where w_k is the weight.

Using the k-means algorithm [9], the obtained correlation values are divided into different clusters to realize the grading of public opinion risk. In this paper, the number of clusters was taken to be 3, and the risk of public opinion was classified as slight level, warning level, and serious level. Based on the above, sudden online public opinion was monitored using a back-propagation neural network (BPNN) [15]. The standard BPNN structure was used, the input nodes were the nine indicator values in Table 2, and the output nodes were the three risk levels, which were expressed as slight level (100), warning level (010), and serious level (001). Aiming at the sensitivity of BPNN to the initial parameters, the improved sparrow search algorithm (ISSA) was designed for optimization, as follows:

1) The BPNN parameters were initialized. The sparrow population were initialized using Cubic chaos mapping:

$$x_{n+1} = \rho x_n (1 - x_n^2),$$

where ρ is the control parameter. When $x_0 = 0.3$ and $\rho = 2.59$, it has good chaotic convenience.

2) The position of finder in the population was updated:

$$M_{i,j}^{t+1} = \begin{cases} M_{i,j}^t \cdot \exp(-\frac{i}{\alpha \cdot iter_{\max}}), & R < ST \\ M_{i,j}^t + QL, & R \geq ST \end{cases}$$

3) The position of follower in the population was updated:

$$M_{i,j}^{t+1} = \begin{cases} Q \cdot \exp(M_w^t - M_{i,j}^t), & i > \frac{n}{2} \\ M_b^{t+1} + |M_{i,j}^t - M_b^{t+1}| \\ \qquad \cdot A^+ \cdot L, & i \leq \frac{n}{2} \end{cases}$$

4) The position of vigilante in the population was updated:

$$M_{i,j}^{t+1} = \begin{cases} M_b^t + \beta \cdot |M_{i,j}^t - M_b^t|, & f_i > f_g \\ M_{i,j}^t + K \cdot [\frac{|M_{i,j}^t - M_w^t|}{f_i - f_w + \epsilon}] & f_i = f_g \end{cases}$$

5) They were continuously updated until the maximum number of iterations was reached, and the optimal parameters of the BPNN were output.

The parameters involved in the ISSA and their meanings are given in Table 3.

Table 3: ISSA parameters

Parameter	Hidden meaning
t	The number of iterations
$iter_{max}$	The maximum number of iterations
$M_{i,j}$	The location of individual sparrows
α	A random number in (0,1)
R	An early warning value in (0,1)
ST	A transfer threshold in (0,1]
Q	A random number
L	An all-one matrix with multi-dimensional in one row
M_b	The position with the poorest fitness
M_w	Current optimal position
A^+	$A^+ = A^T(AA^T)^{-1}$, where A stands for a matrix with multi-dimensional in one row
β	Step-size control parameter
f_i	The fitness of the current individual
f_g	The current global optimal fitness
f_w	The current global worst fitness
K	A random number
ϵ	A constant

3 Analysis of Results

3.1 Experimental Setup

The experiments were conducted on the Windows 10 operating system, using the Python programming language. The learning rate of the BiLSTM-attention method was taken as 0.001, The batch size was taken as 16, and the Adam optimization algorithm was adopted. The hidden layer of BPNN was calcualted based on the empirical formula: $p = \sqrt{m+n} + a$, where m and n are the quantity of input and output nodes and a is a constant between 1 and 10. The value of a was determined to be 7 by the trial-and-error method, and the final BPNN structure was 9-7-3. The training error was taken as 0.00001. As for the experimental data, some sudden online public opinion events of 2023 were crawled on Weibo as the subject of analysis through the crawler data, and 100 events were selected. Table 4 shows some of the raw data.

Taking the data of #Datong issues a report on the investigation of bullying incidents involving minors# as an example, 15,241 posts and 32,514 comments were crawled, which were classified into 18,262 positive data and 29,493 negative data using manual labeling. They were cleaned as a dataset for text sentiment classification experiments, and the ratio of data in the training set and test set was 8:2.

Table 4: Some raw data on public opinion events

Serial number	Event	Topic reading volume	Topic discussion volume	...	Peak propagation speed
Y1	#6.1 magnitude earthquake in Xinjiang Shaya#	96.846 million	7,501	...	807/hour
Y2	#6 dead, 12 injured in vehicle collision in Jiuquan, Gansu#	150 million	2,844	...	190/hour
Y3	#Vendor of vegetables earning 21 yuan was fined 110,000 yuan#	450 million	26,000	...	303/hour
Y4	#Datong issues a report on the investigation of bullying incidents involving minors#	1.4 billion	186,000	...	3,261/hour
...	...	...	...	...	...
Y100	#Eleven people were killed in a coal mine accident in Heilongjiang#	71.299 million	3,023	...	473/hour

3.2 Sentiment Orientation Classification Results

Sentiment orientation was classified using the BiLSTM-attention method. This method was compared with the following methods:

1) Support vector machine (SVM) [8];

2) BPNN [7];

3) Convolutional neural network (CNN) [5].

The results were compared in Table 5.

Table 5: Comparison of the results of categorization of sentiment orientation

	Accuracy	Recall rate	F1 value
SVM	0.618	0.714	0.663
BPNN	0.731	0.723	0.727
CNN	0.773	0.732	0.752
BiLSTM-attention	0.836	0.789	0.812

From Table 4, it can be found that the BiLSTM-attention method had a better performance in classifying the sentiment orientation of Weibo text compared with the SVM, BPNN, and CNN methods. Specifically, in terms of accurary, the BiLSTM-attention method was 0.836, which improved 0.218/0.105/0.063 compared to the SVM/BPNN/CNN methods respectively. In terms of recall rate, the BiLSTM-attention method was 0.789, which improved 0.075/0.066/0.057 compared to the SVM/BPNN/CNN methods. The F1 value of the BiLSTM-attention method was 0.836, which improved 0.149/0.085/0.060 compared to the SVM/BPNN/CNN methods. These results proved the effectiveness of the BiLSTM-attention method in the categorization of sentiment orientation in texts, providing subsequent monitoring of sudden online public opinion with reliable data support.

3.3 Results of Monitoring of Sudden Online Public Opinion

The weights of the indicators were computed according to the entropy approach, and the outcomes obtained are displayed in Table 6.

Table 6: Indicators and weights

Indicator	Weight
Topic reading volume	0.056
Topic discussion volume	0.041
Topic interaction volume	0.042
Topic originality volume	0.078
Official response	0.366
Netizens' sentiment orientation	0.274
Percentage of images disseminated	0.037
Percentage of videos disseminated	0.048
Peak propagation speed	0.058

The correlation of each event was calculated based on GRA, and some results are displayed in Table 7.

The events in Table 7 were divided into three clusters using the k-means algorithm and labeled with risk levels. Then, the indicator values and clustering results were composed into sample data, and the training and testing sets were divided by 8:2 to analyze the performance of the ISSA-BPNN method in monitoring sudden online public opinion, and the outcomes obtained on the testing set are presented in Table 8.

In Table 8, among the 20 events tested, the monitoring results of the traditional BPNN for four events were not

Table 7: Correlation values of sudden online public opinion events

Serial number	Correlation	Serial number	Correlation
Y1	0.6525	Y11	0.6582
Y2	0.6755	Y12	0.5214
Y3	0.8541	Y13	0.5286
Y4	0.9756	Y14	0.6258
Y5	0.5261	Y15	0.7745
Y6	0.7451	Y16	0.6251
Y7	0.2514	Y17	0.5216
Y8	0.8544	Y18	0.5628
Y9	0.6258	Y19	0.4485
Y10	0.7452	Y20	0.4521

in line with the reality: event Y15, which should be a serious level, and the BPNN monitoring result was a warning level; event Y23, which should be a slight level, and the BPNN monitoring result was a warning level; event Y48, which should be a slight level, and the BPNN monitoring result was a warning level; event Y91, which should be a warning level, the BPNN monitoring result was serious level, and the overall monitoring accuracy was 80%. In contrast, the ISSA-BPNN method only deviated from the actual monitoring result on time Y91, which should be a warning level, and the monitoring result was a serious level, with an overall monitoring accuracy of 95%. According to the monitoring results, it can be found that the BPNN optimized by the ISSA had better performance in monitoring sudden online public opinion, and it could make more accurate judgments on the risk level of public opinion according to the established indicators, which can be further applied in practice.

4 Management of Sudden Online Public Opinion Under Legal Perspective

In the current online environment, while online public opinion uncovers the truth and reveals the contradictions of reality, it also poses a great challenge to the government's law-based management of sudden online public opinion. Sudden online public opinion is usually characterized by mobbing, episodic, and eruption. When it erupts completely, public opinion provides guidelines for actions in the real world on one hand, while on the other hand, the guidance from the real world in turn influences public opinion. Actions taken in response to public opinion can either calm or further fuel its development.

Through the monitoring of sudden online public opinion, it is possible to judge the risk level of the public opinion at that time based on the analysis of big data, thus providing certain guidance for the next management and action. However, in the current legal perspective, there are still certain deficiencies in the management of sudden online public opinion. For example, govern ment information disclosure is lagging, failing to fully safeguard the citizens' right to know, and the traditional "blocking, plugging, and deleting" approach is often used in solving problems, which may easily lead to the aggravation of public opinion. Moreover, the absence of specialized laws, the lack of citizens' legal awareness, and the lack of punishment for related acts have led to an unclear boundary between freedom of expression and rumor-mongering and defamation, and netizens' lack of ability to identify rumors and spread them indiscriminately have led to further complication of the incident. From a legal perspective, the following actions can be taken to further realize the management of sudden online public opinion:

Strengthening legislation: Since 2020, laws and regulations concerning the management and dissemination of online information content have been regulated by the following laws:

1) The Data Security Law of the People's Republic of China (2021), which makes several provisions for the standardized processing, exploitation, and utilization of data;
2) The Measures for the Administration of Internet Information Services (2021), which provides some safeguards to facilitate the sound growth of Internet information services;
3) The Regulations on the Management of Internet User Account Information (2022), which provide provisions for Internet users to register and use account information.

Based on the existing laws, it is necessary to further improve and refine them and to further clarify cybercrime, taking into account the current stage of the national situation, to ensure that the relevant parts of the law can be complied with, to strengthen the legal constraints on citizens, and to ensure that public opinion is positively oriented.

Strengthening law enforcement: In the process of law-based management of sudden online public opinion, it is necessary to clarify the main body of law enforcement, avoid mutual shirking of responsibilities, formulate specific accountability standards, and prompt the relevant departments to take the initiative to assume responsibility. At the same time, it should correctly carry out information disclosure, proactively disclose the key information and details of the incident, clearly define the responsibility of creating or spreading rumors for sentencing, and cultivate specialized network law enforcement talents according to the actual situation to monitor public opinion, intervene promptly, and guide public opinion to healthy development.

Table 8: Monitoring results of sudden online public opinion (bold indicates discrepancies with actual results)

Event number	Actual result	The monitoring result of BPNN	The monitoring result of ISSA-BPNN
Y3	(0 0 1)	(0 0 1)	(0 0 1)
Y4	(0 0 1)	(0 0 1)	(0 0 1)
Y7	(1 0 0)	(0 1 0)	(1 0 0)
Y10	(0 1 0)	(0 1 0)	(0 1 0)
Y15	(0 0 1)	(0 1 0)	(0 0 1)
Y19	(1 0 0)	(1 0 0)	(1 0 0)
Y23	(1 0 0)	(0 1 0)	(1 0 0)
Y28	(0 1 0)	(0 1 0)	(0 1 0)
Y32	(0 0 1)	(0 0 1)	(0 0 1)
Y37	(0 1 0)	(0 1 0)	(0 1 0)
Y41	(0 0 1)	(0 0 1)	(0 0 1)
Y48	(1 0 0)	(0 1 0)	(1 0 0)
Y51	(0 1 0)	(0 1 0)	(0 1 0)
Y55	(0 1 0)	(0 1 0)	(0 1 0)
Y63	(1 0 0)	(1 0 0)	(1 0 0)
Y74	(0 0 1)	(0 0 1)	(0 0 1)
Y85	(0 1 0)	(0 1 0)	(0 1 0)
Y87	(1 0 0)	(1 0 0)	(1 0 0)
Y91	(0 1 0)	(0 0 1)	(0 0 1)
Y93	(0 0 1)	(0 0 1)	(0 0 1)

Strengthening media self-regulation: The law-based management of sudden online public opinion also requires the media to consciously report under social ethics, disseminate mainstream values, strengthen the punishment of false and exaggerated reports, enhance the legal awareness training for media practitioners, increase the vetting efforts, control the orientation of the event reports, and prohibit the media from utilizing the emotions of netizens for speculation.

5 Conclusion

Based on big data, this paper analyzed the monitoring and law-based management of sudden online public opinion, took Weibo as an example, established the monitoring indicator of sudden online public opinion based on the calculation of the sentiment orientation of netizens' text, and designed the ISSA-BPNN method to realize the monitoring of sudden online public opinion. After experimental comparison, it was found that both the BiLSTM-attention method and the ISSA-BPNN method both had better performance. They can be applied in practice.

References

[1] S. Bankapur, N. Patil, "An enhanced protein fold recognition for low similarity datasets using convolutional and skip-gram features with deep neural network," *IEEE Transactions on Nanobioscience*, vol. 20, no. 1, pp. 42-49, 2021.

[2] M. E. Barachi, M. Alkhatib, S. Mathew, F. Oroumchian, "A novel sentiment analysis framework for monitoring the evolving public opinion in real-time: Case study on climate change," *Journal of Cleaner Production*, vol. 312, no. 5, pp. 127820, 2021.

[3] Q. Cheng, Y. G. Zhang, Y. Q. Li, "Topic relevance of public health emergencies influence on internet public opinion resonance: simulation based on langevin's equation," *Mathematical Problems in Engineering*, vol. 2021, pp. 1-15, 2021.

[4] F. Deng, L. Yang, Q. Gong, G. Guo, "Evaluation of high-quality economic development based on entropy weight method: taking chengdu-chongqing city agglomeration as example," in *Proceedings of the 2021 4th International Conference on Computers in Management and Business (ICCMB'21)*, pp. 41-47, 2021.

[5] O. Ferraz, H. Araujo, V. Silva, G. Falcao, "Benchmarking convolutional neural network inference on low-power edge devices," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'23)*, pp. 1-5, 2023.

[6] E. N. Furqon, B. Dirgantoro, C. Setianingsih, "Predict buy and sell time for stock price using convolutional neural network and long short-term memory," in *4th International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR'21)*, pp. 138-143, 2021.

[7] D. Guo, J. Wang, S. Li, "Research on short-term traffic demand of taxi in large cities based on BP neural

network algorithm," in *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA'21)*, pp. 1161-1166, 2021.

[8] V. Selvaraju, P. A. Karthick, R. Swaminathan, "Detection of preterm birth from the noncontraction segments of uterine emg using hjorth parameters and support vector machine," *Journal of Mechanics in Medicine and Biology*, vol. 23, no. 06, 2023.

[9] C. J. Swinney, J. C. Woods, "K-means clustering approach to UAS classification via graphical signal representation of radio frequency signals for air traffic early warning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24957-24965, 2022.

[10] M. Tang, "Design of visual model and solutions for public opinion monitoring and analysis for big data," in *Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS'21)*, pp. 295-301, 2021.

[11] C. Valmohammadi, F. F. Razi, F. Einy, "Six sigma project selection using the hybrid approach FAHP-FTOPSIS and grey relational analysis model," *IEEE Engineering Management Review*, vol. 49, no. 2, pp. 134-146, 2021.

[12] H. Wang, J. Yang, R. Wang, L. Shi, "Remaining useful life prediction of bearings based on convolution attention mechanism and temporal convolution network," *IEEE Access*, vol. 11, pp. 24407-24419, 2023.

[13] X. Wang, C. Ma, S. Chen, "Detecting opinion evaluation object drift in Chinese microblog posts," *Proceedings of the Association for Information Science and Technology*, vol. 56, no. 1, pp. 794-795, 2019.

[14] J. Xie, S. Zhang, L. Lin, "Prediction of network public opinion based on bald eagle algorithm optimized radial basis function neural network," *International Journal of Intelligent Computing and Cybernetics*, vol. 15, no. 2, pp. 184-200, 2022.

[15] Q. Zeng, H. Jiang, Q. Liu, G. Li, Z. Ning, "Design of a high-temperature grease by BP neural network and its preparation and high-temperature performance studies," *Industrial Lubrication and Tribology*, vol. 74, no. 5, pp. 564-571, 2022.

[16] W. Zhang, J. Ji, Y. Li, X. Wang, J. Liu, "Prediction of public opinion early warning level of medical dispute cases based on ontology," in *IEEE International Conference on Power Electronics, Computer Applications (ICPECA'21)*, pp. 361-366, 2021.

[17] Z. Zhao, "Intelligent analysis and positioning of political public opinion in universities," *International Journal of Artificial Intelligence Tools: Architectures, Languages, Algorithms*, vol. 31, no. 4, pp. 1-17, 2022.

Biography

Fan Tu, born in December 1988, holds a Doctor of Law degree. She graduated from East China University of Political Science and Law in September 2020. Currently, She works as a lecturer at Northwest Normal University. She is interested in comparative law and artificial intelligence law.

Research on English Data Security Aggregation Based on Neighbor Propagation Clustering

Jianhou Nie

(Corresponding author: Jianhou Nie)

School of Foreign Languages, Zhengzhou University of Science and Technology
Zhengzhou 450064, China
Email: zzll_201@foxmail.com

(Received Oct. 26, 2023; Revised and Accepted Feb. 7, 2024; First Online Apr. 25, 2024)

The Special Issue on Data Fusion, Deep Learning and Optimization Algorithms for Data Privacy Protection
Special Editor: Prof. Shoulin Yin (Harbin Institute of Technology)

Abstract

There are some problems in the existing security aggregation schemes. For example, most schemes do not take into account the privacy of the global model, and cloud servers that know the plaintext of the global model can infer the privacy of users from the global model. The use of costly public key encryption technology to protect the privacy of the global model and cannot support user drop off, which is not conducive to resource-limited users to participate in training. If users quit training due to network reasons, the whole training process will be affected. In addition, the existing scheme does not consider the integrity of the data during the interaction, which makes the original data uploaded by the user cannot be obtained by other users and the cloud server. Therefore, we propose a novel English data security aggregation model based on neighbor propagation clustering. A clustering structure based on neighbor propagation is constructed, and each data point is assigned to the nearest representative point, so as to build the similarity matrix between daily scene samples, which is used as the standard to judge the clustering center. An anti-vibration attenuation factor is introduced, the similarity matrix is initialized, and the optimal cluster number is determined. The security analysis shows that the new scheme is provably safe, and the performance analysis shows that the new scheme is efficient and practical.

Keywords: Anti-Vibration Attenuation Factor; Data Security Aggregation; Neighbor Propagation Clustering; Similarity Matrix

1 Introduction

After years of development of college informatization, many colleges and universities have accumulated a large number of business and related data. Data is different from tangible products, but it also has the concept of quality, but because of the low quality of data, the existence of a large number of redundant data and the availability of data is not high, so that a lot of data processing work still needs to be manual to ensure the correct, wasting a lot of manpower and time [5, 16, 17].

In order to extend the life cycle of the data network and reduce the transmission of redundant data, it is necessary to collect and process raw data collaboratively in the network to reduce the amount of raw data sent. Data aggregation technology is one of the commonly used methods. In recent years, many researchers have proposed related data security aggregation protocols. He *et al.* [7] proposed the cluster-based private data aggregation (CPDA) protocol, its core idea was to use the nature of cluster protocol and polynomial algebra to protect privacy in data aggregation. Wireless sensor nodes were randomly grouped into clusters. In each cluster, the addition property of polynomials was used to calculate the aggregate result. The protocol had the disadvantage of high computing and communication overhead. Cominetti *et al.* [3] proposed the additively homomorphic encryption (AHE) protocol, which allowed devices to perform data aggregation on ciphertext and used an end-to-end encryption mechanism to achieve data aggregation. Li *et al.* [9] proposed the slice-mix-aggregate (SMART) protocol, which was mainly based on the correlation attribute of segmentation technology and additive. Each node encrypted the raw perception data by cutting it into data fragments. IPHCDA (integrity protecting hierarchical concealed data aggregation) protocol [14] used a homomorphic encryption algorithm based on elliptic curve encryption to provide data integrity and confidentiality. This protocol supported integrity verification, had higher security and high accuracy of aggregation results. Its disadvantage was that it was expensive in computation and communication.

Hahn *et al.* [6] proposed a verifiable security aggregation protocol to verify the correctness of the aggregation

results returned by the cloud server. However, during the verification process, the communication cost would increase linearly with the gradient dimension, resulting in poor system performance. Guo *et al.* [2] used the commitment scheme and linear homomorphic hash function to verify the correctness of the aggregate results and improve the computational efficiency of the system. Reference [10] used ElGamal homorphic encryption technology combined with Diffie-Hellman key exchange protocol and Shamir secret sharing algorithm to propose a scheme that could tolerate user drop and resist participant collusion attacks. The above schemes only consider the privacy of the user's local model. Reference [18] pointed out that cloud servers with known global models could infer users' privacy information through model inversion attacks, which could reduce system security. Reference [12] used full-homomorphic encryption to protect the privacy of the global model. Reference [15] used Paillier homomorphic encryption technology combined with bilinear aggregate signature to verify the correctness of the aggregate result returned by the server, but the protocol [13] failed to support users to exit the system at any time. Due to the large amount of English data and the high similarity between classes, the data cannot be securely aggregated using the current methods. Therefore, we propose a novel English data security aggregation model based on neighbor propagation clustering.

2 Preliminaries

2.1 Homomorphic Encryption

Homomorphic encryption algorithm can calculate ciphertext data directly without decryption. Suppose $E_K()$ is an encryption function, K is the encryption key, and $D_K()$ is the corresponding decryption function. If there is a valid algorithm Alg that satisfies $Alg(E_K(x), E_K(y)) = Ex(x \circ y)$ under an operation, then $E_K()$ is a homomorphic encryption under an operation $\circ$.

2.2 Elliptic Curve ElGamal Cryptosystem

ElGamal cryptosystem is a typical public key cryptosystem [11]. The ElGamal cryptosystem based on elliptic curves has the property of addition homomorphism. The security of elliptic curve ElGamal algorithm is based on elliptic curve discrete logarithm problem. Elliptic curve discretization problem is that for the equation $Q = kP$, we know that it is easier to calculate Q if we know k and P. On the other hand, it is harder to compute k if Q and P are known. For an elliptic curve E over a finite field, let $q = p^n$, $n = 2n'$ and the plaintext message be an integer m, then m can be expressed as $m = m_0 + m_1 p + \cdots + m_{n-1} p^{n-1}$. When using ElGamal encryption, the message m is encoded on the elliptic curve with the encoding function $map()$, and the point on the elliptic curve is decoded with the decoding function $rmap()$ when decrypting.

2.3 BLS Signature

BLS signature [4] is a digital signature algorithm based on bilinear mapping, which has the advantages of short signature, short public key, high security, anonymous authentication, etc., and can aggregate multiple signatures into one signature, reducing the communication overhead and computing overhead of the system. The technique consists of the following algorithms.

- Initialization algorithm. The signer selects the bilinear map $e : G_1 \times G_2 \to G_T$, and the hash function $h : 0, 1^* \to G_2$.
- Key generation algorithm. The signer randomly selects the private key x and calculates the public key $y = g^x \in G_1$.
- Signature algorithm. The signer uses x and message $M_i \in 0, 1^*$ to calculate $h = h(M)$ and signature $\sigma = h^x$.
- Verification algorithm. Given the signer's public key y, message M, and signature σ. The verifier calculates $h = h(M)$, and accepts if $e(g_1, \sigma) = e(y, h)$ holds; Otherwise, refuse.

3 Proposed Data Security Aggregation

Through comparison with the clustering results of K-means clustering, hierarchical clustering and other clustering algorithms, it can be seen that the nearest neighbor propagation method can transform the clustering results into clusters with potential clustering centers, so that the clustering results will not take too long to find the centers, and the clustering results will be more stable through the predetermined number of clusters.

In order to construct the similarity matrix, it is necessary to collect scene samples in the English data. Under the condition that all samples are collected, the similarity matrix is constructed as shown in Equation (1):

$$W = \begin{bmatrix} W(q_1q_1) & \cdots & W(q_1q_i) & \cdots & W(q_1q_{365}) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ W(q_iq_1) & \cdots & W(q_iq_i) & \cdots & W(q_iq_{365}) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ W(q_{365}q_1) & \cdots & W(q_{365}q_i) & \cdots & W(q_{365}q_{365}) \end{bmatrix} \quad (1)$$

In the matrix of Equation (1), q_i represents the scene sample. The non-diagonal parameter represents the Euclidean distance in the day scene, that is, the true distance between two samples in m-dimensional space. In everyday scenarios, the diagonal element can be used as a measure of the cluster center, so it is set as a reference value. The reference value has a great influence on the

number of clustering results. Compared with other clustering methods, the nearest neighbor propagation clustering algorithm determines whether the daily scene sample can be used as the clustering center by constructing the similarity matrix. If appropriate, it can be used as a clustering center based on nearby propagation clustering. On the contrary, if it is not suitable, it is necessary to construct the similarity matrix by re-counting the daily scene samples, and then select the scene samples suitable for the clustering center to achieve the accurate determination of the clustering center.

Combined with the clustering center determined above, in order to avoid the shock in the iteration process, the current iteration results are compared and analyzed with the previous iteration results to obtain the iteration update results. Using the nearest neighbor propagation clustering method, the English data security aggregation process is designed as follows:

Step 1: Initializing the similarity matrix. For the similarity matrix initialization, the diagonal elements should be regarded as the same value, and the values of confidence and reference values should be set to 0 without prior knowledge, thus completing the initialization of the similarity matrix.

Step 2: Determining the optimal clustering number. The intra-class index reflects the clustering effectiveness of individual samples. If the intra-class index value is larger, the clustering effect of individual samples is better. On this basis, the data set is statistically analyzed, and the clustering results are compared, and the average value is used as the clustering index. The larger the mean value is, the more obvious the clustering effect of the data set will be. The maximum value of the mean value is the optimal cluster number.

On this basis, the classification index based on distance measure is used to analyze the validity of the clustering results. The average inter-class and intra-class division index value of data agglomeration class is as follows:

$$avgk(i) = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{n_j} k(j,i). \tag{2}$$

In Equation (2), $k(j,i)$ represents the inter-class and intra-class representation data set. The optimal number of clusters is calculated according to equation (2), which is as follows:

$$\beta = arg\max_{i=1} avgk(i). \tag{3}$$

The optimal clustering number can be determined by equation (3).

Step 3: Update the clustering results. The updating process of clustering results is as follows:

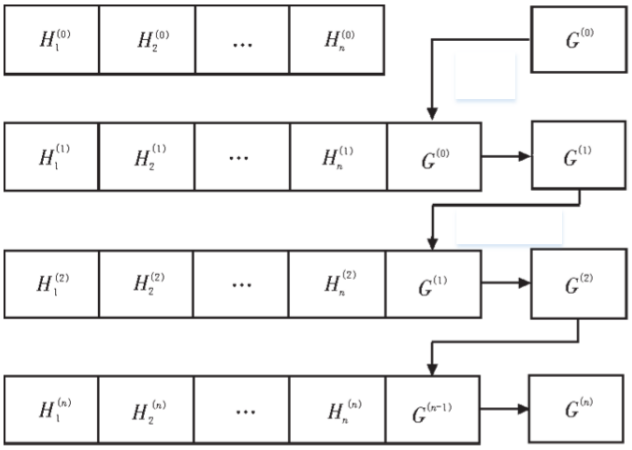


Figure 1: Attenuation model

1) Initializing K clusters and setting the weight of each cluster center to 0.
2) After reading N text data, the cluster center weight of each text data is set to 1, and N text data is merged with K cluster centers, and the update result of cluster centers is obtained by using the nearest neighbor propagation algorithm. When updating the cluster center, the new cluster results need to be weighted to obtain the newly added data. The formula can be expressed as follows:

$$y_n = arg\max_{k} x_n^t \delta_k. \tag{4}$$

In Equation (4), δ_k is the sum of new data and historical data.

3) The higher the weight of the new cluster center, the greater its proportion. When $N+K$ group data is aggregated into a new cluster center of K group, weight attenuation calculation will be performed for each new cluster center, as shown in Figure 1.

In Figure 1, H represents the newly added data; G indicates historical data. A new attenuation weight of cluster center can be obtained by counting new data, historical data and setting attenuation coefficient.

4) Repeat Step 2 until the data process is completed or terminated manually.

Step 4: Output clustering results. Since the K class cannot be used as the input parameter of the classifier directly, the clustering results of K clusters can be obtained. According to the output clustering results, the English data security aggregation process is designed, as shown in Figure 2.

As can be seen from Figure 2, each node is regarded as a collection node, the required minimum similar data is summarized, the least node is selected as the

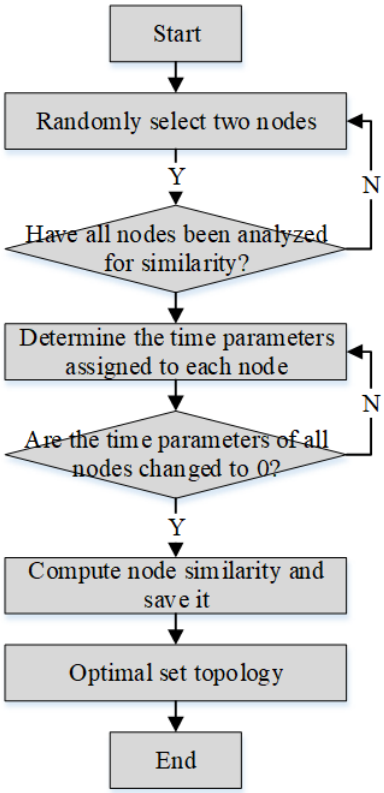


Figure 2: Data security aggregation process

collection node, and the data of the node is forwarded to the optimal collection topology generated by the node, so as to complete the security aggregation of data.

Each data feature weight in English data will be assigned a different value, and the value size needs to be calculated according to the importance of its feature in the region. After the data set representation is complete, the similarity between the different data is analyzed. The judging formula is:

$$sim(i,j) = \frac{\sum_{k=1}^{m}(w_i \cdot w_j)}{\sqrt{\sum_{k=1}^{m} w_i^2}\sqrt{\sum_{k=1}^{m} w_j^2}}. \quad (5)$$

In Equation (5), w_i and w_j represent the weights of data i and j respectively. When solving the minimum similarity of each node, each node has a time parameter, so the node with the least similarity can be found first. When the network topology is set to 0, there is no need to transmit link information to nodes in packets, thus realizing the secure aggregation of English data.

4 Security Analysis

- Confidentiality. Since the user uses symmetric homomorphic encryption technology to encrypt the gradient, CS cannot get the global model parameters by operating only on the ciphertext. Therefore, the proposed scheme provides confidentiality:

- Authentication. User i uses his own identity to register with TA in advance, and after sending data to other users and CS, user j and CS realize authentication of user identity while verifying data signature.

- Integrity. Using the BLS signature technology, user i signs ciphertext C_i and sends (σ_i^1, C_i) to CS. Since the opponent cannot know x_i, it cannot generate a legal σ_i^1. If an adversary forges a signature or modifies the data content, it will be detected when the signature is verified. Therefore, the scheme realizes data integrity protection in the interaction process.

- Resist multi-user collusion attacks. Using Shamir secret sharing technology, when less than one user colludes, the mask value used in the user mask gradient ciphertext cannot be obtained. Therefore, the proposed scheme can resist the collusion attack of multiple users.

- Resist collusive attacks between users and cloud servers. The malicious user and CS conspired to obtain the gradient ciphertext, but because of the existence of the mask, the malicious user could not get the true gradient value. Therefore, the proposed scheme can resist the collusive attack between the user and the cloud server.

5 Performance Comparison

This section gives a functional comparison between the proposed scheme and related schemes including SECPDA [4], EEDAM [1], as shown in Table 1. F1 represents global model privacy; F2 indicates resistance to collusive attacks; F3 stands for verifiability; F4 indicates data integrity; F5 stands for dropped call robustness. SECPDA could resist collusive attacks and support user dropouts, but it failed to protect the global model privacy and fails to verify the correctness of aggregate results. EEDAM protected the privacy of the global model and supported the correctness verification of the aggregated results, but this scheme could not support user drop-off. At the same time, the above schemes fail to protect the data integrity in the interaction process. The proposed scheme in this paper can satisfy all the above functions.

Based on the Charm cryptographic library and Java language simulation tests, this section tested relevant cryptographic operations and their execution time, as shown in Table 2. The experimental environment is i7-7700HQ (2.80GHz) CPU and 64-bit Ubuntu operating system with 4GB memory. Based on the security of 128 bits, both the proposed scheme and A use bilinear mapping. We select bilinear mapping $e : G_1 \times G_2 \to G_T$. G_1 is a q-order cyclic group, and q is a prime number of 512 bits. Let n represent the number of users participating in the training process, n_d represent the number of users

Table 1: Function comparison

Function	SECPDA	EEDAM	Proposed
F1	No	Yes	Yes
F2	Yes	No	Yes
F3	No	Yes	Yes
F4	No	No	Yes
F5	Yes	No	Yes

dropping out during the training, and $t = 0.6n$ represent the threshold of the secret sharing protocol. Table 3 shows the calculation cost comparison between the proposed solution and A and B at the client side and the cloud server side when the number of dropped users $n_d = 0.3n$.

Table 2: Cryptographic operation execution time/ms

Symbol	Decryption	Time
T_{Pair_G}	Bilinear pair operation	19.8594
$T_{Mul_{G_1}}$	Multiplication operation under G_1	0.0015
$T_{Exp_{G_1}}$	Exponential operation under G_1	0.7351
$T_{Mul_{G_2}}$	Multiplication operation under G_2	0.0165
$T_{Exp_{G_2}}$	Exponential operation under G_2	1.3257

Figures 3 $\sim$ 4 respectively show the calculation cost comparison between the proposed scheme and SECPDA and EEDAM at the client side and cloud server side as the number of users changes. As can be seen from Figure 3, when the number of users is 50, 100, 150 and 200, the calculation cost of the proposed scheme is reduced by 110.28ms, 220.56ms, 330.84ms and 441.12ms, respectively, compared with SECPDA. It can be inferred that the proposed scheme is better than SECPDA as the number of users increases. It can be seen from Figure 4 that the running time of the proposed scheme is lower than that of EEDAM. After calculation, the proposed scheme is 98.10% lower than that of EEDAM. As can be seen from Figure 3, the running time of the proposed scheme and SECPDA is almost the same. As can be seen from Figure 4, the proposed scheme has obvious advantages compared with EEDAM.

This section gives the communication cost of the proposed scheme and the comparison with the communication cost of SECPDA and EEDAM under the premise of realizing the same function. Based on the security of 128 bits, the size of the elements in group G_1 is defined as 512 bits. The size of elements in G_2 group is 1024 bits. The elements in G_T of the bilinear mapping group are 3072 bits. The value of N is 1024 bits. The size of Z_n^* is 6144 bits. The size of $Z_{n^2}^*$ is 12288 bits, the size of Z_p is 256 bits, and the user ID length is 32 bits. Table 4 shows the communication cost comparison between the proposed scheme and SECPDA, EEDAM. The number of

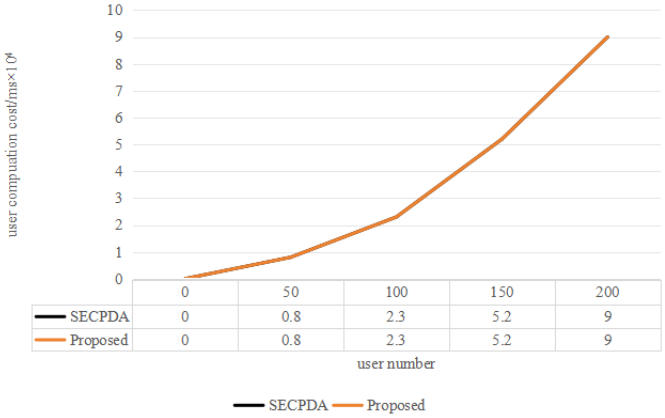


Figure 3: Comparison of calculation cost between proposed scheme and SECPDA

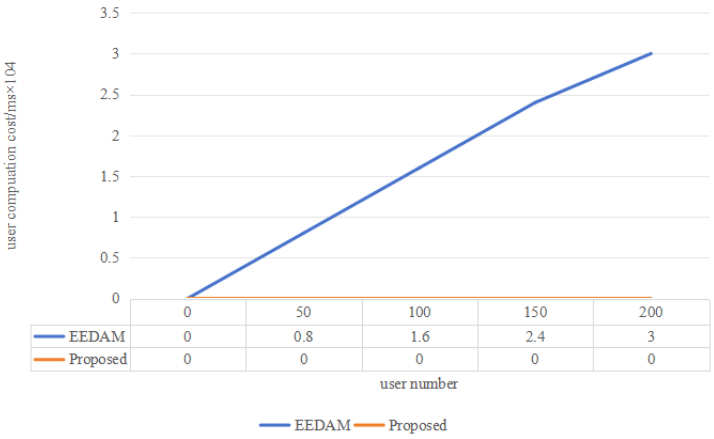


Figure 4: Comparison of calculation cost between proposed scheme and EEDAM

Table 3: Calculation cost comparison

Scheme	The computing cost of the client	Cloud server-side computing costs
SECPDA	$0.0012n^3 + 2.08n^2 + 1.422n$	$0.0003n^3 + 0.0044n^2 + 0.0034n + 0.7324$
EEDAM	$0.0012n^3 + 2.08n^2 - 0.7844n$	$0.0003n^3 + 0.0044n^2 - 0.003n$
Proposed	$2.9625n$	$0.0018n - 0.0003$

users and the number of dropped users in related schemes are n and $0.6n$ respectively.

In summary, the proposed scheme requires less communication cost and is more suitable for resource-constrained data aggregation systems.

6 Conclusions

In this paper, a new security aggregation protocol is proposed using the nearest neighbor propagation clustering technique combined with the double mask protocol. This scheme uses symmetric homomorphic encryption technology to solve the problem of high cost of homomorphic encryption in existing literatures using public key system. At the same time, the nearest neighbor propagation clustering avoids the defect that the bilinear aggregation signature technology cannot resist the user collusion attack and can effectively verify the aggregation results. The security proof shows that the proposed scheme can satisfy the confidentiality, authentication, integrity, and resist the collusion attacks between users and between users and cloud servers, providing a more comprehensive functional guarantee. Performance analysis shows that compared with other literatures, the proposed system is more efficient and can better meet the actual needs.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] A. Ahmed, S. Abdullah, M. Bukhsh, I. Ahmad and Z. Mushtaq, "An energy-efficient data aggregation mechanism for IoT secured by blockchain," *IEEE Access*, vol. 10, pp. 11404-11419, 2022.

[2] D. Boneh, J. Drake, B. Fisch, A. Gabizon, "Halo infinite: Recursive ZK-snarks from any additive polynomial commitment scheme," *Cryptology ePrint Archive*, 2020.

[3] E. L. Cominetti and M. A. Simplicio, "Fast additive partially homomorphic encryption from the approximate common divisor problem," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2988-2998, 2020.

[4] H. Dou, Y. Chen, Y. Yang, Y. Long, "A secure and efficient privacy-preserving data aggregation algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 1495-1503, 2022.

[5] B. Fu, N. Damer, "Face morphing attacks and face image quality: The effect of morphing and the unsupervised attack detection by quality," *IET Biometrics*, vol. 11, no. 5, pp. 359-382, 2022.

[6] C. Hahn, H. Kim, M. Kim and J. Hur, "VerSA: Verifiable secure aggregation for cross-device federated learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 36-52, 2023.

[7] W. He, X. Liu, H. Nguyen and K. Nahrstedt, "A cluster-based protocol to enforce integrity and preserve privacy in data aggregation," *2009 29th IEEE International Conference on Distributed Computing Systems Workshops, Montreal, QC, Canada*, pp. 14-19, 2009. doi: 10.1109/ICDCSW.2009.18.

[8] S. Irawadi, "Nonsingular matrix as private key on ElGamal cryptosystem," *Journal of Physics: Conference Series. IOP Publishing*, vol. 1821, no. 1, 2021.

[9] C. Li, Y. Liu, "ESMART: energy-efficient slice-mix-aggregate for wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 9, no. 12, 2013.

[10] J. Liu, J. Zhang, S. Yin, "Hybrid chaotic system-oriented artificial fish swarm neural network for image encryption," *?Evolutionary Intelligence*, vol. 16, pp. 77-87, 2023.

[11] X. Luo, Z. Zhou, L. Zhong, J. Mao, C. Chen, "An effective integrity verification scheme of cloud data based on BLS signature," *Security and Communication Networks*, vol. 2018, pp. 1-11, 2018.

[12] Z. Min, G. Yang, A. Sangaiah, "A privacy protection-oriented parallel fully homomorphic encryption algorithm in cyber physical systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1-14, 2019.

[13] I. Muhammad, S. Yin, H. Li, S. Karim, A. Laghari, "Comprehensive review of emerging cyber security trends and developments," *International Journal of Electronic Security and Digital Forensics*, 2023. doi: 10.1504/IJESDF.2025.10059222

[14] S. Ozdemir, Y. Xiao, "Integrity protecting hierarchical concealed data aggregation for wireless sensor networks," *Computer Networks*, vol. 55, no. 8, pp. 1735-1746, 2011.

[15] R. Qiu, M. Ai, F. Zheng, L. Liang and Y. Li, "Privacy-preserving of power consumption big

Table 4: Communication cost comparison

Method	User calculation cost	Cloud server-side computation cost	Total communication cost
SECPDA	$768n + 2048$	$1075.2n$	$1843.2n + 2048$
EEDAM	$768n + 512$	$716.8n$	$1484.8n + 512$
Proposed	1536	$1792n$	$1792n + 1536$

data based on improved group signature and homomorphic encryption," *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Shenyang, China*, pp. 25-29, 2020, doi: 10.1109/AUTEEE50969.2020.9315655.

[16] P. Theofilou, "Evaluation of quality of life and fatigue in dialysis patients: The contribution of social support and satisfaction from nursing staff," *World Journal of Nursing Research*, pp. 38-45, 2021.

[17] S. Yin, H. Li, L. Teng, A. A. Laghari, V. V. Estrela, "Attribute-based multiparty searchable encryption model for privacy protection of text data," *Multimedia Tools and Applications*, 2023. ttps://doi.org/10.1007/s11042-023-16818-4

[18] P. Zhang, M. Tang, W. Susilo and M. Zhang, "Efficient non-interactive polynomial commitment scheme in the discrete logarithm setting," *IEEE Internet of Things Journal*, 2023. doi: 10.1109/JIOT.2023.3319338.

Biography

Jianhou Nie biography. Jianhou Nie is with the School of Foreign Languages, Zhengzhou University of Science and Technology. Interests: Data analysis, Security analysis, English teaching, Translation.

RNN-GSW: A Homomorphic Encryption Scheme for Economic Data Based on Recurrent Neural Network and GSW

Yueyue Dong

(Corresponding author: Yueyue Dong)

School of Business Administration & Zhengzhou University of Science and Technology
Zhengzhou City 450000, China
Email: newmansuper@163.com

(Received Oct. 29, 2023; Revised and Accepted Feb. 7, 2024; First Online Apr. 25, 2024)

The Special Issue on Data Fusion, Deep Learning and Optimization Algorithms for Data Privacy Protection

Special Editor: Prof. Shoulin Yin (Harbin Institute of Technology)

Abstract

Existing Brakerski-Gentry-Vaikuntanathan (BGV) multi-key homomorphic encryption algorithms have some problems in resisting quantum attacks and constructing secure multi-party computing schemes. For example, there are some problems such as complex key calculation and large ciphertext size, therefore, a homomorphic encryption scheme for economic data based on recurrent neural network and Gentry-Sahai-Waters (GSW) is proposed. In this scheme, the main operation part is encrypted by GSW homomorphic encryption scheme. Generate economic big data series based on recurrent neural network; The encrypted data is obtained through forward feedback and backward feedback, and the encrypted data packet is replaced by the linear combination of the encrypted source data packet. The generation of shared key and joint decryption are completed by using the existing multi-key homomorphic encryption. Theoretical analysis shows that this encryption scheme can reduce the key size, reduce the complexity of homomorphic multiplication and improve the efficiency of encryption operation. This scheme has strong anti-attack capability and more significant advantage in storage efficiency.

Keywords: Economic Data Security; Homomorphic Encryption; Linear Combination; Recurrent Neural Network

1 Introduction

Homomorphic encryption is a type of encryption technology that allows homomorphic operations on ciphertext without decryption, after which the correct result can be obtained. Homomorphic encryption can be applied to the entrusted calculation [3, 19]. The entrusting party encrypts the data using the public key and transmits the ciphertext to the computing party. The computing party returns the encrypted result to the entrusting party through the corresponding calculation, and the entrusting party decrypts the correct result with the private key.

In many scenarios, data may come from multiple agents. These agents work together to calculate the results without giving other agents access to their own private data. Such problems can be reduced to the problem of secure multi-party computation, and multi-key homomorphic encryption is a solution in such scenarios [25]. The scheme allows each subject to use its own public key for encryption, and then use ciphertext extension to extend the ciphertext from all parties, then perform homomorphic operations on the ciphertext and get the resulting ciphertext, and finally decrypt the correct result by the joint decryption of each subject.

The earliest multi-key homomorphic encryption scheme is LTV12 proposed by Lopez-Alt *et al.* [14] in 2012. Later, Doroz *et al.* [10] improved the LTV12 scheme and proposed a more efficient scheme (DHS16). In 2017, Chongchitmate *et al.* [8] constructed a Multi-Key Fully Homomorphic Encryption (MKFHE) scheme CO17 with circuit privacy based on LTV12. In 2020, Che *et al.* [5] proposed CZL+20, a scheme that did not need to calculate the key, using the bit discarding technology and ciphertext extension technology. This kind of schemes were developed from the Number Theory Research Unit (NTRU) type homomorphic encryption scheme. The difficult problem assumptions based on this paper were non-standard assumptions on polynomial rings, and an efficient joint decryption protocol could not be constructed, so its practical application was limited [17].

In 2015, Clear *et al.* [9] proposed a multi-key homhomic encryption scheme CM15 of GSW (Gentry-Sahai-Waters)

type. The security of the scheme was based on Learning With Errors (LWE) problem. The difficulty of this problem was attributed to the standard assumptions of lattice cryptography. In this scheme, the ciphertext extension technique was proposed for the first time, which could extend a single key homomorphic encryption scheme to a multi-key homomorphic encryption scheme. In 2016, Mukherjee *et al.* [18] simplified the ciphertext extension process of CM15 and proposed the MW16 scheme. This scheme allowed one round of distributed decryption, and could further construct two rounds of secure multi-party computing protocols. Peikert *et al.* [20] proposed PS16, a scheme with multiple hops. This scheme allowed participants to join the calculation process in real time and dynamically, but there was a certain limit on the number of participants. Brakerski *et al.* [4] proposed a fully dynamic scheme BP16. On the basis of PS16, the scheme had no limit on the number of participants, but such schemes had problems such as complex ciphertext expansion and slow calculation.

Chen *et al.* [7] first proposed CZW17, a multi-key homomorphic encryption scheme of BGV (Brakerski-Gentry-Vaikuntanathan). Compared with GSW schemes, BGV schemes had a simple ciphertext expansion mode, and did not need to use data other than ciphertext during the expansion process. Therefore, it could effectively support multi-hop characteristics (adding new users during homomorphic operations). The obvious defects of BGV type multi-key homomorphic encryption schemes were the complexity of computational key generation and large computational key size. In 2019, Li *et al.* [16] used mixed homomorphic multiplication between RBGV (Ring BGV) and RGSW (Ring GSW) ciphertext to generate computational keys, reduced the extended ciphertext size in CZW17, and on this basis designed a multi-key homomorphic encryption scheme LZY+19. Chen *et al.* [6] proposed the CDKS19 scheme. In this scheme, a new computational key generation method was proposed, the process of homomorphic multiplication was improved, and the scheme was applied to the privacy computation of neural networks. However, the key exchange process of BGV multi-key homomorphic encryption scheme was still relatively complicated, and the number and size of keys needed to generate the calculated keys are large. In 2021, Zhou *et al.* [29] proposed a scheme to build multi-key homomorphic encryption using cumulative public keys, however, the decryption process caused the private key to be exposed.

Most of the above multi-key homomorphic encryption schemes have disadvantages such as large key size and complex homomorphic operation. In contrast, single-key homomorphic encryption has the advantages of smaller key size and simple homomorphic operation. Therefore, this paper proposes a multi-key homomorphic encryption scheme based on GSW and recurrent neural networks, which combines the advantages of single-key homomorphic encryption and multi-key homomorphic encryption, and verifies its security and effectiveness through theoretical analysis.

2 Related Works

2.1 Homomorphic Encryption

Homomorphic encryption technology can be divided into semi-homomorphic encryption, partial homomorphic encryption and full homomorphic encryption according to the types and times supported by ciphertext data operations. Among them, partially homomorphic encryption (PHE) [24] only supports addition or multiplication for an infinite number of homomorphic operations. Some-what-homomorphic encryption (SWHE) [13] supports both addition and multiplication homomorphisms, but the number of homomorphisms is limited. Fully homomorphic encryption (FHE) [11] supports an infinite number of addition and multiplication homomorphisms. The first two homomorphic schemes are limited in practical application, but full homomorphic encryption supports a complete range of ciphertext operations, such as data mining, ciphertext retrieval, outsourcing computing and other intensive operations, which can calculate ciphertext data based on full homomorphic encryption algorithms. Therefore, full homomorphic encryption has a wide range of application prospects.

A complete homomorphic encryption scheme consists of four algorithms: $KeyGen$, $Encrypt$, $Decrypt$ and $Eval$. The input parameters of the key generation algorithm $KeyGen$ are security parameters, and the output is a pair of public and private keys (pk, sk). pk is the public key, and sk is the private key. Security parameters are related to the length of the key, and a large enough security parameter can ensure the security of the algorithm. The input parameters of $Encrypt$ are the plaintext message m and the encrypted public key pk, and the output is the ciphertext c. The input parameters of $Decrypt$ are ciphertext c and the private key sk, and the output is the plaintext m obtained after decrypting ciphertext c. The input parameters of the ciphertext algorithm $Eval$ include three parts: public key pk, circuit and ciphertext list. Where a circuit is a formal description of any computation algorithm, and all ciphertexts in the ciphertext list are encrypted using public key pk. Ciphertext arithmetic is the key to realize the homomorphism property. The result of $Eval$ must be decrypted correctly for the encryption homomorphism to hold.

2.2 Ciphertext Extension

In the case of single key, homomorphic multiplication of learning with errors over rings (RLWE)-based ciphertext involves two steps: vector multiplication and relinearization. For the input ciphertext ct_1 and ct_2, their vector product is calculated first to obtain an extended ciphertext satisfying $< ct, sk \otimes sk >=< ct_1, sk > \cdot < ct_2, sk >$. Since $sk \cdot sk$ contains the non-linear part s^2, a re-linearization procedure that is the core operation of

homomorphic operations needs to be performed to convert the extended ciphertext into a typical ciphertext that encrypts the same plaintext information. This process requires a re-linearized key (compute key), which is obtained by encrypting $s2$ under sk, so the re-linearization process can be understood as a key switching process [27].

In the case of multiple keys, the extended ciphertext related to k different users (different private keys) is $\bar{ct} = (c_0, c_1, \cdots, c_k) \in R_q^{k+1}$, which can be decrypted by the extended private key $\bar{sk} = (1, s_1, s_2, \cdots, s_k)$ in series, and the plaintext $\mu =< \bar{ct}, \bar{sk} >= c_0 + \sum_{i=0}^{k} c_i \cdot s_i$ can be obtained. As in the case of single key, the vector product is first performed and the extended ciphertext corresponding to $\bar{sk} \otimes \bar{sk}$ is returned. Since there is also a nonlinear part $s_i \cdot s_j$ in $\bar{sk} \otimes \bar{sk}$, a re-linearized key (compute key) needs to be generated as well. The key consists of multiple $s_i \cdot s_j$ ciphertexts. Obviously, the element $s_i \cdot s_j$ that $\bar{sk} \otimes \bar{sk}$ contains depends on the private keys of two different users. Therefore, the relinearized key corresponding to the nonlinear element cannot be generated by one way, which is different from the traditional homomorphic operation in the case of single key.

2.3 Multi-party Computing

In the traditional computing model, such as the Turing machine model in single machine state, the input, output and operation program are all owned by one party alone. Multi-party computing refers to the situation where multiple participants provide data or computing resources and perform joint calculations, which raises issues such as fairness, data privacy, and computing costs compared to unilateral computing. Common concepts such as secure multi-party computing, outsourced computing, distributed computing, and centralized computing where data is provided by multiple parties belong to the category of multi-party computing. Among them, the basic idea of secure multi-party computation (MPC) [23] refers to a system where multiple participants can safely compute a convention function without trusted third parties. Safe computation means that each participant cannot obtain input and output information from other participants during the function computation.

There are many factors that affect the privacy information leakage of secure multi-party computing schemes. It is challenging to design a secure and efficient implementation scheme of secure multi-party computing, which prompts people to seek a balance between the availability and privacy of schemes. At present, the research of secure multi-party computing mostly focuses on how to prevent the disclosure of privacy information in the calculation process, that is, how to prevent one participant from obtaining the input or output information of other participants through the calculation process. However, the possibility that one actor can deduce the input or output of other actors from their legitimate function inputs and outputs has not been sufficiently studied.

3 BGV Single Key Homomorphic Encryption Scheme

BGV scheme is a finite series homomorphic encryption scheme, and adopts analog-digital exchange, key exchange and other ways to control noise, so each layer has its own public key and private key. There are two construction methods of BGV scheme, which are based on LWE problem and RLWE problem. This paper only describes the construction method based on RLWE problem.

The BGV homomorphic encryption scheme consists of four parts: key generation (KeyGen), encryption (Enc), decryption (Dec) and homomorphic operation (Eval). The specific steps for each part are as follows:

1) Key generation (KeyGen): Input circuit depth L, security parameter λ. Select the noise distribution $\chi = \chi(L, \lambda)$, which is a bounded distribution over R. Select L decreasing modules $q_L > q_{L-1} > \cdots > q_0$. Choose an integer p with all q_l reciprocity, for each $l = 0, 1, 2, \cdots, L$, perform the following calculation:

   - Sample $s_l \leftarrow \chi, s_l \in R_q$ from the error distribution and obtain the private key of this layer as sk_l.
   - Sample an uniformly $a_l \leftarrow R_q$ from R_q, sample $e \leftarrow \chi$ from error distribution, and calculate the public key corresponding to the private key sk, as shown in Equation (1):

   $$pk_l = (b_l, a_l) = (-a_l \cdot s_l + e_l mod q_l, a_l). \quad (1)$$

   - When l is not 0, it is also necessary to calculate key exchange parameters, as shown in Equation (2):

   $$\tau_{sk'_l \to sk_{l-1}} = SwitchKeyGen(sk'_l, sk_{l-1}). \quad (2)$$

   Where $sk'_l = sk_l \otimes sk_l \in R_q^4$ is the tensor product of the private key and the private key.

   After key generation, it gets public key pk_l, private key sk_l, calculation key $\tau_{s'_l \to s_{l-1}}$.

2) Encryption (Enc): For any plaintext $\mu \in R_p$, which needs to be encrypted, it is necessary to sample $r \leftarrow \chi$, $e \leftarrow \chi$ from the error distribution and output the ciphertext of layer L (initial layer), as shown in Formula (3):

$$c = r \cdot pk_l + (m + e, 0). \quad (3)$$

3) Decryption (Dec): In order to decrypt the $l-th$ layer ciphertext c, it is necessary to calculate according to formula (4):

$$\mu =< c, sk_l > mod q_l mod p. \quad (4)$$

Where $< *, * >$ represents the inner product operation.

4) Homomorphic operation (Eval): Input the calculated function C, ciphertext c_i, calculate the key evk, and output the calculation result. BGV type homomorphic encryption scheme supports homomorphic addition and homomorphic multiplication operations, which are implemented as follows:

- Homomorphic addition: Input two ciphertexts. The ciphertexts must have the same modulus and corresponding private key. If the conditions are not met, use analog-to-digital exchange and key exchange to perform ciphertext operations. Assume that the modulus of the two ciphertexts is q_l. First, it calculates following formula:

$$c_3 = c_1 + c_2 mod q_l \in R_{q_l}^2. \tag{5}$$

Then the ciphertext is zeroized to get $c_3' = (c_3|0) \in R_{q_l}^2$, where $(*|*)$ represents the operation of vector concatenation, and the key corresponding to the ciphertext is $sk_l' = sk_l \otimes sk_l \in R_q^4$, where the multiplication symbol represents the tensor product; Then, the ciphertext key is exchanged to sk_{l-1} through key exchange, as shown in Formula (6):

$$\bar{c}_3 = SwitchKey(\tau'_{s_l \to s_{l-1}}, \bar{c}_3). \tag{6}$$

Finally, the modulus of the ciphertext is reduced to q_{l-1} through analog-digital exchange, and the result is obtained, as shown in Equation (7):

$$c_3 = ModulusSwitch(\bar{c}_3, q_l, q_{l-1}). \tag{7}$$

- Homomorphic multiplication: If the conditions are not met, the analog-digital exchange and key exchange are used to operate the ciphertext so that the conditions are met. First, it calculates Formula (8):

$$\tilde{c}_3 = c_0 \otimes c_1 \in R_{q_l}^4. \tag{8}$$

In this case, the key corresponding to the ciphertext is $sk_l' = sk_l \otimes sk_l$. Then, the key of the ciphertext is exchanged to sk_{l-1} through key exchange, as shown in Formula (9):

$$\bar{c}_3 = SwitchKey(\tau'_{s_l \to s_{l-1}}, \tilde{c}_3). \tag{9}$$

Finally, the modulus of the ciphertext is reduced to q_{l-1} through analog-digital exchange, and the result is obtained, as shown in Equation (10):

$$c_3 = ModulusSwitch(\bar{c}_3, q_l, q_{l-1}). \tag{10}$$

Key exchange and analog-digital exchange are algorithms in BGV homomorphic encryption schemes. Analog-to-digital switching switches the modulus of the ciphertext. After the calculation is complete, the module of the ciphertext is switched to a smaller module, which can reduce the absolute size of the noise in the ciphertext and control the increase of noise. Under the premise of controllable noise scale, this part does not affect the correctness of the final result. The key exchange algorithm can change the private key of ciphertext without decrypting it, and does not change the corresponding plaintext. The key exchange in homomorphic addition is also optional, but after homomorphic multiplication, the key exchange must be used to control the scale of ciphertext growth.

4 Data Series Generation Based on Recurrent Neural Network

First of all, the unstructured economic big data is segmented, and the Time module is obtained after it is segmented into small fragments. Then the Fourier transform is used to convert it to the FFT module in frequency domain, and the data is processed more deeply. It is processed as a three-dimensional tensor to make it adapt to the input dimension of the neural network [1, 2, 28], the number of training samples is a one-dimensional tensor, the length of the big data sequence in the hidden layer is a two-dimensional tensor, and the big data sequence vector is a three-dimensional tensor.

It is very important to determine the parameters, which is related to the quality of learning results and the length of training time. The initial input value is selected in the interval $(-1, 1)$, and the data is normalized to obtain the input and output tensors X and Y. Moving X back one time step, it can get $Y = x_1, \cdots, x_t$.

Char-RNN is a character-level deep recurrent neural network. In this study, a single layer long short-term memory network (LSTM) is used to train recurrent neural networks to learn sequence information in text content, while for unstructured economic big data sequence files, two-layer LSTM and bidirectional LSTM are used for training.

The input tensor $X(N, D, T)$ is taken as the training data, and the number of iterations, batches, the number of hidden layer units and the number of network layers constitute the input of LSTM, and the learned recurrent neural network model is taken as the output of LSTM. The training process is as follows:

- Step 1: Determine whether the weight matrix exists, and then set the weight matrix as the initial parameter of the recurrent neural network model for training;
- Step 2: Solve the hidden layer element error and error gradient;
- Step 3: After iteration, the loss function is solved, and the weight matrix is changed according to a certain law until the end of the training.

Y is the loss function and the $\hat{Y}$ is output of the recurrent neural network. MSE represents the mean square error of Y and $\hat{Y}$. Using the recurrent neural network training data, the recurrent neural network will output Y after each training. By comparing the target output $\hat{Y}$, determine the error value of $\hat{Y}$, and change the weight [12]. After several iterations, the error value is reduced to the minimum, and when the actual output is closest to the target output, the weight parameter is stored to prepare for further generation of big data series.

The economic big data file can be obtained after LSTM training, and the training data is input into the recurrent neural network, and the predicted value of the economic big data series can be obtained after one training of the big data model. The algorithm process of generating big data sequence is as follows:

- Step 1: On the basis of constructing recurrent neural network, input the optimal weight matrix obtained after training into the network;
- Step 2: The seed sequence A in the training data is input into the big data sequence generation algorithm;
- Step 3: Predict the subsequent big data sequence by the given sequence.

5 GSW Scheme

Let κ be the security parameter and L be the number of levels of homomorphic encryption (Leveled-FHE). A brief description of the GSW scheme is given, which is initially defined according to functions $BitDecomp$, $bitdecomp^{-1}$ and $Flatten$, but this paper adopts the simplified approach of Xu *et al.* [26], that is, the definition of tool matrix G.

1) GSW initialization algorithm $GSW.Setup(1^{\kappa}, 1^{L})$.
   - Choose a mode q of bit with parameter $n = n(\kappa, L) \in N$ and error distribution $\chi = \chi(\kappa, L)$ on Z so that the LWE problem is at least 2^{κ} safe against known attacks, choose a parameter $m = m(\kappa, L) = O(nlbq)$.
   - Output parameter $params = (n, q, \chi, m)$, let $\ell = lbq + 1$ and $N = (n+1)\ell$
2) GSW key generation algorithm $GSW.KeyGen(params)$.
   - Evenly select $t = (t_1, t_2, \cdots, t_n)^T \leftarrow Z$ and calculate $s \leftarrow (1, -t^T)^T = (1, -t_1, -t_2, \cdots, -t_n)^T \in Z_q^{(n+1)\times 1}$.
   - Uniformly choose a random common matrix $B \leftarrow Z_q^{m\times n}$ and an error vector $e \leftarrow \chi^m$.
   - Compute the vector $b = Bt + e \in Z_q^m$ and construct the matrix $A = (b|B)$, and satisfy $As = (b|B)s = (Bt + e|B)(1, -t)^T = Bt + e - Bt = e$.
   - Return private key $sk \leftarrow s$ and public key $pk \leftarrow A$.
3) GSW encryption algorithm $C \leftarrow GSW.Enc(pk, \mu)$.
   - Let G be the above $(n+1)\times N$ dimensional tool matrix, and uniformly choose a random matrix $R \leftarrow 0, 1^{m\times N}$.
   - Encrypt single bit message $\mu \in 0, 1$ and generate ciphertext $C = \mu G + A^T R(modq) \in Z_q^{(n+1)\times N}$. It should be noted that in the original GSW scheme, the encryption algorithm uses $Flatten(\mu I + BitDecomp(RA)) \in 0, 1^{N\times N}$. Where I is an identity matrix.
4) GSW decryption algorithm $\mu' \leftarrow GSW.Dec(sk, C)$.
   - Input the private key $sk = s \in Z_q^{n+1}$, so that k satisfies $q/4 < 2^{k-1} \le q/2$, where $C[k]$ is the $k-th$ column of C.
   - Calculate $x \leftarrow < C[k], s > (modq)$ in the range $(-q/2, q/2)$, where $< C[k], s >= C[k]^T s$, and $C^T s = \mu G^T s + R^T As = \mu(2, 4, 6, \cdots)^T + R^T e$. From the above, it can be seen that the $k-th$ column of the ciphertext matrix C selected in the calculation corresponds to the $k-th$ coordinate of the vector $< C[k], s >$, that is, $\mu 2^{k-1} + R_k^T e$.
   - Output $\mu' = |x/2^{k-1}|$. Therefore, if $|x| < 2^{k-2} \le q/4$, return 0; otherwise return 1.
5) GSW operation algorithm $GSW.Eval(pk, (C_1, C_2, \cdots, C_l))$.
   - Addition operation $GSW.Add(C_1, C_2)$. Output $C_1 + C_2 = (\mu_1 + \mu_2)G + A^T(R_1 + R_2)$.
   - Multiplication operation $GSW.Mult(C_1, C_2)$. Output $C_1 G^{-1}(C_2) = (\mu_1 G + A^T R_1)G^{-1}(C_2) = \mu_1 C_2 + A^T R_1 G^{-1}(C_2) = \mu_1\mu_2 G + A^T(R_1 G^{-1}(C_2) + \mu_1 R_2)$.

6 Scheme Analysis

The basic idea of the scheme proposed in this paper is to transform a multi-key homomorphic encryption problem into a single-key homomorphic encryption problem, thereby reducing the size of ciphertext and improving the efficiency of homomorphic encryption. This section will analyze the scheme proposed in this paper from the two aspects of security and computational efficiency respectively, and prove that the scheme proposed in this paper has significantly improved efficiency compared with other multi-key homomorphic encryption schemes for complex computational functions under the premise of ensuring security.

6.1 Security Analysis

In the scheme proposed in this paper, all participants first generate a shared public key and computational key. Each participant uses the public key to encrypt, and uses the generated computational key to complete the homomorphic operation process of single-key homomorphic encryption, and obtains the homomorphic operation result. Finally, all parties decrypt together through the multi-key homomorphic encryption method to obtain the operation result. Each of these procedures will be shown to be safe for honest but curious participants.

For the simplified multi-key homomorphic encryption process, the $d_{i,0}$, $d_{i,1}$ and $d_{i,2}$ required in the calculation process are obtained by encrypting the private key s_i with another key r_i, and r_i is also encrypted by s_i. The problem of solving s_i by $d_{i,0}$, $d_{i,1}$ and $d_{i,2}$ is LWE problem, so the process is safe.

For the decryption process, on the one hand, the decryption process is completed by the multi-key homomorphic encryption scheme; on the other hand, because the decryption process involves the computation party and there is homomorphic multiplication, the partial decryption result of the multi-key homomorphic encryption is independent of the private key, so the process is also secure.

6.2 Efficiency Analysis

The complexity of BGV multikey homomorphic encryption mainly comes from the operations related to homomorphic multiplication. Compared with single key homomorphism multiplication, the calculation of key required for relinearization after multi-key homomorphism multiplication becomes more complicated. In order to illustrate this problem, this paper analyzes the forms of computational keys for homomorphic encryption in the case of single key and multi-key.

For single-key homomorphic encryption, assume that the two ciphertexts encrypted with the private key $sk = (1, s)$ are $c_1 = (b_1, a_{1,1}, a_{1,2})$, $c_2 = (b_2, a_{2,1}, a_{2,2})$. The result of the multiplication of ciphertext decryption is shown $Dec(c_1) \cdot Dec(c_2) = (b_1 + a_1 \cdot s) \cdot (b_2 + a_2 \cdot s) = b_1 b_2 + (b_1 a_2 + b_2 a_1)s + a_1 a_2 s^2$.

For multiple-key homomorphic encryption, assume that the two ciphertexts encrypted with the private key $sk_1 = (1, s_1)$ and $sk_2 = (1, s_2)$ are $c_1 = (b_1, a_{1,1}, a_{1,2})$, $c_2 = (b_2, a_{2,1}, a_{2,2})$. The result of the multiplication of ciphertext decryption is shown $Dec(c_1) \cdot Dec(c_2) = (b_1 + a_{1,1}s_1 + a_{1,2}s_2) \cdot (b_2 + a_{2,1}s_1 + a_{2,2}s_2) = b_1 b_2 + \cdots + a_{1,1}a_{2,1}s_1^2 + \cdots$.

The quadratic terms of s_1 and s_2 in the above formula can be eliminated by using the re-linearization technique. Compared with the single key case where there is only one quadratic term, the number of quadratic terms in multi-key homomorphic encryption is not only large, but also contains quadratic terms shaped like $s_i s_j$. Due to the existence of such special quadratic terms, a single participant cannot complete the calculation of the calculated key. In the existing BGV multi-key homomorphic encryption scheme, in the key generation stage, each participant only calculates the key generation materials, and then the computational party calculates the computational key through these calculation key generation materials. Since the calculation of the key is related to the private elements of the participant, the above operations need to be carried out under the premise of ensuring that the private elements are not leaked. Therefore, calculating the computational key is always an important factor affecting the efficiency of BGV multi-key homomorphic encryption.

In addition to significantly reducing the complexity of the process of calculating the key, the shared key encryption proposed in this paper can also reduce the size of the key and ciphertext. For the existing BGV multi-key homomorphic encryption scheme, the ciphertext extension method is needed to convert the ciphertext from different participants to the case of encryption by the same key, so as to carry out homomorphic operation. This process uses ciphertext concatenation, so that the size of the ciphertext is enlarged. However, the scheme proposed in this paper avoids the above problems because it uses a shared key.

The time and space complexity comparison between the scheme proposed in this paper and other BGV multi-key homomorphic encryption schemes is shown in Table 1.

Table 2 provides a comparison between the proposed scheme and several typical schemes. It can be seen from Table 2 that the three basic features of the proposed scheme are the same as those of Reference [15]. The security of both schemes is based on RIWE assumption, multiple key rings and batch processing.

7 Conclusions

In order to verify the encryption storage performance of unstructured data, unstructured economic data is taken as the research object. We proposed a homomorphic encryption scheme for economic data based on recurrent neural network and Gentry-Sahai-Waters (GSW). First, in the data series acquisition experiment, the mean square error of different iterations is compared and analyzed. The results show that with the increase of iterations, the difference between the actual big data series and the target data becomes smaller. Secondly, experiments on big data encryption and anti-attack show that the proposed method can conceal the real information in big data, and the encryption effect is significant, and the big data after encryption has a good resistance to malicious attacks, proving that it has a good anti-attack and higher storage efficiency.

Acknowledgments

This study was supported by the Fund Project: Phased achievement of Henan Provincial Philosophy and Social

Table 1: Analysis of homomorphic encryption efficiency

Schemes	Calculation key size	Ciphertext size	Calculation key generation time	Homomorphic multiplication time
Reference [22]	$O(k^3n)$	$O(kn)$	$O(k^3n)$	$O(k^3n)$
Reference [21]	$O(k^2n^2)$	$O(kn)$	$O(k^2n^2)$	$O(k^2n^2)$
Reference [15]	$O(kn)$	$O(kn)$	$O(kn)$	$O(k^2n)$
Proposed	$O(n)$	$O(n)$	$O(k^2n)$	$O(n)$

Table 2: Comparison of main features with different schemes

Scheme	Hypothesis	Key ring	Batch Processing
Reference [22]	LWE	Single	nonsupport
Reference [21]	LWE/RLWE	multiple	nonsupport
Reference [15]	RLWE	multiple	support
Proposed	RLWE	multiple	support

Science Planning Project "Research on the Realization Path of E-commerce to Improve the Business Performance of small farmers in Henan Province", project number: 2022BJJ106. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. K. Abiodun, A. E. Adeniyi, A. O. Victor, J. B. Awotunde, O. G. Atanda and J. K. Adeniyi, "Detection and prevention of data leakage in transit using LSTM recurrent neural network with encryption algorithm," in *2023 International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG), Omu-Aran, Nigeria*, pp. 01-09, 2023. doi: 10.1109/SEB-SDG57117.2023.10124503.

[2] N. Abughazalah, A. Latif, M. Hafiz, M. Khan, A. S. Alanazi, I. Hussain, "Construction of multivalued cryptographic boolean function using recurrent neural network and its application in image encryption scheme," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5403-5443, 2023.

[3] A. Acar, H. Aksu, A. Uluagac, M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1-35, 2018.

[4] Z. Brakerski, R. Perlman, "Lattice-based fully dynamic multi-key FHE with short ciphertexts," in *Annual international cryptology conference. Berlin, Heidelberg: Springer Berlin Heidelberg*, pp. 190-213, 2016.

[5] X. Che, T. Zhou, N. Li, H. Zhou, Z. Chen and X. Yang, "Modified multi-key fully homomorphic encryption based on NTRU cryptosystem without key-switching," *Tsinghua Science and Technology*, vol. 25, no. 5, pp. 564-578, 2020.

[6] H. Chen, W. Dai, M. Kim, Y. Song, "Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 395-412, 2019.

[7] L. Chen, Z. Zhang, X. Wang, "Batched multi-hop multi-key FHE from ring-LWE with compact ciphertext extension," in *Theory of Cryptography: 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II 15. Springer International Publishing*, pp. 597-627, 2017.

[8] W. Chongchitmate, R. Ostrovsky, "Circuit-private multi-key FHE," in *IACR International Workshop on Public Key Cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg*, pp. 241-270, 2017.

[9] M. Clear, C. McGoldrick, "Multi-identity and multi-key leveled FHE from learning with errors," in *Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II 35. Springer Berlin Heidelberg*, pp. 630-656, 2015.

[10] Y. Dor, Y. Hu, B. Sunar, "Homomorphic AES evaluation using the modified LTV scheme," *Designs, codes and crytography*, vol. 80, no. 2, pp. 333-358, 2016.

[11] N. M. Hijazi, M. Aloqaily, M. Guizani, B. Ouni and F. Karray, "Secure federated learning with fully homomorphic encryption for IoT communications," *IEEE Internet of Things Journal*, 2023. doi: 10.1109/JIOT.2023.3302065.

[12] T.-L. Huoh, Y. Luo, P. Li and T. Zhang, "Flow-based encrypted network traffic classification with graph neural networks," *IEEE Transactions on Network*

and Service Management, vol. 20, no. 2, pp. 1224-1237, 2023.

[13] H. B. Kwon, S. Kosieradzki, J. Blevins and J. Ueda, "Encrypted coordinate transformation via parallelized somewhat homomorphic encryption for robotic teleoperation," *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Seattle, WA, USA*, pp. 228-233, 2023. doi: 10.1109/AIM46323.2023.10196122.

[14] A. López-Alt, E. Tromer, V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Symposium on the Theory of Computing, ACM*, 2012. DOI:10.1145/2213977.2214086.

[15] C. Li, L. Yang, S. Yu, W. Qin, J. Ma, "SEMMI: Multi-party security decision-making scheme for linear functions in the internet of medical things," *Information Sciences*, vol. 612, pp. 151-167, 2022.

[16] N. Li, T. Zhou, X. Yang, Y. Han, W. Liu and G. Tu, "Efficient multi-key FHE with short extended ciphertexts and directed decryption protocol," *IEEE Access*, vol. 7, pp. 56724-56732, 2019.

[17] K. Lin, M.-S. Hwang, "Research on data security and privacy protection of smart grid," *International Journal of Network Security*, vol. 25, no. 6, 2023.

[18] P. Mukherjee, D. Wichs, "Two round multiparty computation via multi-key FHE," in *Advances in Cryptology-EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II 35. Springer Berlin Heidelberg*, pp. 735-763, 2016.

[19] K. Munjal, R. Bhatia, "A systematic review of homomorphic encryption and its contributions in healthcare industry," *Complex & Intelligent Systems*, vol. 9, no. 4, pp. 3759-3786, 2023.

[20] C. Peikert, S. Shiehian, "Multi-key FHE from LWE, revisited," in *Theory of cryptography conference. Berlin, Heidelberg: Springer Berlin Heidelberg*, pp. 217-238, 2016.

[21] R. Sendhil, A. Amuthan, "Verifiable quaternion fully homomorphic encryption scheme for mitigating false data injection attacks by privacy preservation in fog environment," *Journal of Information Security and Applications*, vol. 71, 2022.

[22] R. K. Sheu, Y. C. Lin, M. S. Pardeshi, C. Y. Huang, K. C. Pai, L. Chen, C. Huang, "Adaptive autonomous protocol for secured remote healthcare using fully homomorphic encryption (AutoPro-RHC)," *Sensors*, vol. 23, no. 20, pp. 8504, 2023.

[23] V. Sucasas, A. Aly, G. Mantas, J. Rodriguez and N. Aaraj, "Secure multi-party computation-based privacy-preserving authentication for smart cities," *IEEE Transactions on Cloud Computing*, vol. 11, no. 4, pp. 3555-3572, Oct.-Dec. 2023, doi: 10.1109/TCC.2023.3294621.

[24] B. Wang, H. Li, Y. Guo, J. Wang, "PPFLHE: A privacy-preserving federated learning scheme with homomorphic encryption for healthcare data," *Applied Soft Computing*, vol. 146, 2023.

[25] X. Wang, S. Yin, H. Li, L. Teng, S. Karim, "A modified homomorphic encryption method for multiple keywords retrieval," *International Journal of Network Security*, vol. 22, no. 6, pp. 905-910, 2020.

[26] W. Xu, B. Wang, Q. Qu, T. Zhou, P. Duan, "Modified multi-key fully homomorphic encryption scheme in the plain model," *The Computer Journal*, vol. 66, no. 10, pp. 2355-2364, 2023.

[27] S. Yin, H. Li, A. A. Laghari, T. R. Gadekallu, G. A. Sampedro and A. Almadhor, "An anomaly detection model based on deep auto-encoder and capsule graph convolution via sparrow search algorithm in 6G internet-of-everything," *IEEE Internet of Things Journal*, 2024. doi: 10.1109/JIOT.2024.3353337.

[28] S. Yin, J. Liu, L. Teng, "A sequential cipher algorithm based on feedback discrete hopfield neural network and logistic chaotic sequence," *International Journal of Network Security*, vol. 22, no. 5, pp. 869-873, 2020.

[29] T. Zhou, L. Chen, X. Che, W. Liu, Z. Zhang, X. Yang, "Multi-key fully homomorphic encryption scheme with compact ciphertexts," *Cryptology ePrint Archive*, 2021. https://eprint.iacr.org/2021/1131.

Biography

Yueyue Dong biography. Yueyue Dong, female, Xinyang, Henan, Master's degree, Associate Professor, is with the School of Business Administration, Zhengzhou University of Science and Technology. Research direction: Rural Economic Development and Management, Data analysis.

A Multi-layer Data Encryption Method Based on Reverse Artificial Swarm Algorithm and Packet Convolutional Chaotic Sequence

Yuankun Du[1], Fengping Liu[2], and Fei Wang[3]
(Corresponding author: Fei Wang)
School of Big Data and Artificial Intelligence, Zhengzhou University of Science and Technology[1]
School of Information Engineering, Zhengzhou University of Science and Technology[2]
School of Information Engineering, Henan Institute of Animal Husbandry Economics[3]
Zhengzhou 450064, China
Email: duyuankk@163.com
(Received Nov. 20, 2023; Revised and Accepted Feb. 7, 2024; First Online Apr. 25, 2024)
The Special Issue on Data Fusion, Deep Learning and Optimization Algorithms for Data Privacy Protection
Special Editor: Prof. Shoulin Yin (Harbin Institute of Technology)

Abstract

In order to improve the security of data and reduce the risk of data being modified and stolen during transmission and storage, a multi-layer data encryption method based on reverse artificial swarm algorithm and packet convolutional chaotic sequence is proposed. Firstly, residual and grouped convolution blocks extract and utilize image features. Secondly, in order to avoid falling into the local optimal, the reverse learning strategy is introduced in the three stages of population initialization, hiring bees, and observing bees, respectively. The reverse order strategy between two points and the element exchange strategy is adopted to accelerate the optimization speed. According to the chaotic Frank sequence, the data is encrypted by multiple layers. The experimental results show that the proposed method has a high scrambling performance. The time required for ciphertext generation is less than 100ms, the encryption efficiency is high, and the data security is improved.

Keywords: Multi-Layer Data Encryption; Packet Convolutional Chaotic Sequence; Residual Convolution Block; Reverse Artificial Swarm Algorithm

1 Introduction

In the context of the continuous progress of network technology and computer technology, People gradually begin to use computers for office and communication, and computers can realize information communication and sharing among users [12, 15, 18]. Mobile office has gradually become the mainstream trend, and the office data in computers is increasing day by day, which leads to some problems and security risks. For example, some illegal elements sneak attack and steal the mobile office data transmitted in the network channel by some means, causing serious problems The loss. In an environment where data security cannot be guaranteed, it is of great significance to study mobile office data encryption methods [13].

Bhushan *et al.* [3] proposed a cloud-assisted ciphertext policy attribute base data sharing encryption method on blockchain. For the data to be encrypted, the symmetric key was encrypted by attribute encryption technology, and the symmetric key was stored in the cloud server. The key ciphertext was embedded in the blockchain to realize data encryption. The ciphertext generated by this method had low scrambling, and the security of the encrypted data was poor. Zhang *et al.* [19] proposed a social network privacy data protection method based on blockchain, which classified and processed the data that needed to be encrypted, established Hash function to anonymize the classified data, and encrypted the data through asymmetric encryption algorithm. The blockchain was simulated by python to achieve data encryption protection. This method took a long time to generate ciphertext, thus prolonged the data encryption time, and had the problem of low encryption efficiency.

In order to solve the problems in the above methods, a multi-layer data encryption method based on reverse artificial swarm algorithm and packet convolutional chaotic sequence is proposed. This method uses chaotic sequence to encrypt data double-layer, so as to improve the security of data and enhance the effect of data encryption.

2 Proposed Method

2.1 Data Preprocessing

In order to improve the security of data, it is necessary to de-noise it. By combining the independent component analysis method [5, 9] and the empirical mode decomposition method, the noisy data in the data is eliminated.

The M-dimensional data is represented by $X = [x_1, x_2, \cdots, x_m]^T$, which is obtained by linear aliasing of n independent components $d = [d_1, d_2, \cdots, d_n]^T$.

$$x_i = \sum_{j=1}^{n} s_{ij} d_j. \tag{1}$$

Where s_{ij} represents the mixing coefficient, the matrix $X = Sd$ is used to describe the above equation. Where S stands for mixed matrix. The observation signal model will be interfered by noise in the daily application process, and the data plus noise model is established:

$$X = Sd + o. \tag{2}$$

Where, the vector o is composed of noise. On the basis of the added noise model, data x_0 with noise is described by the following formula.

$$x_0 = s_0 d + \sum_{i=1}^{m} s_i o_i. \tag{3}$$

Where s_i represents the weight corresponding to noise o_i. s_0 describes the weights corresponding to useful data d. In order to process noisy data, it is necessary to transform one-dimensional data into multidimensional data through virtual observation channel. The standard independent component model X is established by using $O = [o_1, o_2, \cdots, o_m]^T$ to represent the virtual noise present in the data:

$$X = \begin{bmatrix} s_0 & s_1 & s_2 & \cdots & s_m \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} d \\ o_1 \\ o_2 \\ \vdots \\ o_m \end{bmatrix} \tag{4}$$

By solving the inverse matrix corresponding to the matrix S and combining the independent component analysis method, the separation matrix is obtained, and then the optimal estimate $Y = S^{-1}X = \bar{D}$ of the original data is obtained. The useful data of the data is extracted from the output signal $\bar{D}$, and the denoising of the data is realized.

The analysis of the above process shows that establishing virtual channel is the key to data denoising. Empirical mode decomposition method [4, 8] is adopted to decompose data X:

$$X = \sum_{i=1}^{m} IMF_i + r_i. \tag{5}$$

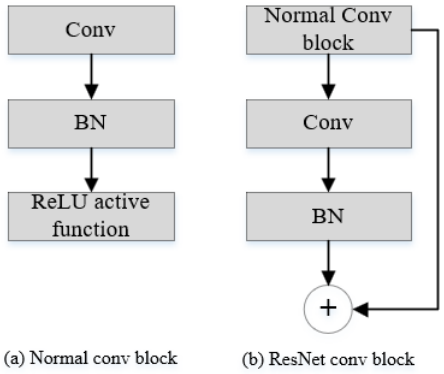


Figure 1: Two convolution block structures in this paper

Where r_i represents the remainder term. IMF_i describes the decomposed intrinsic modal components. The time spectrum corresponding to IMF_i obtained from the decomposition of the above formula is calculated, and the mutual relation number $T(d, IMF_i)$ is set to analyze the relationship between useful data d and the inherent modal component IMF_i.

$$T(d, IMF_i) = \frac{cov(d, IMF_i)}{\sqrt{cov(d,d)cov(IMF_i, IMF_i)}}. \tag{6}$$

In the formula, $cov(a, b)$ describes the covariance of a and b, and the noise o is separated from the data according to the cross-relation number $T(d, IMF_i)$:

$$o = \sum_{i=0}^{n} IMF_i. \tag{7}$$

2.2 Grouped Convolution Block

The middle part of the network consists of four residual convolution blocks, its structure is proposed in reference [2], as shown in Figure 1. Its structure is basically the same as the basic convolutional block, including the convolutional layer, batch normalization (BN) layer, and nonlinear activation function, the main difference is that a short connection is added between the input and output, allowing the low-level and high-level feature maps to be added, forcing the network to constantly fit the residual mapping. Using this structure can solve the problem that the accuracy reaches saturation and then deteriorates rapidly as the depth of the network increases, and in the process of backpropagation of network training, such residual structure avoids the problem that the weight becomes smaller and smaller due to a large number of derivation and compounding operations, resulting in the disappearance of the gradient.

Group convolution was first proposed in AlexNet and had been improved in ResNeXt [7] networks. Compared with common convolution, it can reduce the network parameters, greatly reduce the computation with the same accuracy, and it is not easy to overfit. So the network

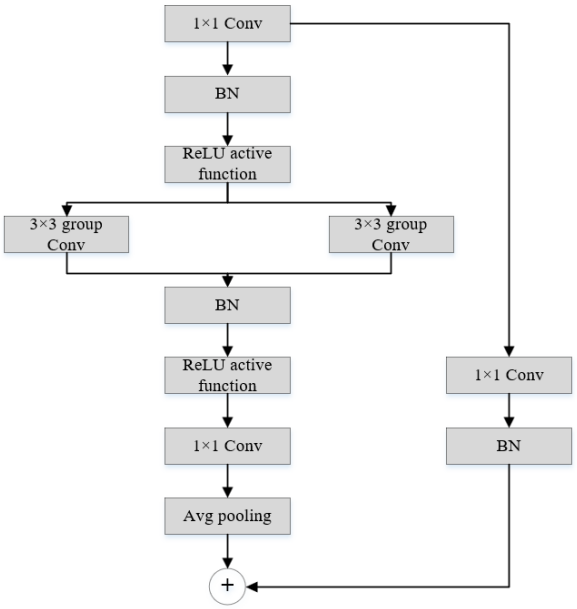


Figure 2: Grouping convolution block structure

uses two block convolution blocks after the residual convolution block to reduce the parameters of the network and optimize the performance of the network.

The specific structure of the grouped convolution block adopted in this paper is shown in Figure 2. First, 1×1 convolution is used to reduce the dimension of the input feature graph on the channel, followed by 3×3 group convolution, and then 1×1 convolution is used to restore the channel, and finally, the convolution result of the group convolution block is output through the cross-layer short connection structure. Small 1×1 and 3×3 convolution kernels are used in this paper because more small convolution kernels work better than fewer large ones, and with fewer parameters.

2.3 Reverse Artificial Colony Optimization

Artificial bee colony algorithm is an intelligent optimization algorithm [6] based on the foraging process of bees. Aiming at the optimization problem of data encryption network, this paper introduces the reverse learning strategy based on the artificial bee colony algorithm. The calculation formula of the reverse learning strategy is as follows:

$$R = A + B - r. \tag{8}$$

Where A and B are the maximum and minimum values of the elements in the individual respectively. r is the element before the calculation. R is the calculated element. As shown in Figure 3, the individual before calculation is [7,5,6,2,1,3,4,8], where the maximum value is 8 and the minimum value is 1. Formula (8) is used to calculate each element, and the calculated individual is [2,4,3,7,8,6,5,1].

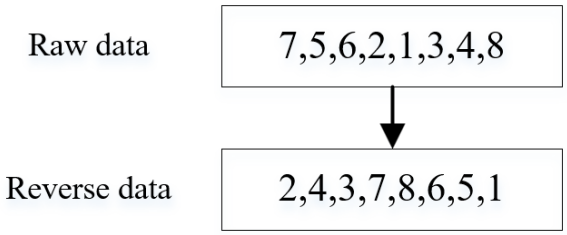


Figure 3: Reverse learning diagram

The introduction of reverse learning strategy expands the search scope of initial population, hired bees and observation bees, and avoids the local optimization in the iterative process to the greatest extent.

The steps of the reverse artificial colony algorithm are described as follows:

1) In the population initialization stage, the initial candidate solution required by the simulation experiment is randomly generated, the reverse solution of the candidate solution is calculated, and the initial population is determined according to the fitness value.

2) The hiring bee phase. In this stage, the candidate solution is obtained by updating the initial population, the reverse solution of the candidate solution is calculated, and the rowable solution and the optimal solution are determined according to the fitness value.

3) Observing bee phase. In this stage, the candidate solution is obtained by updating the feasible solution of the hired bee stage, and the reverse solution of the candidate solution is calculated. The feasible solution and the optimal solution are determined according to the fitness value.

4) Scout bee phase. In this stage, the solution that has not been updated in the iteration process is selected according to the selection probability, and the optimal solution of this stage and the current algebraic optimal solution are retained according to the fitness value. The selection probability expression is as follows:

$$l = \frac{F(i)}{\sum_{i=1}^{N_p} F(i)}. \tag{9}$$

Where $F(i)$ is the fitness value, that is, the objective function value of this paper. N_p is population size.

5) Determine whether the termination conditions are met. If not, return to step 2; If yes, output the global optimal solution.

2.3.1 Encoding

In this paper, the integer encoding method based on workpieces is adopted, assuming that the individual is [8,5,3,6,7,2,1,4], taking the above individual as an example, the length of the individual represents the number of workpieces to be processed, and the number of workpieces to be processed is 8. Each number represents a workpiece number, and the workpiece numbered "8" is the first workpiece to be machined. And so on, the order of the individuals represents the processing order of the first process.

2.3.2 Assignment Strategy

After the processing sequence of the first process is determined, the processing sequence of the subsequent processes is assigned by the longest processing time (LPT) rule [1]. When the station of the parallel machine is idle and multiple workpieces are queued in the temporary storage area, the workpiece with the longest total processing time is placed on the idle machine for processing. When the parallel machine station is free and there is only one work piece in the staging area, the work piece is allocated according to the shortest processing time on the parallel machine.

2.3.3 Neighborhood Search Strategy

The original artificial bee colony algorithm is used to solve continuous problems. Therefore, the following two neighborhood search strategies are adopted in this paper to improve the update mechanism of the artificial bee colony algorithm, so that the proposed algorithm meets the discretization characteristics of the hybrid flow shop scheduling problem.

A. Reverse order strategy between two points

The reverse order strategy between two points is to generate a new neighborhood solution by randomly selecting two elements in an individual and arranging the elements in reverse order between the two selected elements. For example, if the selected elements are 8 and 4 and the sum of the number of elements between them is greater than 3, arrange the elements between 8 and 4 in reverse order; If the selected elements appear to be adjacent during element selection, the positions of the two randomly selected elements are swapped. The selected elements are 6 and 7, swap the positions of 6 and 7; If the sum of the selected elements and the number of elements between them is equal to 3, the three elements are arranged in reverse order. The selected elements are 3 and 7, the middle element is 6, and the three elements are arranged in reverse order.

B. Element exchange strategy

The element exchange strategy is to generate a new neighborhood solution by exchanging two randomly selected element positions, as shown in Figure 4. If the selected elements are 3 and 6, swap the positions of 3 and 6.

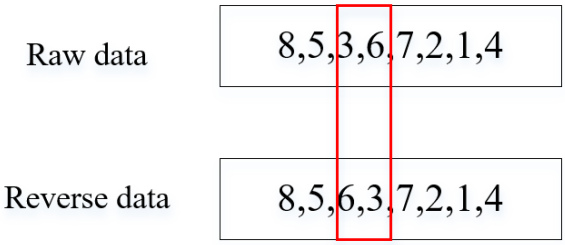


Figure 4: Element exchange strategy diagram

2.3.4 Population Initialization

The quality of the initial population greatly affects the optimization speed, so the reverse learning strategy is introduced in this paper to improve the quality of the initial population. N_p individuals are randomly generated as the initial candidate solution, and the reverse solution of the initial candidate solution is calculated according to Equation (8). In order to ensure the consistency of population size, individuals with the top N_p fitness values are retained as the final initial population according to the elite optimization strategy.

2.3.5 Bee Hiring Phase

In this stage, the candidate solution is generated by using the reverse order strategy between two points on the basis of the initial population, and the feasible solution is retained according to the elite optimization strategy. The specific steps are as follows:

- The reverse order strategy between two points was used to update the initial population and generate candidate solutions.
- The reverse solution of the candidate solution is calculated according to Equation (8).
- In order to ensure the consistency of population size, according to the fitness values of the candidate solution and its reverse solution, the individuals with the top N_p fitness values are retained as the feasible solutions at this stage, and the optimal solution E_i^l is retained.

2.3.6 Bee Observation Stage

In this stage, the feasible solutions of the hiring bee stage are updated by the element exchange strategy to generate candidate solutions, and the feasible solutions of this stage are retained according to the elite optimization strategy. The specific steps are as follows:

- The element exchange strategy is used to update the feasible solutions in the hiring bee stage to generate candidate solutions.

Algorithm 1 Pseudocode of hired bee phase algorithm

for $i = 1$ to N_p
j:=z-uniform(1,1,random13)
k:=z-uniform(1,1,random13)
if $j = k$ **then**
  j:=z-uniform(1,1,random13)
end if
start-point:=j
end-point:=k
if end-point,start-point¿3 **then**
  Reverse order the elements between the selected elements
end if
if end-point,start-point=2 **then**
  Exchange the selected element
end if
if end-point,start-point=3 **then**
  Sort the selected elements in reverse order
end if
The reverse solution of the candidate solution is calculated according to Equation (8).
The individual with N_p before the fitness value retained entered the observation bee stage as the feasible solution, and retained the optimal solution E_i^l.

- The reverse solution of the candidate solution is calculated according to Equation (8).
- In order to ensure the consistency of population size, according to the fitness values of the candidate solution and its reverse solution, the individuals with the top N_p fitness values are retained as the feasible solutions in this stage, and the optimal solution E_i^l in this stage is retained.

Algorithm 2 Pseudocode of observation bee phase algorithm

for $i = 1$ to N_p
j:=z-uniform(1,1,random13)
k:=z-uniform(1,1,random13)
if $j = k$ **then**
  j:=z-uniform(1,1,random13)
end if Exchange the selected element
The reverse solution of the candidate solution is calculated according to Equation (8).
The individual with N_p before the fitness value retained entered the scout bee stage as the feasible solution, and retained the optimal solution E_i^l.

2.3.7 Scout Bee Stage

In this stage, the solution that has not been updated in the iteration process is updated using the element exchange strategy, and the optimal solution in this stage is retained according to the elite optimization strategy [11, 16, 17]. The specific steps are as follows:

- According to the selection probability calculated by Equation (9), the solution that has not been updated for many times is selected and the element exchange strategy is used to generate a new feasible solution.
- The optimal solution E_i^3 in this stage is retained according to the fitness value.
- The current algebraic optimal solution E_i^4 is retained according to the fitness value.
- The global optimal solution E_{best} is output when the iteration termination condition is met.

Algorithm 3 Pseudocode of scout bee phase algorithm

for $i = 1$ to N_p
The selection probability is calculated according to Equation (9).
The solution that has not been updated many times is selected according to the selection probability.
j:=z-uniform(1,1,random13)
k:=z-uniform(1,1,random13)
if $j = k$ **then**
  j:=z-uniform(1,1,random13)
end if
Exchange the selected element.
The optimal solution E_i^3 in this stage is retained according to the fitness value.
The current algebraic optimal solution E_i^4 is retained according to the fitness value.
if Satisfy termination condition **then**
  Output the global optimal solution E_{best}.
end if
if otherwise **then**
  Return to the hired bee phase to the next generation
end if
End.

3 Experiment and Analysis

In order to verify the overall effectiveness of the multi-layer data encryption method based on chaotic sequence, it is necessary to test it.

Multi-layer data encryption method based on chaotic sequence, cloud-assisted ciphertext policy attribute base data sharing encryption method based on blockchain (reference [14]) and social network privacy data protection method based on blockchain (reference [10]) are used to carry out encryption tests on data. The ciphertext bits after encryption by the three methods are shown in Figures 5 $\sim$ 7.

It can be seen from Figures 5 $\sim$ 7 that after the proposed method is used to encrypt mobile office data, there is no rule in the change of ciphertext bits, while there is a certain rule in the ciphertext bits generated by the

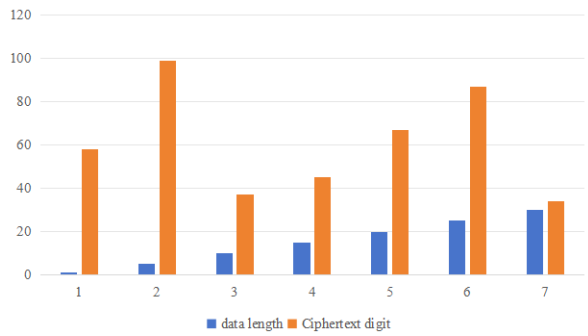


Figure 5: Ciphertext digit with the proposed scheme

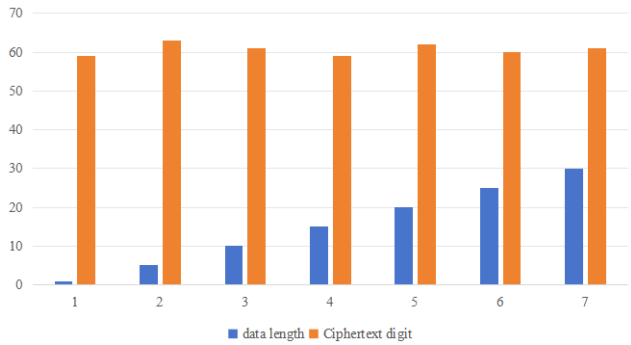


Figure 6: Ciphertext digit with the reference [14]

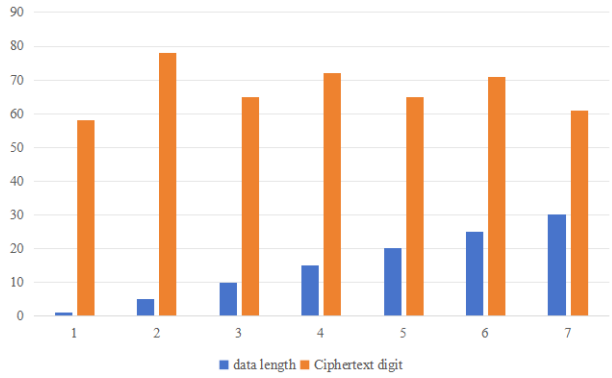


Figure 7: Ciphertext digit with the reference [10]

reference [14] method and the reference [10] method. By comparing the test results of the proposed method, the reference [14] method and the reference [10] method, it can be seen that the ciphertext generated by the proposed method has a high scrambling property, indicating that the proposed method has a good encryption effect.

On the basis of the above tests, the time required to generate ciphertext by the proposed method, the reference [14] method and the reference [10] method is analyzed, and the test results are shown in Table 1.

According to the data in Table 1, the proposed method takes less than 130ms to generate the ciphertext, while the other two methods take longer than 300ms to generate the ciphertext. The proposed method is much lower than the time required for ciphertext generation by the reference [14] method and the reference [10] method, because the proposed method combines the independent component analysis method and empirical mode decomposition method to de-noise data, avoid the interference of noise in the encryption process, shorten the ciphertext generation time and improve the encryption efficiency of the proposed method.

4 Conclusions

In order to improve the security of data and avoid the occurrence of data forgery and loss, it is necessary to encrypt the data. To this end, a multi-layer data encryption method based on reverse artificial swarm algorithm and packet convolutional chaotic is proposed that this method deploys data denoising processing and uses chaotic sequence to deploys multi-layer encryption of data, which improves data security and shortens data encryption time, and verifies that the proposed method has good performance in the field of data encryption and ensures the security of data transmission and storage.

Acknowledgments

This work was supported by: Project type: Henan Province science and technology research project. Project number: 232102210082. Project name: Application research of pig state and behavior recognition and disease warning based on spatiotemporal multi-objective. 2021 Science and Technology Research Project of Henan Provincial Science and Technology Department "Real-time cattle status Recognition and disease early Warning Application Research based on Convolutional representation Flow" (Project number: 212102210138). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] A. Abreu, H. Fuchigami, "An efficiency and robustness analysis of warm-start mathematical models for

Table 1: Ciphertext generation time of different methods/ms

Iteration number	Proposed	reference [14]	reference [10]
10	114	322	355
20	122	327	362
30	109	328	350
40	108	333	356
50	116	329	358
60	117	320	347
70	115	337	364
80	123	326	369
90	112	318	358
100	109	325	353

idle and waiting times optimization in the flow shop," *Computers & Industrial Engineering*, vol. 166, 2022.

[2] M. Ayazoğlu, "IMDeception: Grouped information distilling super-resolution network," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA*, pp. 755-764, 2022, doi: 10.1109/CVPRW56347.2022.00091.

[3] B. Bhushan, G. Sahoo, "Recent advances in attacks, technical challenges, vulnerabilities and their countermeasures in wireless sensor networks," *Wireless Personal Communications*, vol. 98, pp. 2037-2077, 2018.

[4] C. Deng, Y. Huang, N. Hasan, Y. Bao, "Multi-step-ahead stock price index forecasting using long short-term memory model with multivariate empirical mode decomposition," *Information Sciences*, vol. 607, pp. 297-321, 2022.

[5] A. Gupta, A. G. Ravelo-Garcia and F. M. Dias, "A motion and illumination resistant non-contact method using undercomplete independent component analysis and levenberg-marquardt algorithm," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 10, pp. 4837-4848, 2022.

[6] E. Kaya, B. Gorkemli, B. Akay, D. Karaboga, "A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 115, 2022.

[7] A. Kumari, S. Rao, P. Reddy, "Design of hybrid dental caries segmentation and caries detection with meta-heuristic-based ResneXt-RNN," *Biomedical Signal Processing and Control*, vol. 78, 2022.

[8] M. Li, D. Xu, J. Geng, W. Hong, "A ship motion forecasting approach based on empirical mode decomposition method hybrid deep learning network and quantum butterfly optimization algorithm," *Nonlinear dynamics*, vol. 107, no. 3, pp. 2447-2467, 2022.

[9] W. -S. Li, C. -Z. Peng, F. -P. Lai, P. -S. Lai and Y. -S. Chen, "Independent component analysis for the multitag detection of frequency-coded chipless RFID," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 8, pp. 7057-7072, 2022.

[10] Z. Li, I. Rukhlenko, W. Zhu, "Microwave metasurface hologram for holographic imaging and its data encryption applications," *Journal of Optics*, vol. 24, pp. 11, 2022.

[11] J. Liu, J. Zhang, S. Yin, "Hybrid chaotic system-oriented artificial fish swarm neural network for image encryption," *Evolutionary Intelligence*, vol. 16, pp. 77-87, 2023. https://doi.org/10.1007/s12065-021-00643-5.

[12] S. Liu, "Computer network information security and protection measures under the background of big data," in *Journal of Physics: Conference Series. IOP Publishing*, vol. 1881, no. 3, pp. 032092, 2021.

[13] S. Pal, Z. Jadidi, "Analysis of security issues and countermeasures for the industrial internet of things," *Applied Sciences*, vol. 11, no. 20, pp. 9393, 2021.

[14] M. Ramachandra, M. Srinivasa Rao, W. Lai, J. Babu, K. Hemalatha, "An efficient and secure big data storage in cloud environment by using triple data encryption standard," *Big Data and Cognitive Computing*, vol. 6, no. 4, pp. 101, 2022.

[15] B. Song, R. Qiu, "The influence of digital virtual technology on contemporary college students' ideological and political education," *IEEE Access*, 2020. doi: 10.1109/ACCESS.2020.3020167.

[16] L. Teng, Y. Qiao, M. Shafiq, G. Srivastava, A. Javed, T. Gadekallu, S. Yin, "FLPK-BiSeNet: Federated learning based on priori knowledge and bilateral segmentation network for image edge extraction," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1529-1542, 2023.

[17] S. Yin, H. Li, A. A. Laghari, T. R. Gadekallu, G. A. Sampedro and A. Almadhor, "An anomaly detection model based on deep auto-encoder and capsule graph convolution via sparrow search algorithm in 6G internet-of-everything," *IEEE Internet of Things Journal*, 2024. doi: 10.1109/JIOT.2024.3353337.

[18] S. Yin, H. Li, L. Teng, A. Laghari, V. V. Estrela, "Attribute-based multiparty searchable en-

cryption model for privacy protection of text data," *Multimedia Tools and Applications*, 2023. https://doi.org/10.1007/s11042-023-16818-4

[19] S. Zhang, T. Yao, V. K. Arthur Sandor, T. H. Weng, W. Liang, J. Su, "A novel blockchain-based privacy-preserving framework for online social networks," *Connection Science*, vol. 33, no. 3, pp. 555-575, 2021.

Biography

Yuankun Du biography. Yuankun Du is with the School of Big Data and Artificial Intelligence, Zhengzhou University of Science and Technology, Associate Professor, research direction: Artificial Intelligence, Big data analysis.

Fengping Liu biography. Fengping Liu (Senior Engineer) is with the School of Information Engineering, Zhengzhou University of Science and Technology, Research interests: Software development, Information management and information system.

Fei Wang biography. Wang Fei (1982.7-), male, Ph. D., lecturer, Han nationality, born in Zhengzhou, Henan Province. He is with School of Information, Henan Institute of Animal Husbandry Economics, His research interests: Machine vision and deep Learning.

A Model for Sustainable Data Encryption, Storage, and Processing in Edge Computing-driven Internet of Things

Chenze Huang and Ying Zhong
(Corresponding author: Chenze Huang)
Research and Development Institute of Northwestern Polytechnical University
Shenzhen, Guangdong 518057, China
Email: yxy_0713@qq.com

(Received Dec. 5, 2023; Revised and Accepted Mar. 4, 2024; First Online Apr. 25, 2024)
The Special Issue on Advanced Security System for Network in Fog and Edge Computing
Special Editor: Dr. Salim El Khediri (Qassim University, Saudi Arabia), Dr. Virginia Pilloni (University of Cagliari, Italy), and Dr. Mohamed Lahby (Hassan II University ENS Casablanca, Morocco)

Abstract

Edge computing builds a bridge between IOT devices and data centers. However, due to the low configuration of edge nodes and the limited computing and storage performance, it is difficult for service providers to apply data security models that require many operations to edge computing nodes. In this paper, we propose a unified edge computing data encryption storage and processing model called UdesMec. The model uses a unified edge node data encryption protocol and ensures that the data encryption model can be integrated with the application scenario. In view of the low data processing performance and limited storage capacity of edge nodes, the use of secret sharing and homomorphic encryption algorithms enables all data transmission to be carried out on ciphertext, and most of the calculations are transferred to the cloud server. The encryption model ensures the security of the user's IoT privacy data. The experiment evaluated its encryption performance and system usability and showed that the encryption model guarantees the security of user privacy data.

Keywords: Cloud Computing; Data Security; Edge Computing; Heterogeneous Network; Internet of Things

1 Introduction

With the rapid development of IOT technology and 5G network, intelligent transportation, location service, mobile payment and other new service modes continue to appear [13, 23]. The number of smart phones, wearable devices, network TV and other sensor devices has shown an explosive growth, followed by the massive data generated by IoT terminals [20].

Edge computing is a new service model [21], the data or tasks can be performed computing at the network edge near the data source. Part or all of the computing tasks of the original cloud computing center are migrated to the vicinity of the data source for execution. The network edge can be any functional entity from the data source to the cloud computing center. These entities are equipped with edge computing platforms that integrate the core capabilities of network, computing, storage and application, and provide real-time, dynamic and intelligent computing services for end users. The concept of data processing nearby also provides better structured support for data security and privacy protection [19]. Edge computing plays an increasingly important role in the fields of Internet [29], industrial robot, driverless, intelligent transportation and so on. As a new decentralized architecture, it extends the storage, computing and network resources of cloud computing to the edge of the network to support large-scale collaborative internet applications.

Due to the complexity and real-time of the edge computing service mode, the multi-source heterogeneity of data, and the limited resources of the terminal, the data security and privacy protection mechanism in the traditional cloud computing environment is no longer suitable for the protection of massive data generated by edge devices [2]. Data storage security, sharing security, computing security and privacy protection are becoming more and more prominent [25]. In addition, the advantage of edge computing is that it breaks through the limitation of terminal hardware, and makes portable devices such as mobile terminals participate in service computing, which realizes mobile data access, intelligent load balancing and low management cost. However, it also greatly increases the complexity of access devices. Due to the limited resource of mobile terminals, their data storage and com-

puting capacity and security algorithm execution capacity also have certain limitations.

In the huge terminal network, the data storage security of terminal devices, data sharing security and other issues have become extremely complicated. Because edge computing applications often rely on real-time collected data to control devices, once the central processing node is attacked, wrong control commands are issued to the edge computing network, which can seriously endanger the safety of people and property [11]. In this paper, we propose a unified data encryption storage and processing model for edge computing system. In view of the characteristics of the huge number of sensor nodes in the edge network, the diversity of device, the heterogeneity of the network, etc. [6], a unified edge node data encryption protocol is proposed, and the data encryption model can be integrated with application scenarios. The model divides the entire edge computing system into cloud service layer, edge layer and application layer. In view of the low performance of edge node data processing and limited storage capacity in edge computing [3], we adopt secret sharing and homomorphic encryption algorithms, so that all data is transmitted in ciphertext and most of the computation is transferred to the cloud server. The cloud directly calculate and analyze the ciphertext transmitted from the edge node. The encryption model ensures the security of user privacy data. The innovations of this paper are summarized as follows:

- We design a complete data encryption model suitable for the field of edge computing, which is applied to the edge node and cloud server to ensure that the data collected from the IoT devices can be safely transmitted to the edge layer, cloud service layer and application layer in the form of ciphertext.
- We design an application layer authentication and access control mechanism to enable users to outsource their IoT sensor data to the computing resources provided by cloud service providers without worrying about the security threats of malicious attackers. Cloud service providers can integrate server resources for edge computing, provide services for different user groups in a unified hardware and software system, and adopt an access control mechanism to ensure that each user can only access the data they own.
- Through a large number of experiments, we evaluate the communication and computing performance overhead of the model, test the feasibility of the model in the actual edge computing application scenarios, and prove the effectiveness of our proposed model.

2 Related Work

Nowadays, there are relatively mature and complete security protection schemes and technical systems in cloud computing [22]. However, Due to the new features of lightweight equipments and heterogeneous architecture in edge computing, the security models of traditional cloud computing have become no longer applicable [10]. Among them, the application of cryptographic technology in the edge computing environment is of great significance to solve data security issues [18].

In the edge computing architecture, cloud servers, edge servers and users have different ownership and control rights to key information, which makes key negotiation for edge computing environment more complex [4]. Both the communication between edge nodes and the communication between edge nodes and cloud servers require the support of key agreement technology. In addition, any link in the edge computing model requires a security key as protection. For example, the device layer requires an authentication key, and the communication layer requires a session key. A complete and secure key system is essential. Jia *et al.* [7] designed an identity-based anonymous authentication key agreement protocol for the mobile edge environment. The protocol can complete the identity verification process of both parties only in a single message exchange round, and guarantees user anonymity and non-traceability. Although the computing time has obvious advantages compared with other schemes, the specific implementation of the protocol is still limited by the computing power of the devices. The authors described a three-party key negotiation scheme [8] applied in CK security model, however, it does not provide anonymity protection for devices and servers.

Encryption of communication data is the most basic and common application of cryptography in the field of edge computing [28]. Compared with the cloud, the data storage capacity of the edge network is much less, and the data stored in the edge can be localized information shared by multiple users. Since the public key encryption mechanism is still too complex in the edge computing framework [12], terminal devices with limited resources cannot afford such a huge amount of calculation when performing decryption operations. How to lighten the encryption algorithm is still an important and challenging task. The most common public key encryption system is an identity-based public key encryption system [10]. The identity encryption system takes the identity information of the subject as input, outputs the corresponding public key information through the key derivation function, and generates the corresponding private key [5]. This approach solves the cost problem when using the digital certificate. Kim *et al.* [9] proposed an identity-based broadcast encryption technology to support data encryption in the edge computing architecture. By entrusting partial decryption to the edge, it greatly reduces the computing and communication overhead required by the terminal device. When the data in the edge device needs to be further processed and analyzed, the edge devices outsource the data to a third party to obtain more advanced computing resources and services, which results in the separation of the ownership and the data control. For the problem

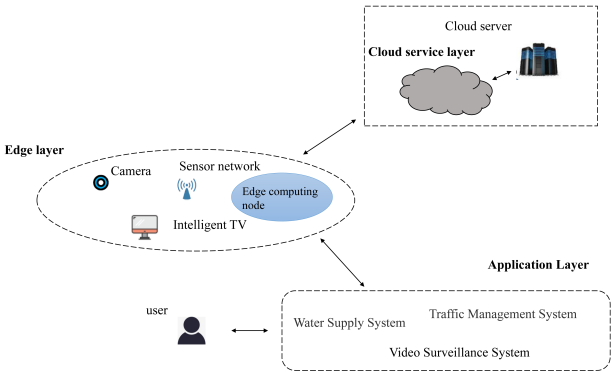


Figure 1: Hierarchical security integration model diagram

of data outsourcing, fully homomorphic encryption [15] is a feasible scheme, and its characteristics ensure that the results of algebra operation on ciphertext are consistent with that of plaintext after algebra operation.

Traditional cloud computing-oriented privacy protection mechanisms are not fully applicable to edge devices [4], and the algorithm execution ability carried by edge nodes is also limited. However, the existing related work cannot integrate cloud computing and edge computing well [20]. In this paper, we propose a hierarchical security integration model of end-cloud integration, data can be safely stored in the cloud and edge nodes according to application requirements. Furthermore, the model can safely carry out end-cloud data interaction, serve relevant applications, and give full play to the end-cloud integration of cloud computing and edge computing.

3 UdesMec Model

The entire hierarchical security integration model is shown in Figure 1, which consists of three layers: cloud service layer, edge layer and application layer.

Edge Layer: It is composed of IoT device networks and edge computing nodes. Each terminal network has a certain number of edge computing nodes responsible for the network data collection and processing. They are usually computers, servers or base stations with computing performance. They collect data from IoT terminals, and provide certain data calculation and data storage function.

Cloud Service Layer: The cloud service layer is usually composed of cloud servers, which are provided by service providers and are responsible for receiving encrypted pre-processed data from edge computing nodes. Since the data is homomorphic encrypted by the edge node, it can be encrypted directly in the cloud server, without any decryption operation in the cloud service layer, and do not disclose the privacy data of any IoT device.

Application Layer: For data owners, they connect a large number of IoT devices to the whole edge computing network through the application software provided by the service provider. The IoT devices transmit data to the edge nodes and store them in the cloud. Users need to enjoy corresponding services in the application system provided by the service provider, such as viewing and analyzing data. For service providers, they need to maximize the use of their own server resources to provide services to the customer. The system can meet the different users access requirements for different data, and has a complete user authentication and access control mechanism to ensure the data security and ensure that the user data is not stolen by other malicious attackers.

3.1 Edge Layer Ciphertext Model

This section mainly takes the system implemented in the experimental part as an example to introduce the edge layer ciphertext model. The IoT terminals are cameras in the distributed camera network, which transmits the original video data to the edge computing, and the edge computing node identifies the person in the image through the image processing function, and generates the person's coordinates, the person's color histogram and other data. The character color histogram is structured data, due to the large number of data lines, it is regarded as unstructured data and transmitted in the form of file in the security integrated system. For the data model of the system, it is necessary to generate an independent meta database for each edge node to store independent keys and other information.

Given an edge node a, according to RSA algorithm, randomly generate two prime numbers p_1 and p_2 (usually use prime numbers exceeding 512 bits, such as 1024 bits) to obtain their product n. As shown in Formula (1), two prime numbers $p_1 = 53$, $p_2 = 59$ are generated for node a, and $n = 3127$, $\varphi(n)= 3016$ are obtained (To ensure the readability of the data and simplify the calculation, the keys are all set to smaller values.). It is also necessary to generate a random number p, which represents a positive integer that is relatively prime to n.

$$\varphi(n) = (p_1 - 1)(p_2 - 1) \tag{1}$$

After setting the metadata of the edge node, the edge node is initialized and can receive the original video data from cameras. After the recognition processing of each frame of camera data, the structured data as shown in Table 1 is obtained. It mainly includes the frame sequence number id, camera number cam_id, time stamp $timestamp$, coordinates of the person square x_1, y_1, x_2, y_2, and RGB histogram file of the person. The symbols S and PRI are reserved fields for ciphertext operation. Edge node a generates a separate column key ck for each field, and obtains a ciphertext value V_e of each field according to the generated n. V_{key} is generated by ck_A and r_i, as shown in Formula (2).

$$V_{key} = g(r, (x, y)) = xp^{ry \bmod \varphi(n)} \bmod n \tag{2}$$

Then, V_e is generated through V_{key} and plaintext V. V_e is the ciphertext value after the data is encrypted, as shown in Formula (3).

$$V_e = E\left(V, V_{key}\right) = VV_{key}^{-1} \bmod n \tag{3}$$

where V_{key}^{-1} represents the modular inverse of V_{key}.

$$V_{key}V_{key}^{-1} \bmod n = 1 \tag{4}$$

The encrypted structured data contains the extracted structured data such as person's coordinates, while the histogram provides the required person recognition data information in the form of file. The edge node only needs to upload the ciphertext value V_e and the person histogram file to the cloud service layer, which is stored and processed by the cloud server. In the case of limited storage capacity of edge nodes, only small original video data can be retained, or even the original video data cannot be retained. In addition, due to the use of the RabbitMQ message queue to transmit data between the cloud service layer and the edge layer, once the edge layer encounters a weak network or no network environment, the data will wait at the edge node until the network is restored and try to send again, without losing data due to network delays.

3.2 Ciphertext Model of Cloud Service Layer

The cloud server collects the frame data from all edge nodes. Due to the uncontrollable factors such as network delay, the same frame data from different nodes do not be strictly delivered to the cloud server at the same time, and the frame synchronization between different devices needs to be carried out through the frame sequence number. Therefore, it can be seen from Table 2 that the frame sequence number is not encrypted, but stored in plaintext, which is convenient for facilitating data synchronization of cloud service and avoiding decrypting the data in each frame synchronization. After the frame synchronization of the cloud server, the system compares the characters with the gallery characters, uses the character re-identification technology to obtain the trajectory data of the characters appearing under the camera, and stores them in ciphertext.

The cloud service layer directly stores ciphertext on the cloud server to ensure the security of the character track data. Because the data model adopts a homomorphic encryption algorithm, the data owner can directly perform the required operation on ciphertext, and obtain track plaintext data by decrypting data when it is necessary to view data.

3.3 User Authentication and Key Transmission in Application Layer

The application layer is composed of service providers and users. A service provider owns the cloud server resources and deploys the application of security integration model at the edge node. It transfers and stores the database key (includes n, p, p_1, p_2 and column key ck) from the edge node in its own security database, acting as a trusted third party (TTP) provides authentication mechanism for data owners to view and process their own data. In addition, the key information stored in the edge node can also be saved by the data owner, and the encryption and decryption client application can be designed by itself. This model can also ensure the security of the data in the cloud service provider.

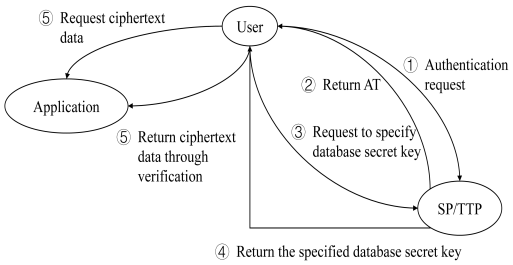


Figure 2: Flow of ciphertext acquisition

Since the service provider manages the data key of the edge node, when users need to view the plaintext data, they need to obtain the key for decryption. The model uses a user authentication mechanism, and users obtain their corresponding keys through identity verification. The service provider (SP) has two entity resources: cloud server and TTP. The user first sends a user login request to TTP, and applies for an access token (AT) with its own identity information. After TTP verifies the user identity, it returns AT generated by the user. The user authorizes AT to the client application, and the system uses AT to apply to SP for database key information. After SP verifies the authenticity of AT, it returns the database key information (n, p, $\varphi(n)$, ck, etc.) to the client application. After the client application obtains the key information, it uses AT to apply to the cloud server under the jurisdiction of SP to obtain the ciphertext. After the system in the cloud server verifies AT, it transmits the ciphertext of the specified database to the client application according to the information carried by AT. The client application uses the database key to decrypt by itself, and returns the plaintext data to the client, Figure 2 describes the acquisition process of the entire ciphertext.

3.4 Data Analysis and Application

3.4.1 Ciphertext Data Operation in Cloud Service Layer

In the security integration model, the IoT terminal device data is preprocessed by the edge node and uploaded to the cloud for encrypted storage. The data is stored in the cloud in the form of ciphertext, which can support the data owner to perform operations on the cloud data and get the ciphertext calculation results. After the cloud

Table 1: Comparison of communication overhead of the ciphertext model

Field	Plaintext value V	Column key ck	Ciphertext value V_e
Frame sequence number *id*	1	NULL	NULL
Camera number *cam_id*	1	(2,2)	391
Timestamp *timestamp*	0001	(2,3)	1759
Character coordinate x_1	23	(2,2)	2739
Character coordinate y_1	94	(5,7)	1163
Character coordinate x_2	194	(1,3)	806
Character coordinate y_2	471	(11,13)	503
Histogram data file *histogram*	*xxx,npy*	NULL	NULL
S	1	(3,1)	2606
PRI	3	(2,3)	391

Table 2: An example of the data structure of the character track in the cloud server ($n = 3127$, $p = 2$)

Field	Plaintext value V	Column key ck	Ciphertext value V_e
Frame sequence number *id*	1	NULL	NULL
Camera number *cam_id*	1	(2, 2)	391
Time stamp *timestamp*	0001	(2, 3)	1759
Character coordinate x_1	23	(2, 2)	2739
Character coordinate y_1	94	(5, 7)	1163
Character coordinate x_2	194	(1, 3)	806
Character coordinate y_2	471	(11, 13)	503
S	1	(3, 1)	2606
PRI	3	(2, 3)	391

re-identifies the person in each frame of data that is transmitted, it needs to calculate the relative position of the person in the camera and use it as a support to draw the person trajectory map. This operation needs to add the ciphertext values x_1 and x_2 of the coordinate data to obtain the ciphertext result, and then decrypt the ciphertext at the data owner to obtain the relative position of the character in the x-axis of the video. The UdesMec system supports most operators of SQL statements. First, the encrypted query implementation of addition, subtraction and multiplication operators is introduced.

1) Multiplication Operator

The data table T has two encrypted columns, A and B. Which have column secret keys $ck_A = \langle x_A, y_A \rangle$ and $ck_B = \langle x_B, y_B \rangle$ respectively. Assuming that the result of multiplying A and B is column C, the column key of column C is $ck_C = \langle x_C, y_C \rangle$. To obtain the value of C from the values of A and B, C_e and ck_C need to be calculated. The protocols of the user side *mul_x* (Custom Protocol Stack) and *mul_y* (Custom Protocol Stack) are executed to obtain ck_C as follows.

$$ck_C = \langle x_C, y_C \rangle = \langle x_A y_B, x_A + y_B \rangle \tag{5}$$

Execute the protocol *mul_y_c_e* on cloud database to get c_e as follows.

$$C_e = A_e B_e mod n \tag{6}$$

According to formulas (2)∼(4), the following formula is obtained.

$$C_{key} = x_C \cdot p^{ry_C} = A_{key} \cdot B_{key} \, (mod \; n) \tag{7}$$

Therefore, it can be proved that

$$C = C_e \cdot C_{key} = A \cdot A_{key}^{-1} \cdot B \cdot B_{key}^{-1} \cdot A_{key} \cdot B_{key} = A \cdot B \tag{8}$$

2) Addition and Subtraction Operators

For multiplication, the result column can be obtained by multiplying A_e and B_e, and ck_C can be obtained by ck_A and ck_B. To support addition and subtraction operators, the key update operation U and two auxiliary columns S and PRI need to be introduced to complete the operation.

The value of each row in column S is a constant 1, while each value in PRI is a random prime number. The key update operation U can generate a new column, so that $C = U\,(A, < x_C, y_C >)$. The generated column C is equal to column A updated with operator U (C=A), and ck_C has a specific initial value according to the needs of operation. $S_e = S_{key}^{-1}$ can be obtained from Formula (3). Define two temporary variables j and k, and make the user perform the following operations to obtain j and k.

$$j = y_S^{-1} \, (y_C - y_A) \, mod \; \varphi \, (n) \tag{9}$$

$$k = x_A x_S^j x_C^{-1} \tag{10}$$

After getting j and k, which are sent to the cloud database. The cloud database performs the following operations.

$$C_e = k \cdot A_e \cdot S_e^j \tag{11}$$

After calculation, the value of column C is equal to that of column A, which can be proved as follows.

$$C = C_e C_{key} = x_A \cdot x_S^j \cdot A_e \cdot \left(S_{key}^{-1}\right)^j \cdot p^{ry_C} = A \tag{12}$$

The key update operation U assists other operators to complete the operation. The key update operation U is designed to ensure the safe operation of these operators, its own security must be guaranteed. In Section III.E, we will prove the security of key update operation.

As with multiplication, consider the operation C=A+c, where c is a constant. First, calculate $A + (S \cdot c)$, S is a constant column. Then C= A+B is calculated to complete the operation.

In the edge node database, the metadata is $n = 3127$, $p = 2$, $p_1 = 53$, p_2= 59. According to the Formula (4), the two coordinate values $x_1 = 23$ (ck_{x_1} = ⟨2,2⟩) and $x_2 = 194$ (ck_{x_2} = ⟨1,3⟩) are encrypted.

$$V_{key} = g\left(r, (x, y)\right) = xp^{ry \bmod \varphi(n)} \bmod n \tag{13}$$

The ciphertext values are $v_{e_{x1}} = 2739$ and $v_{e_{x2}}$= 806, which come from the same frame data. The two ciphertext values are stored in the cloud service layer. The cloud does not need to obtain the metadata, but only needs to perform the key update operation according to the following formula.

$$j = y_S^{-1}\left(y_C - y_A\right) mod \varphi\left(n\right) \tag{14}$$

$$k = x_A x_S^j x_C^{-1} \tag{15}$$

Then, according to Formula (16), the ciphertext is added ($v_{e_{x1}}$ and $v_{e_{x2}}$) in the cloud to obtain the temporary ciphertext $v_e = 834$.

$$v_e = v'_{e_{x1}} + v'_{e_{x2}} \tag{16}$$

Through Formula (17), the plaintext value corresponding to ciphertext 834 is 217, which is consistent with the result of direct plaintext addition.

$$V = D\left(V_e, V_{key}\right) = V_e V_{key} \bmod n \tag{17}$$

The temporary ciphertext v_e= 834 can be stored in the cloud server according to the user needs, waiting for the application layer to obtain it, or performing other operations on it. Due to the support of homomorphic encryption algorithm, the cloud can calculate the ciphertext without decryption, which meets most of the data operation and analysis needs.

3.4.2 User Layer Data Application

The data owner directly performs data analysis operations on the cloud encrypted data in the application layer system, and transfers all calculation and storage requirements to the cloud service layer. It only needs to obtain the ciphertext result and decrypt it. In the edge node of the edge layer, only the unique database key of the node is stored, which avoids the privacy data leakage of other edge nodes due to the capture of the edge node by an attacker. In addition, the attacker can only obtain the key data of current edge node database, but because the attacker cannot pass the TTP user authentication, and the ciphertext of cloud service layer is only open to the users who pass the TTP authentication. The attacker cannot disguise as a normal user to log in to the system and use the stolen key to obtain the data. The user group (data owner) can pass the TTP verification and use the key to directly perform ciphertext operations on the ciphertext existing in the cloud, or directly transmit the ciphertext from the cloud to the client for decryption and viewing.

4 Experiments

For the system with the proposed security model (UdesMec), user privacy data is stored and calculated in the form of ciphertext. We mainly analyze the feasibility of the system, and further verify the proposed system can guarantee data security and privacy. The experiment adopts a cloud server (Intel Xeon CPU e3-1270 @ 3.40GHz) as the cloud service layer to perform ciphertext calculations, and four single machines as edge nodes (CPU: Intel Core i5-6200U@2.3GHz , graphics card: NVIDIA GeForce GTX 760 2GB). The camera capture video resolution is 640×480, and the video capture rate is limited to 3FPS. We have developed a human trajectory tracking and trajectory prediction system in a multi-camera scene. The prototype program of the system is written in Python.

4.1 Analysis of Communication Cost of System

In this section, we mainly analyze the communication overhead of introducing a security model, and compare the proposed encryption method with other state-of-art algorithms. The comparison algorithms are listed as follows.

AggBPE [14]: The ciphertext is stored in the form of $v_e = g^m r^n \bmod n^2$ and stored as two different ciphertext x_i and ${x_i}^2$. Suppose that the number of n bits in the experiment is 1024 and the number of n^2 bits is 2048, then the storage bits of the data generated by N IoT devices are 4096×N bits.

LPDA [1]: The algorithm aggregates x_i and ${x_i}^2$ into a ciphertext $c_{is} = \left[1 + n \times a_j \times \left(x_i \times \alpha_0 + x_i^2\right)\right] \bmod n^2$,

Therefore, the number of bits stored by LPDA is $2048 \times N$.

TPCS [26]: the algorithm uses a pseudonym and a Paillier password system to protect the privacy of data processors. In addition, it designs three authentication protocols, which ensures that only the legitimate processor can pass the authentication.

The communication overhead of the three models under different equipment numbers is shown in Table 3. It can be seen that the security model proposed in this section has obvious advantages over other encryption algorithms in terms of communication overhead.

Table 3: Comparison of communication overhead of ciphertext models

IoT device number	AggBPE	LPDA	TPCS	UdesMec
0	0	0	0	0
100	0.45	0.28	0.37	0.19
200	0.75	0.46	0.69	0.28
300	1.21	0.58	1.12	0.39
400	1.58	0.76	1.38	0.41
500	2.08	1.1	1.76	0.5
600	2.43	1.22	2.18	0.56
700	2.78	1.43	2.34	0.69
800	3.18	1.54	2.83	0.73
900	3.58	1.69	3.07	0.93
1000	4.05	2.01	3.29	1.1

4.2 Analysis and Evaluation of System Operation Cost

To ensure the fairness of the experiment, we set the parameters of all models to the same number of bits and perform encryption operations on the plaintext value. The experiment is runs 50 times, and the average value is taken as the result of the experiment. The experimental results are shown in Figures 3 and 4. The experimental results show that compared with the plaintext operation, the proposed algorithm and the other three models have certain performance overhead. But compared with the other three algorithms, the ciphertext model proposed in this paper has less computing overhead and better performance both at the edge and in the cloud.

The model is oriented to a unified edge computing data processing scenario and is suitable for different structured data models. To evaluate the computational cost of the ciphertext model for different data processing, we also analyze the average time cost of the ciphertext under common operators. The results are shown in Table **??**. Which lists the multiples of the operation time of the three different operators in the ciphertext compared to the operation time of the plaintext. For the encryption model, the addition and subtraction operation is compared with the multiplication operation, because the additional key

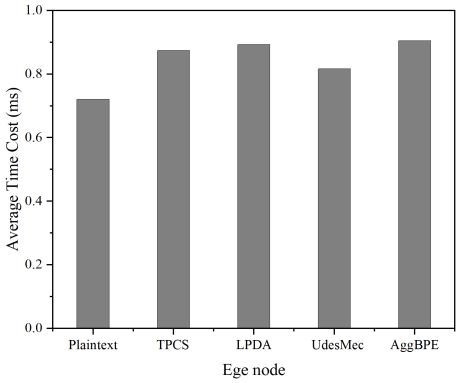


Figure 3: Average encryption cost per frame

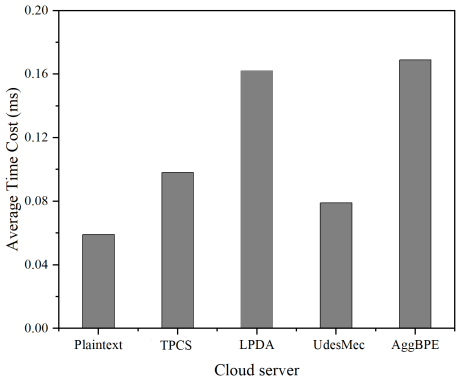


Figure 4: Average encryption cost per frame

update mechanism is added, which causes the operation to take longer. Due to the development of modern computer hardware, ordinary addition and subtraction operations do not constitute any overhead to the hardware. For example, although the ciphertext model has reached 50 times that of plaintext addition and subtraction, it is only 0.0087ms and 0.0084ms per frame, which is limited for the system overhead.

Table 4: The multiple of operation cost of data ciphertext operator relative to plaintext cost

Operation	Multiply	Add	subtract
Average Time Cost	9.86×	58.19×	51.86×

4.3 System Feasibility Assessment

In the experiment, we use nine cameras to simulate a camera network. In this network, the camera continuously collects raw video data, the number of frames per second is set to 3, and the collected images are analyzed using the YOLO algorithm [16] to extract the movement information and color histogram features of the characters appearing in the video. The communication between the cloud service layer and the edge layer adopts socket communication mode based on TCP transmission, and the obtained frame data is directly transmitted to the cloud

server. The cloud server performs ciphertext calculations on the received data in real time to obtain the ciphertext information of the relative position of the person. In the experiment, assuming that the width of the camera resolution is *Camw* and the length is *Camh*, the calculation formula of the relative position *rp* of the person under the x-axis of the camera is as follows:

$$rp = \frac{(x_1 + x_2)/2 - Camw/2}{Camw/2} \tag{18}$$

From the above formula, it can be seen that to obtain the relative position *rp*, a non-integer multiplication operation is required for the encrypted value. However, the data in the system data model can only be calculated in the integer range. Therefore, for operations involving decimals, if the user needs the final plaintext result, a compromise method needs to be taken to retain the ciphertext value of x_1+x_2 and decrypt it in the application layer. In actual scenarios, corresponding coding designs can be designed according to different computing requirements. In the experiment, two scenarios are used to verify the system delay.

The experimental results are shown in Figures 5 and 6. In the first scenario, the average processing time of each frame is about 0.35s, and the maximum delay reaches 0.37s. In the second scenario, compared with the first scenario, the processing time of each frame rises to about 0.37s. This is because as the number of cameras increase accordingly, the collected data increases correspondingly, and the cloud server and edge nodes need more time to process these data.

However, the computational pressure caused by the increased cameras is shared by the edge nodes, and the cloud server only needs to process simple structured data. The average processing time per frame in the second scenario is only increased 0.02s compared to one-camera scenario. For 3FPS video, the time to generate frame data is about 0.334s, the security integrated system can display the character track data before the current time point to the user in near real time in both scenarios. In a large-scale distributed camera network, with the help of the edge computing model, more edge nodes can be deployed to ensure the efficiency of camera data processing. It is also possible to increase the cloud server or improve the configuration of the cloud server to ensure the efficiency of data aggregation processing, thereby ensuring the scalability of the system.

4.4 Query Performance

The main purpose of TPC-H (TPC Benchmark-H) [27] is to evaluate the decision support ability of specific queries. The benchmark simulates the database operation in the decision support system, tests the response time of complex queries in the database system, and takes the number of queries executed per hour as the measurement index. For the analysis of query performance, the experiment uses TPC-H to test the system. TPC-H test includes all commonly used query operators and complex queries. TPC-H usually means that the database can support routine use and deal with some complex business scenarios. It is compared with other two widely studied algorithms of encrypting database model (MONOMI [24] and CryptDB [17]).

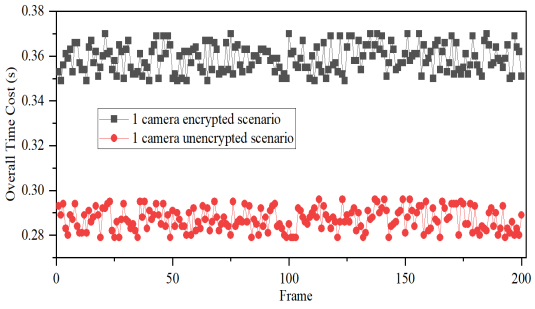


Figure 5: Delay analysis in real time processing (1 camera)

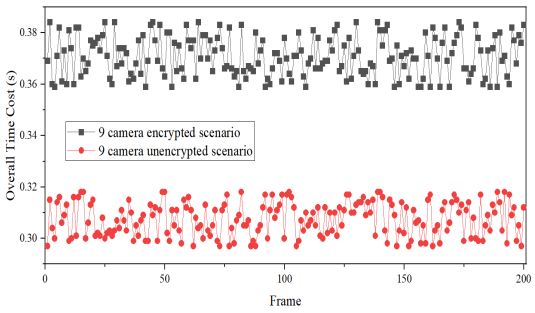


Figure 6: Delay analysis in real time processing (9 camera)

In the experiment, all the statements of TPC-H can be executed correctly. Tables 4∼ 6 show the ratio of the execution time of the 22 sentences of UdesMec executing TPC-H to the execution time of the plaintext query. The sentences Q4, Q11, Q12, Q13, Q16, and Q21 are not listed in UdesMec because they do not involve ciphertext operations. CryptDB and MONOMI models cannot support Q13, Q15 and Q16. It can be seen that UdesMec is much more efficient than CryptDB in the execution time of most statements, which is close to MONOMI.

5 Conclusion

Aiming at the large number of edge nodes and the characteristics of heterogeneous network, we propose a unified edge computing data encryption storage and processing model in this paper. Because the edge node has the characteristics of low data processing performance and limited storage capacity, secret sharing and homomorphic encryption algorithms are used to enable all IoT terminal device data to be calculated in cipher text, transferring most of the amount of computation to the cloud service layer. The model reduces the computing and storage pressure of edge computing nodes, and most of the calculation en-

Table 5: The proportion of execution time with plaintext (Q1∼Q7)

Methods	Q1	Q2	Q3	Q4	Q5	Q6	Q7
CrptDB	38.17X	2.4X	4.6X	3.5X	3.45X	6.7X	2.8X
MONOMI	2.6X	2.5X	2.4X	2.1X	1.9X	2.2X	1.7X
UdesMec	11.29X	1.9X	1.9X	NULL	1.89X	15.79X	1.68X

Table 6: The proportion of execution time with plaintext (Q8∼Q14)

Methods	Q8	Q9	Q10	Q11	Q12	Q13	Q14
CrptDB	5.6X	4.8X	4.94X	5X	4.95X	NULL	6.1X
MONOMI	2.42X	2.45X	2.5X	2.54X	2.5X	NULL	2.51X
UdesMec	2.48X	2.4X	2.45X	NULL	NULL	NULL	2.43X

sures is transferred to the cloud service layer. The model reduces the computing and storage pressure of edge computing nodes, and ensures that device data can still be efficiently collected and processed under the encryption algorithm. The security integration model also includes an application-layer authentication mechanism to ensure the privacy of data owners. At the same time, service providers can integrate their own server resources, provide services for different user groups in a unified software and hardware system, and use an access control mechanism to ensure that each user can only access the data they own. The experiment evaluated the communication cost and computing cost of the model, and tested the feasibility of the model in actual application scenarios.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant Nos. 62002226 and 61772018.

References

[1] C. Chakraborty, S. B. Othman, F. A. Almalki, and H. Sakli, "Fc-seeda: Fog computing-based secure and energy efficient data aggregation scheme for internet of healthcare things," *Neural Computing and Applications*, vol. 36, no. 1, pp. 241–257, 2024.

[2] Y. Chen, D. Wang, N. Wu, and Z. Xiang, "Mobility-aware edge server placement for mobile edge computing," *Computer Communications*, vol. 208, pp. 136–146, 2023.

[3] P. Cong, J. Zhou, L. Li, K. Cao, and K. Li, "A survey of hierarchical energy optimization for mobile edge computing: A perspective from end devices to the cloud," *ACM Computing Surveys*, vol. 53, no. 2, p. 38, 2020.

[4] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4321–4334, 2020.

[5] J. Han, L. Chen, S. Schneider, H. Treharne, S. Wesemeyer, and N. Wilson, "Anonymous single sign-on with proxy re-verification," *IEEE transactions on information forensics and security*, vol. 15, no. 1, pp. 223–236, 2020.

[6] C. Hnab, C. Ylab, C. Fsab, and D. Lty, "Heterogeneous edge computing open platforms and tools for internet of things," *Future Generation Computer Systems*, vol. 106, pp. 67–76, 2020.

[7] N. Kaliya and D. Pawar, "Unboxing fog security: a review of fog security and authentication mechanisms," *Computing*, vol. 105, no. 12, pp. 2793–2819, 2023.

[8] N. H. Kamarudin, N. H. S. Suhaimi, F. A. Nor Rashid, M. N. A. Khalid, and F. Mohd Ali, "Exploring authentication paradigms in the internet of things: A comprehensive scoping review," *Symmetry*, vol. 16, no. 2, p. 171, 2024.

[9] J. Kim, "Studies on inspecting encrypted data: Trends and challenges," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, pp. 189–199, 2023.

[10] H. Liu, P. Zhang, G. Pu, T. Yang, S. Maharjan, and Y. Zhang, "Blockchain empowered cooperative authentication with data traceability in vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4221–4232, 2020.

[11] J. Liu, X. Wang, S. Shen, G. Yue, S. Yu, and M. Li, "A bayesian q-learning game for dependable task offloading against ddos attacks in sensor edge cloud," *IEEE Internet of Things Journal*, 2020.

[12] G. K. Mahato and S. K. Chakraborty, "Securing edge computing using cryptographic schemes: a review," *Multimedia Tools and Applications*, pp. 1–24, 2023.

[13] A. Matin, M. R. Islam, X. Wang, H. Huo, and G. Xu, "Aiot for sustainable manufacturing: Overview, challenges, and opportunities," *Internet of Things*, p. 100901, 2023.

[14] S. Mehrotra, S. Sharma, J. Ullman, and A. Mishra, "Partitioned data security on outsourced sensitive and non-sensitive data," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 650–661.

Table 7: The proportion of execution time with plaintext (Q15∼Q22)

Methods	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22
CrptDB	NULL	NULL	5.65X	59.94X	5.64X	6.4X	NULL	4.8X
MONOMI	NULL	NULL	2.51X	2.56X	2.5X	2.61X	NULL	1.9X
UdesMec	5X	NULL	2.45X	22.14X	2.45X	2.48X	NULL	1.74X

[15] A. Murugesan, B. Saminathan, F. A. Urjman, and R. L. Kumar, "Analysis on homomorphic technique for data security in fog computing," *Transactions on Emerging Telecommunications Technologies*, no. 4, 2020.

[16] Y. Sakai, H. Lu, J.-K. Tan, and H. Kim, "Recognition of surrounding environment from electric wheelchair videos based on modified yolov2," *Future Generation Computer Systems*, vol. 92, pp. 157–161, 2019.

[17] E. Saleh, A. Alsa'deh, A. Kayed, and C. Meinel, "Processing over encrypted data: between theory and practice," *ACM SIGMOD Record*, vol. 45, no. 3, pp. 5–16, 2016.

[18] X. Shang, Y. Huang, Z. Liu, and Y. Yang, "Reducing the service function chain backup cost over the edge and cloud by a self-adapting scheme," *IEEE Transactions on Mobile Computing*, 2021.

[19] S. Shen, H. Ma, E. Fan, K. Hu, S. Yu, J. Liu, and Q. Cao, "A non-cooperative non-zero-sum game-based dependability assessment of heterogeneous wsns with malware diffusion," *Journal of Network and Computer Applications*, vol. 91, pp. 26–35, 2017.

[20] G. Srivastava, C. W. Lin, D. Pamucar, and S. Kotsiantis, "Editorial: Applications of fuzzy systems in data science and big data," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 1, pp. 1–3, 2021.

[21] V. Stephanie, M. Chamikara, I. Khalil, and M. Atiquzzaman, "Privacy-preserving location data stream clustering on mobile edge computing and cloud," *Information Systems*, no. 2, p. 101728, 2021.

[22] H. Tabrizchi and M. K. Rafsanjani, "A survey on security challenges in cloud computing: issues, threats, and solutions," *The Journal of Supercomputing*, pp. 1–40, 2020.

[23] C. Wang, "Construction and deployment of a distributed firewall-based computer security defense network," *Int. J. Netw. Secur*, vol. 25, pp. 89–94, 2023.

[24] L. Wiese, T. Waage, and M. Brenner, "Clouddb-guard: A framework for encrypted data storage in nosql wide column stores," *Data & Knowledge Engineering*, vol. 126, p. 101732, 2020.

[25] Z. Wu, S. Shen, X. Lian, X. Su, and E. Chen, "A dummy-based user privacy protection approach for text information retrieval," *Knowledge-Based Systems*, vol. 195, p. 105679, 2020.

[26] C. Zhang, L. Zhu, C. Xu, K. Sharif, X. Liu, X. Du, and M. Guizani, "Achieving trust-based and privacy-preserving customer selection in ubiquitous computing," *arXiv preprint arXiv:1902.04345*, 2019.

[27] J. Zhang, A. Sivasubramaniam, H. Franke, N. Gautam, Y. Zhang, and S. Nagar, "Synthesizing representative i/o workloads for tpc-h," in *10th International Symposium on High Performance Computer Architecture (HPCA'04)*. IEEE, 2004, pp. 142–142.

[28] J. Zhang, H. Zhong, J. Cui, M. Tian, Y. Xu, and L. Liu, "Edge computing-based privacy-preserving authentication framework and protocol for 5g-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7940–7954, 2020.

[29] H. Zhou, S. Shen, and J. Liu, "Malware propagation model in wireless sensor networks under attack–defense confrontation," *Computer Communications*, vol. 162, pp. 51–58, 2020.

Biography

Chenze Huang is a doctoral candidate at Northwestern Polytechnical University. In 2020, he received a master's degree in computer science from Northwestern Polytechnical University. His research direction is network security, artificial intelligence, etc.

Zhong Ying is a doctoral candidate at Northwestern Polytechnical University. He graduated from Shanghai Jiao Tong University with a master's degree in cybersecurity in 2021. He received his Bachelor of Engineering degree from Shanghai Jiao Tong University in 2019. His research interests are network security, data mining, etc.

Research on Abnormal Traffic Intrusion Detection for Power Generation Enterprise Network

Li Tian

(Corresponding author: Li Tian)

State Grid Hubei Electric Power Research Institute
No. 227, Xudong Street, Hongshan District, Wuhan City, Hubei 430077, China
Email: tianl1987@hotmail.com

(Received Jan. 26, 2023; Revised and Accepted Oct. 21, 2023; First Online Apr. 25, 2024)

Abstract

The normal functioning of the power generation enterprise network is closely tied to the functioning of the power generation system, and the necessity of detecting abnormal traffic for intrusion prevention cannot be overstated. Firstly, aiming at the imbalance between abnormal and normal traffic in the power generation enterprise network, this paper proposed an improved synthetic minority oversampling technique (ISMOTE) method to process the balance of the dataset. Then, an improved white shark optimizer (WSO) was designed to optimize the extreme learning machine (ELM) parameters, i.e., using the improved white shark optimizer-extreme learning machine (IWSO-ELM) algorithm to realize abnormal traffic intrusion detection. It was found that the dataset, after applying the ISMOTE, achieved better performance in intrusion detection. The IWSO algorithm improved the intrusion detection effect of ELM by more than 10%. The F1 value of the IWSO-ELM algorithm on the NSLKDD and UNSWNB15 datasets reached 95.10% and 99.34%, respectively, which was better than the decision tree, support vector machine, and other methods. The findings prove the effectiveness of the IWSO-ELM method, making it applicable to implementation in real-world power generation enterprise networks.

Keywords: Abnormal Flow; Extreme Learning Machine; Optimization Algorithm; Power Generation Enterprises; Synthetic Minority Over-Sampling Technique

1 Introduction

Under the influence of the continuous development of computers and the Internet, power generation enterprises are also increasingly moving towards automation and intelligence, leveraging the potential of the Internet and big data to construct power grids [14], realizing efficient perception and processing of data, and providing people with more convenient and high-quality services. However, with the continuous expansion and update of the power generation enterprise network, the exposure of the network is wider, the structure is more complex, and in the face of security threats, it is also more vulnerable. Attackers may remotely attack the network of power generation enterprises to illegally obtain access rights, steal confidential content, or spread malware and viruses through E-mail, websites, etc., to destroy the hardware equipment of the power system.

Attacks on the network of power generation enterprises will affect the normal functioning of the entire system, resulting in failure of the power system, which is not good for social stability. Therefore, it is particularly important to detect the security of the network of power generation enterprises. At present, most power generation enterprises use manual updating and maintenance methods, by strengthening access control, regular backup data, etc., to protect network security, there are also some encrypted communication algorithms, firewall technology has been applied, but these methods can not meet the needs of the growing expansion of power generation enterprise network, and it is necessary to constantly update and improve the security strategy. In order to deal with more and more complex network intrusions and threats. In the attacks on the network of power generation enterprises, the abnormal traffic intrusion based on distributed denial of Service (DDoS) is very common and typical [1], which is known for its distinctive feature of generating a substantial volume of abnormal traffic. At present, the common method is to analyze the characteristics of abnormal traffic through machine learning and other methods to realize intrusion detection.

Do *et al.* [8] used some data dimensionality reduction techniques and the random forest supervised classification algorithm in their study. They found that the method obtained good results on all measures. The model proposed by Ma *et al.* [15] achieved an accuracy of over 99% on the

test dataset using a combination of kernel support vector machine and linear discriminant analysis. Dandil [7] designed a new method combining negative selection algorithm (NSA) and clonal selection algorithm (CSA) to detect abnormal web traffic. The average accuracy of C-NSA reached 94.30%.

Choi *et al.* [6] developed system based on autoencoder for detecting intrusion and found that the accuracy of the approach reached 91.70%. In order to further enhance the performance of abnormal traffic intrusion detection, this paper designed an improved white shark optimizer-extreme learning machine (IWSO-ELM) approach for power generation enterprise networks and proved the availability of this method through experimental analysis. The proposed approach can achieve accurate and fast detection of abnormal traffic and provides a more efficient means for ensuring the safety of power generation enterprise networks, which is beneficial for further enhancing the security and reliability of power generation enterprise networks.

2 Abnormal Traffic Intrusion Detection Method

2.1 Balanced Processing of Datasets

In the network of power generation enterprises, most of the traffic generated in the communication process is normal traffic, and the number of abnormal traffic is small. In the commonly used abnormal traffic intrusion detection dataset, the proportion of abnormal traffic is also small, that is, there is a large imbalance in the dataset, and this imbalance will lead to inadequate training of the algorithm and affect the detection performance [23]. Therefore, this paper designs an improved synthetic minority oversampling technique (ISMOTE) algorithm to achieve a balanced processing of the dataset.

The principle of SMOT is to randomly generate new sample x_{new} by using nearest neighbor x_k of sample x_k. The used formula is:

$$x_{new} = x + rand(0,1) \times (x_k - x).$$

However, this method may affect the distribution density of the dataset and cause noise samples [2]. In order to avoid the marginalization problem of generating new samples by SMOTE, this paper improves SMOTE. Assuming that there is an imbalanced dataset called D, the majority class samples in it are regarded as the negative class, denoted as D_{maj}, and the minority class samples are regarded as the positive class, denoted as D_{min}. The steps to improve SMOTE are as follows.

1) k samples are randomly selected from D_{min} as clustering centers. Then, each sample is allocated into its nearest cluster. The cluster center was recomputed continuously, and the sample division is adjusted until the cluster center did not change. k clusters (c_i) are obtained.

2) In any cluster c_j, three samples (x_o, x_p, x_q) are selected to form a triangle. The center of gravity sample vector of the triangle is:

$$x_g = (\frac{1}{3}\sum_{j=1}^{3} x_{j1}, \frac{1}{3}\sum_{j=1}^{3} x_{j1}, \cdots, \frac{1}{3}\sum_{j=1}^{3} x_{jn}).$$

3) A new sample called x_{new} is randomly generated between x_g and triangle vertex x_j using the following formula:

$$x_{new} = x_j + rand(0,1) \times (x_j - x_g).$$

4) Steps (2) - (3) are repeated in k clusters until $D_{min} \geq D_{maj}$.

5) A balanced dataset called D_{new} is obtained by synthesizing D_{min} and D_{maj}.

2.2 Extreme Learning Machine

Anomaly traffic intrusion detection is actually the classification problem between abnormal and normal traffic. An extreme learning machine (ELM) has the advantages of few parameters and high efficiency [9] and has been well applied in many fields [20]. Therefore, this paper implements anomaly traffic intrusion detection of power generation enterprise network based on ELM.

Suppose that in ELM, the quantity of neurons in the input, hidden, and output layers is n, l, and m, respectively. For training sample $\{x_i, t_i\}_{i=1}^{N}$, after ELM processing, the output of the i-th sample is expressed as:

$$y_i = \sum_{j=1}^{l} \beta_j f(w_j \cdot x_i + b_j),$$

where β_j, w_j, and b_j are the output, input weight, and threshold of the j-th hidden layer neuron.

Target output t_i can be written as:

$$t_i = \sum_{j=1}^{l} \beta_j f(w_j \cdot x_i + b_j).$$

The matrix form is:

$$T = H\beta,$$

where T, H, and β are the target output, hidden layer output, and hidden layer output weight matrix.

The solution procedure of β is written as:

$$\begin{aligned} H\beta^* - T &= \min_{\beta} ||H\beta - T||, \\ \beta^* &= H^+ T, \end{aligned}$$

where β^* is the least square solution of β and H^+ is the Moore-Penrose generalized inverse of H.

2.3 White Shark Optimizer

There is some blindness in the selection of initial parameters of ELM, which may affect the convergence of the algorithm. Hence, the parameters of ELM are optimized using the white shark optimizer (WSO) in this study. WSO is an algorithm that imitated the navigation and foraging behavior of great white sharks [3], which has the advantages of stable search and high efficiency. Firstly, the shark relies on its exceptional sensory abilities, including acute hearing and a keen sense of smell, to detect the momentary stillness in the water and swiftly approach its prey. The moving velocity of the i-th white shark toward the prey at the $k+1$-th iteration can be written as:

$$\begin{aligned} v_i^{k+1} &= \mu[v_i^k + p_1(w_{gbest_k} - w_i^k) \times c_1 \\ &\quad + p_2(w_{best}^{v_i^k} - w_i^k) \times c_2], \\ \mu &= \frac{2}{|2-\tau-\sqrt{\tau^2-4\tau}|}, \\ p_1 &= p_{max} + (p_{max} - p_{min}) \times e^{-(4k/K)^2}, \\ p_2 &= p_{min} + (p_{max} - p_{min}) \times e^{-(4k/K)^2}, \end{aligned}$$

where μ stands for the constriction factor, τ stands for the acceleration coefficient, generally equal to 4.125, w_{gbest_k} represents the globally optimal position of the shark at the k-th iteration, $w_{best}^{v_i^k}$ is the optimal position of the population that has been known, w_i^k is the position of the i-th shark at the k-th iteration, p_1 and p_2 are influencing parameters, p_{min} and p_{max} are the maximum and minimum speeds of the shark, generally equal to 0.5 and 1.5, respectively, c_1 and c_2 represent random numbers in [0,1].

White sharks continue to search for preys and move towards optimal prey. The position of the i-th shark at the $k+1$-th iteration is written as:

$$\begin{aligned} w_i^{k+1} &= \begin{cases} w_i^k \cdot \neg w_0 + u \cdot a + l \cdot b, & rand < mv \\ w_i^k + v_i^k / f, & rand \geq mv) \end{cases} \\ w_0 &= a \oplus b, \\ f &= f_{min} + \frac{f_{max} - f_{min}}{f_{max} + f_{min}} \\ mv &= \frac{1}{a_0 + e^{(K/2-k)/a_1}}. \end{aligned}$$

where, $\neg$: negate operator; w_0: logical vector; $\oplus$: XOR operation; a, b: unary quadratic vector; u, l: upper and lower bounds of the search space; $rand$: a random number in the range of [0,1] obeying a uniform distribution; f: frequency of wave motion; f_{min}, f_{max}: the minimum and maximum values of wave movement frequency, generally 0.07 and 0.75; mv: the movement force of the shark when it is close to the prey; a_0 and a_1: positive constants, generally 6.25 and 100.

After finding the optimal prey, white sharks start to approach the optimal attack position to kill the prey. The position of the i-th shark relative to its prey during the i-th iteration can be expressed as:

$$\begin{aligned} w_i^{k+1} &= w_{gbest_k} + r_1 \overrightarrow{D_w} sgn(r_2 - 0.5), r_3 < s_s, \\ \overrightarrow{D_w} &= |rand \times (w_{gbest_k} - w_i^k)|, \\ s_s &= |1 - e^{-a_2 \times k/K)}|. \end{aligned}$$

where r_1, r_2, r_3: random numbers in [0,1]; $\overrightarrow{D_w}$: the distance between the shark and its prey; s_s: the olfactory and visual intensity of the white shark; a_2: positive constant, usually 0.0005.

The behavior of the shark is further simulated to update the position of the shark. Then,

$$w_i^{k+1} = \frac{w_i^k + w_i^{k+1}}{2 \times rand}$$

However, the initial population of white sharks is generated by randomization, so the diversity is not strong. In order to improve this, this paper designs an improved WSO (IWSO). Sinusoidal chaotic map [21] is introduced to initialize the population, and the formula is:

$$x_{k+1} = a x_k^2 \sin(\pi x_k),$$

where $x \in [0,1]$, $x_0 = 0.7$, $a = 2.3$.

2.4 Abnormal Traffic Intrusion Detection Method for Power Generation Enterprise Network

The IWSO-ELM method has the following process.

1) The ISMOTE method is used to generate a balanced dataset, and the training and test sets were divided.

2) The ELM and IWSO parameters are initialized.

3) The IWSO algorithm is employed to optimize the weight threshold of ELM, and the fitness was the mean square error (MSE) between the output of the algorithm and the true value.

4) The optimal weight threshold of ELM is obtained. H and β are calculated to establish an ELM model.

5) The test set is used for testing to realize abnormal traffic intrusion detection.

3 Experiment and Analysis

3.1 Experimental Setup

The experiment used Windows 10 operating system with i7-10750H CPU and Python version 3.6. In the IWSO-ELM algorithm, the population size was 100, and the maximum number of iterations was 200. To avoid randomness, each experiment was repeated ten times, and the average value was calculated.

3.2 Experimental Dataset

NSLKDD [19]

It is a dataset commonly used for intrusion detection evaluation, each flow record contains 41 features and a label. There are four kinds of abnormal flows and one normal flow, and the training set and test set are distributed as displayed in Table 1.

Table 1: NSLKDD data distribution

Type	Training	Test
Normal	67,343	9,711
DoS	45,927	7,458
Probe	11,656	2,421
R2L	995	2,754
U2R	52	200

UNSWNB15 [16]

It is a mixed dataset of real normal activity flow and attack flow. Each flow record contains 42 features and a class label. There are nine kinds of abnormal flow and one normal flow. Table 2 presents the distribution of the training set and test set.

Table 2: Distribution of UNSWNB15 data

Type	Training	Test
Normal	560,000	37,000
Analysis	2,000	677
Backdoor	1,746	583
DoS	12,264	4,089
Exploits	33,393	11,132
Fuzzers	18,184	6,062
Generic	40,000	18,871
Reconnaissance	10,491	3,496
Shellcode	1,133	378
Worms	130	44

3.3 Evaluation Indicators

Intrusion detection was usually evaluated on the basis of a confusion matrix (Table 3).

Table 3: Confusion matrix

		Detection results	
		Normal	Abnormal
The real situation	Normal	TP	FN
	Abnormal	FP	TN

In Table 3, TP: normal traffic is detected as normal traffic: FN: normal traffic is detected as abnormal traffic; FP: abnormal traffic is detected as normal traffic; TN: abnormal traffic is detected as abnormal traffic. The specific indicators are presented in Table 4.

Table 4: Evaluation indicators

Indicator	Formula
Accuracy	$ACC = \frac{TP+TN}{TP+TN+FP+FN}$
Precision	$PRE = \frac{TP}{TP+FP}$
Recall rate	$REC = \frac{TP}{TP+FN}$
F1 value	$F1 = 2 \times \frac{(2 \times PRE \times REC}{PRE+REC}$

3.4 Result Analysis

Firstly, the IWSO-ELM algorithm was used in intrusion detection to compare the effects of different dataset balancing processing methods, including:

1) Original dataset;
2) SMOTE [11];
3) ADASYN [22];
4) BorderlineSMOTE [10].

Table 5 displays the results.

Table 5: Comparison of various balancing methods (unit: %)

	ACC	PRE	REC	F1
NSLKDD				
Original dataset	81.12	82.33	80.77	81.54
SMOTE	85.36	83.99	84.14	84.06
ADASYN	88.33	87.19	89.21	88.19
BorderlineSMOTE	92.33	90.02	92.16	91.08
ISMOTE	96.46	93.21	97.07	95.10
UNSWNB15				
Original dataset	83.26	80.17	82.33	81.24
SMOTE	87.64	85.64	86.37	86.00
ADASYN	88.37	86.12	87.06	86.59
BorderlineSMOTE	92.74	95.61	93.27	94.43
ISMOTE	99.83	99.12	99.56	99.34

Table 5 shows that when using the original dataset for intrusion detection, the results were not good. For the NSLKDD dataset, the ACC value was 81.12%, and the F1 value was 81.54%. For the UNSWNB15 dataset, the ACC value was 83.26%, and the F1 value was 81.24%. After using the dataset balance method, there was a noticeable enhancement in the detection outcomes. Although

the results of the SMOTE and ADASYN methods for the above two datasets were improved than the original dataset, they were still less than 90%, and the results of the BorderlineSMOTE method for the two datasets were more than 90%. However, compared to them, the results obtained by the ISMOTE method were better. For the NSLKDD dataset, the ACC value obtained by the ISMOTE method was 96.46%, and the F1 value was 95.10%, which were 4.13% and 4.02% higher than the BorderlineSMOTE method respectively.

For the UNSWNB15 dataset, the ACC value obtained by the ISMOTE method was 99.83%, and the F1 value was 99.34%, which were 7.09% and 4.91% higher than the BorderlineSMOTE method, respectively. These results suggested that the improved SMOTE demonstrated a positive impact in enhancing the efficacy of abnormal traffic intrusion detection. In the actual power generation enterprise network, there is also a phenomenon that the normal traffic is greater than the abnormal traffic, so the method can be applied in the actual power generation enterprise network to accomplish the objective of balancing the dataset. The IWSO was compared with other optimization algorithms in order to prove the improvement of detection effect, including:

1) Without the optimization algorithm;
2) The genetic algorithm (GA) [12];
3) The particle swarm optimization (PSO) algorithm [13];
4) The firefly algorithm (FA) [5];
5) The WSO.

The results are presented in Table 6.

Table 6 shows that without optimization algorithm, the ACC value and F1 value of the ELM method for the NSLKDD dataset were 81.33% and 80.73%, and the ACC value and F1 value of the same method for the UNSWNB15 dataset were 85.77% and 86.07%, respectively. After optimization by different algorithms, the performance of ELM in abnormal traffic detection were improved to varying degrees. In comparison, the GA had poor optimization effect on the ELM method, the PSO algorithm and FA had better optimization effect than the GA, but not as good as the WSO. The IWSO had the best optimization effect on the ELM method. Compared with without the optimization algorithm, the improvement of each index on the two datasets was more than 10%. This result demonstrates the significance of parameter optimization, as it can effectively enhance the detection performance of algorithms.

Finally, the IWSO-ELM method was compared with other current methods for anomaly traffic intrusion detection, including:

1) Decision tree (DT) [18];
2) Support vector machine (SVM) [4];
3) Multilayer perceptron (MLP) [17].

According to Table 7, for the NSLKDD dataset, the DT approach performed poorly, achieving an F1 value of 83.05%, and the SVM and MLP approaches achieved F1 values of 84.25% and 86.94%, respectively, slightly higher than the DT method. The IWSO-ELM method achieved a higher F1 value of 95.10%; compared with the DT, SVM and MLP methods, it was increased by 12.05%, 10.85%, and 8.16%, respectively. For the UNSWNB15 dataset, the F1 value of the IWSO-ELM method was 99.34%, which was 16.09%, 14.76%, and 14.18% higher than the DT, SVM, and MLP methods, respectively. In general, the IWSO-ELM method obtained the best results. Therefore, this method can be applied to the actual power generation enterprise network to realize the protection of the power generation enterprise network.

4 Conclusion

In this paper, an abnormal traffic intrusion detection method called IWSO-ELM was designed for the network of power generation enterprises. Through experimental analysis on two datasets NLSKDD and UNSWNB15, it was found that the detection effect was greatly improved after the ISMOTE balance processing. The IWSO also played a great role in the optimization of ELM parameters. Compared with other algorithms, the IWSO-ELM method had better performance in various indicators, which proves the reliability of this method. The proposed approach can be promoted and applied in real power generation enterprise networks.

References

[1] D. Acarali, K. R. Rao, M. Rajarajan, D. Chema, M. Ginzburg, "Modelling smart grid IT-OT dependencies for DDoS impact propagation," *Computers & Security*, vol. 112, pp. 1-16, 2022.
[2] Asniar, N. U. Maulidevi, K. Surendro, "SMOTE-LOF for noise identification in imbalanced data classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3413-3423, 2021.
[3] M. Braik, A. Hammouri, J. Atwan, M. A. A. Al-Betar, M. A. Awadallah, "White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems," *Knowledge-Based Systems*, vol. 243, pp. 1-29, 2022.
[4] F. Camastra, V. Capone, A. Ciaramella, A. Riccio, A. Staiano, "Prediction of environmental missing data time series by Support Vector Machine Regression and Correlation Dimension estimation," *Environmental Modelling & Software*, vol. 150, pp. 1-7, 2022.
[5] S. R. S. Chakravarthy, H. Rajaguru, "Detection and classification of microcalcification from digital mammograms with firefly algorithm, extreme learning

Table 6: Comparison of different optimization algorithms (unit: %)

	ACC	PRE	REC	F1
NSLKDD				
ELM	81.33	76.94	84.91	80.73
GA-ELM	83.76 (+2.43)	80.07 (+3.13)	86.54 (+1.63)	83.18 (+2.45)
PSO-ELM	86.64 (+5.31)	82.99 (+6.05)	88.67 (+3.76)	85.74 (+5.01)
FA-PSO	89.75 (+8.42)	86.24 (+9.30)	90.12 (+5.21)	88.14 (+7.41)
WSO-ELM	93.26 (+11.93)	89.93 (+12.99)	94.33 (+9.42)	92.08 (+11.35)
IWSO-ELM	96.46 (+15.13)	93.21 (+16.27)	97.07 (+12.16)	95.10 (+14.37)
UNSWNB15				
ELM	85.77	86.03	86.12	86.07
GA-ELM	87.25 (+1.48)	87.22 (+1.19)	88.33 (+2.21)	87.77 (+1.70)
PSO-ELM	91.07 (+5.30)	90.34 (+4.31)	91.45 (+5.33)	90.89 (+4.82)
FA-PSO	93.22 (+7.45)	91.75 (+5.72)	92.24 (+6.12)	91.99 (+5.92)
WSO-ELM	96.34 (+10.57)	95.65 (+9.62)	96.07 (+9.95)	95.86 (+9.78)
IWSO-ELM	99.83 (+14.06)	99.12 (+13.09)	99.56 (+13.44)	99.34 (+13.26)

Table 7: Comparison with other methods (unit: %)

	ACC	PRE	REC	F1
NSLKDD				
DT	83.27	80.64	85.61	83.05
SVM	85.72	82.23	86.37	84.25
MLP	87.53	85.11	88.86	86.94
IWSO-ELM	96.46	93.21	97.07	95.10
UNSWNB15				
DT	85.36	75.77	92.36	83.25
SVM	86.78	80.66	88.89	84.58
MLP	88.66	81.33	89.37	85.16
IWSO-ELM	99.83	99.12	99.56	99.34

machine and non-linear regression models: A comparison," *International Journal of Imaging Systems and Technology*, vol. 30, no. 1, pp. 126-146, 2020.

[6] H. Choi, M. Kim, G. Lee, W. Kim, "Unsupervised learning approach for network intrusion detection system using autoencoders," *Journal of Supercomputing*, vol. 75, no. 9, pp. 5597-5621, 2019.

[7] E. Dandil, "C-NSA: A hybrid approach based on articial immune algorithms for anomaly detection in web traffic," *IET Information Security*, vol. 14, no. 3, pp. 683-693, 2020.

[8] C. X. Do, N. Q. Dam, N. T. Lam, "Optimization of network traffic anomaly detection using machine learning," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2360-2370, 2021.

[9] S. Guo, B. Wu, J. Zhou, H. Li, C. Su, Y. Yuan, K. Xu, "An analog circuit fault diagnosis method based on circle model and extreme learning machine," *Applied Sciences*, vol. 10, no. 7, pp. 1-16, 2020.

[10] H. Han, W. Y. Wang, B. H. Mao, "Borderline-SMOTE: A new over-sampling method in IM balanced data sets learning," in *International Conference on Intelligent Computing*, pp. 878-887, 2005.

[11] Y. D. Huo, Q. Gu, Z. H. Cai, Y. Lei, "Classification method for imbalance dataset based on genetic algorithm improved synthetic minority over-sampling technique," *Journal of Computer Applications*, vol. 35, no. 1, pp. 121-124, 2015.

[12] G. Jemilda, S. Baulkani, "Moving object detection and tracking using genetic algorithm enabled extreme learning machine," *International Journal of Computers Communications & Control*, vol. 13, no. 2, pp. 162-174, 2018.

[13] M. R. Kaloop, D. Kumar, P. Samui, A. Gabr, J. W. Hu, X. Jin, B. Roy, "Particle swarm optimization algorithm-extreme learning machine (PSO-ELM) model for predicting resilient modulus of stabilized aggregate bases," *Applied Sciences*, vol. 9, no. 16, pp. 1-13, 2019.

[14] W. Lei, H. Wen, J. Wu, W. Hou, "MADDPG-based security situational awareness for smart grid with intelligent edge," *Applied Sciences*, vol. 11, no. 7, pp. 1-19, 2021.

[15] Q. Ma, C. Sun, B. Cui, X. Jin, "A novel model for anomaly detection in network traffic based on kernel support vector machine," *Computers & Security*, vol. 104, no. 2, pp. 1-14, 2021.

[16] N. Moustafa, J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Military Communications and Information Systems Conference (MilCIS'15)*, pp. 1-6, 2015.

[17] F. Panahi, M. Ehteram, A. N. Ahmed, Y. F. Huang, A. Mosavi, A. El-Shafie, "Streamflow prediction with

large climate indices using several hybrid multilayer perceptrons and copula Bayesian model averaging," *Ecological Indicators*, vol. 133, pp. 1-26, 2021.

[18] A. Pradeepika, R. Sabitha, "Examination of diabetes mellitus for early forecast using decision tree classifier and an innovative dependent feature vector based naive bayes classifier," *ECS Transactions*, vol. 107, no. 1, pp. 12937-12952, 2022.

[19] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *The Second IEEE International Conference on Computational Intelligence for Security and Defense Applications*, pp. 53-58, 2009.

[20] M. K. Wali, R. A. Fayadh, N. K. Al_Shamaa, "Electroencephalogram based stress detection using extreme learning machine," *Nano Biomedicine and Engineering*, vol. 14, no. 3, pp. 208-215, 2022.

[21] C. Wannaboon, W. San-Um, "Digital chaotic signal generator using robust chaos in compound sinusoidal maps," *IEICE Transactions on Fundamentals of Electronics Communications & Computer Sciences*, vol. E97.A, no. 3, pp. 781-783, 2014.

[22] T. Xu, G. Coco, M. Neale, "A predictive model of recreational water quality based on adaptive synthetic sampling algorithms and machine learning," *Water Research*, vol. 177, pp. 1-11, 2020.

[23] X. Zhang, R. Li, B. Zhang, Y. Yang, J. Guo, X. Ji, "An instance-based learning recommendation algorithm of imbalance handling methods," *Applied Mathematics and Computation*, vol. 351, pp. 204-218, 2019.

Biography

Li Tian, born in June 1987, has obtained a master's degree in June 2014. He is a senior engineer. He is currently works at State Grid Hubei Electric Power Research Institute. He is interested in network security and power industry control security.

Blockchain Collaborative Coin Mixing Scheme Based on Hierarchical Mechanism

Yan Yan, Qing Liu, and Jingjing Li
(Corresponding author: Qing Liu)

School of Computer and Communication, Lanzhou University of Technology
No.287 Langongping Road, Qilihe District, Lanzhou 730050, China
Email: 212081203023@lut.edu.cn

(Received Apr. 14, 2023; Revised and Accepted Sept. 22, 2023; First Online Apr. 25, 2024)

Abstract

The coin mixing mechanism is one of the technologies to realize the privacy protection of blockchain transaction information. Aiming at the problems of low efficiency and insufficient security of the current coin mixing mechanisms, this paper proposes a blockchain collaborative coin mixing mechanism based on a hierarchical mechanism called GroupShuffle. The mechanism first randomly allocates participants of the coin-mixing transaction into three large groups of comparable size. Each large group is further randomly separated into several small groups with similar numbers of participants. The layered encryption method is adopted within each small group to transfer the output addresses and realize the first layer of shuffling. Then, the last node of each small group represents the group to conduct the second layer of shuffling. Finally, the last node of each large group completes the transfer and shuffling of all output addresses using the layered encryption. Security analysis and experiments show that when the participants of the GroupShuffle mechanism are 27 and 108, the mixing time only takes 8 seconds and 38 seconds. Compared to other existing mixing mechanisms, the proposed GroupShuffle mechanism greatly improves the mixing efficiency, shortens the transaction time, and ensures security during the mixing transaction process.

Keywords: Blockchain; Coin Mixing Mechanism; Hierarchical Mechanism; Privacy Protection

1 Introduction

The blockchain-based distributed ledger integrates asymmetric encryption system, P2P network, consensus algorithm, smart contract and other technologies, which can ensure the consistency and immutability of transaction records [24]. However, the participats of the transaction in the blockchain system need to verify their transaction data when reaching a consensus mechanism. Therefore, the transaction data of blockchain is open and transparent. Take Bitcoin as an example, although the pseudonym mechanism of anonymous identity authentication is adopted, the analysis of sensitive information such as the amount and address in the public ledger may still threaten the transaction identity privacy of the participants and result in the leakage of transaction privacy [10].

Coin mixing mechanism is one of the important technologies to solve the privacy protection problem of blockchain transactions [9, 11, 15]. It improves the privacy and anonymity of cryptocurrency by cutting off the connection between sender and receiver in currency transactions, making it difficult to track the ownership and purpose of currency without changing the trading results. The existing coin mixing mechanisms can be classified into two categories [8]. The first one is the centralized coin mixing mechanism represented by MixCoin [4] and Blindcoin [19]. Users have to send their transactions to a third-party service provider and complete the coin mixing transaction intensively. Although this kind of solution is easy to implement, it cannot guarantee the credibility of the third-party service provider, and there is still a risk of privacy leakage. Once the data have been revealed from the third-party service provider, it is easy to expose the transaction privacy information of all the participats. Another kind of solution is the decentralized coin mixing mechanism represented by CoinJoin [13] and CoinShuffle [16]. Although it overcomes the defects of third-party service provider, this kind of scheme requires all the participates to conduct transaction online at the same time. Besides, the coin mixing progress takes long time and it is vulnerable to the denial of service attacks [2].

Therefore, this paper proposes a blockchain collaborative coin mixing scheme based on hierarchical mechanism, called GroupShuffle. All the participants of the coin mixing transaction are randomly grouped to form a hierarchical structure with basically equal number of members both in the large group and in the small group. The layered encryption method is used to transmit the output addresses of small group members and large group members. The privacy of participants in the coin mixing transaction is protected through the hierarchical shuffling,

while saving transaction time and improving coin mixing efficiency.

2 Related Work

MixCoin [4] and BlindCoin [19] are the typical representatives of the centralized coin mixing technology. MixCoin achieves privacy protection by hiding the relationship between transaction input addresses and output addresses. However, the above operations are performed centrally by a third-party coin mixing service provider. Therefore, the mapping relationship between the input addresses and output addresses of all the participants in this method is visible to the third-party, and cannot be prevented from obtaining the transaction privacy. The BlindCoin scheme proposed by Valenta [19] *et al.* used blind signature technology to blind the output address to ensure that the third-party coin mixing service providers realizes the unlinkability of the internal address while providing the coin mixing service. The third-party coin mixing service providers cannot understand the specific process of the user's coin mixing, so it can prevent it from leaking the user's coin mixing transaction information. The disadvantage of this solution is that it increases the system overhead and the time overhead of the coin mixing process. Literature [17] proposed a blind signature coin mixing scheme based on Elliptic Curves, which also ensures internal anonymity and improves computational efficiency to a certain extent. In the anonymous digital currency DASH [25], a node can apply to become a coin mixing node, but a certain deposit must be paid to the system to obtain the right to mix coin. If the node conducts illegal operations, it will lose a high guarantee fee. The establishment of guarantee fees increases the cost of illegal operation of coin mixing nodes and reduces the possibility of nodes doing evil.

The decentralized coin mixing scheme does not require a trusted third-party service provider, it only requires the interactions between all the participants. Maxwell [13] took the lead in proposing a decentralized coin mixing scheme, called CoinJoin protocol. This scheme utilizes the feature that each transaction in Bitcoin can have multiple input and output addresses to merge multiple transactions, making it difficult to distinguish which input corresponds to which address. However, users who participate in coin-mixing in the CoinJoin scheme may discover other users' information during the interaction with other users,since there is no central node and coin mixing fees, the scheme is vulnerable to DoS attacks and Sybil attacks, and it cannot guarantee internal unlinkability. Ruffing [16] and others proposed a decentralized output address shuffling mechanism, called CoinShuffle, which is based on CoinJoin, can complete the decentralized mixing process and ensure that any user cannot know the information of other users. However, the efficiency of this scheme is low, the coin mixing process requires all participating nodes to be online, and it is vulnerable to denial of service attack. The degree of anonymity of this scheme is related to the number of users participating in coin mixing within a certain period of time, and the anonymity is weak. The TTShuffle mechanism proposed in literature [5] groups the participants of the coin mixing transaction on the basis of the CoinShuffle mechanism, improving the single point of failure problem faced by the CoinShuffle coin mixing mechanism. However, when the number of coin mixing participants increases significantly, the operational efficiency of this mechanism will drop significantly. The IMShuffle mechanism proposed in literature [18] realizes the efficient transfer of output addresses through randomly grouping and selection of intermediaries, which shortens the running time of coin mixing transaction. However, it may lead to the problem that the enthusiasm of the last node of each group to conduct coin mixing transactions will be reduced. Literature [3] proposed a decentralized coin mixing protocol based on blockchain advertisements to anonymously search for coin mixing participants, called Xim. As the number of users participating in coin mixing increases, the cost of the attack will also increase linearly, thereby effectively avoiding denial of service attacks. The CoinParty mechanism proposed in initerature [26] uses secure multi-party computing to simulate a trusted third party. Even in the case of malicious operation or failure of some nodes involved in mixing coins, the coin mixing process is still valid.

Literature [12] proposed a blockchain system model in which privacy protection and supervision coexist, which protects the privacy of receiving addresses while providing a supervision mechanism, and designs group confidential transactions to obscure payment addresses and hide transaction digital assets. Literature [1] used aggregated signatures to create coin mixing transaction, and participants control a jointly created aggregated address to complete the shuffling operation. Literature [21] proposed a coin mixing scheme with a decentralized signature protocol, which avoids third-party by designing a specific mixing user negotiation process. A distribution method for collecting private keys is introduced, reducing the risk of privacy leaks during cryptocurrency mixing. Literature [6] proposed a consortium blockchain double privacy protection method based on group signature and homomorphic encryption, which realizes the privacy protection of the identity of the payer and the transaction amount under the premise of satisfying the traceability and verifiability of the transaction. Literature [20] used a one-way aggregation signature scheme to aggregate the input transaction signatures of all transactions, and combined the idea of currency mixing with encrypted transactions to realize a fully anonymous block that protects the privacy of the identity of payee and payer and transaction amount blockchain system. Literature [14] proposed a decentralized coin mixing scheme with a customizable mixed amount. Users do not need to be online all the time, but only need to submit the mixed funds and the output address encrypted by layers to the coin mixing server group. The coin mixing server group supervises each other but

cannot know the details of the coin mixing, which ensures the security and privacy of the coin mixing. Literature [22] proposed a certificateless signcryption mechanism based on blockchain, which can make good use of the nontamperable feature of blockchain, prevent illegal users from substituting public key of the user, and guarantee signature non-repudiation. Literature [23] used a kind of vague set to improve DPoS consensus mechanism, which can allow each node to vote for the agent nod to improve the security and fairness of blockchain. Literature [7] placed Cryptography data and operations such as user key and generated user address, as well as sensitive privacy information processing in blockchain transaction process in SGX security zone to protect data privacy.

3 Improve Mixing Efficiency via Hierarchical Mechanism

3.1 Mixing Efficiency of CoinShuffle Mechanism

The CoinShuffle mechanism adopts the layered encryption scheme [16] to hide the output transaction addresses. The first participant uses other's public keys to create a layered encryption of his own output address. The next user receives this message and uses his own private key to decrypt the first layer of the encrypted message. Then, he creates a layered encryption of his output address with the remaining public keys of other users, shuffles it with the decrypted message of the previous user, and sent them to the next user. Analogously, the layered encryption and shuffling operation will be carried out by other participant until the proxy node receives and decrypts to get the output addresses of all the users. During the abovementioned transmission and shuffling process, none of the participants can acquire the transaction relationship between others, which makes the input and output transaction addresses of coin mixing users internally unlinkable.

Suppose the total number of participants in the coin mixing transaction with the CoinShuffle mechanism is $N, (N \geq 3)$. Participant 1 creates a layered encryption C_1=Enc(ek_2,Enc(ek_3,...,Enc(ek_N,$vk_1^{'}$))) with his output address $vk_1^{'}$ and public key $\{ek_2,ek_3,...,ek_N\}$ of others, and sends it to participant 2. In the above process, participant 1 needs to perform $(N-1)$ times of encryption operation. Upon receiving C_1, participant 2 decrypts the message with his private key dk_2 and encrypts his Bitcoin output address $vk_2^{'}$ with the public keys of the remaining $(N-2)$ participants to obtain C_2=Enc(ek_3,Enc(ek_4,...,Enc(ek_N,$vk_2^{'}$))). Then, participant 2 adds C_2 to the vector of decrypted messages and shuffles the extended vector randomly to obtain a new vector C_2. During this process, participant 2 completes $(N-2)$ times of encryption operation, 1 decryption operation, and 1 shuffling operation. Analogously, when the penultimate participant receives the encrypted information sent by the previous participant, he needs to perform $(N-2)$ times of decryption operation, 1 encryption operation, and 1 shuffling operation. For the proxy node, after receiving the encrypted information from the penultimate participant, he has to conduct $(N-1)$ times of decryption operation to obtain the output addresses $\{vk_1^{'},vk_2^{'},...,vk_{N-1}^{'}\}$ of other users except himself. Subsequently, the proxy node adds his own output address $vk_N^{'}$ and shuffles all the output addresses to obtain the final output address list. Table 1 summarizes the number of encryption, decryption, and shuffling operations required in the CoinShuffle mechanism with N participants. The

Table 1: The number of operations for CoinShuffle

Node ID	Encryption	Decryption	Shuffling
1	N-1	0	0
2	N-2	1	1
3	N-3	2	1
.....	.....	.....	.....
N-1	1	N-2	1
N	0	N-1	1
Total	$\frac{N(N-1)}{2}$	$\frac{N(N-1)}{2}$	N-1

overall time consumption of the coin mixing transaction is not only associated with the number of encryption, decryption, and shuffling operations, but also related to the specific execution time of these operations. Suppose the execution times of encryption, decryption, and shuffling are t_{en}, t_{de}, and t_{sh}, respectively. Then, the overall time consumption of the CoinShuffle mechanism with N participants can be expressed as:

$$T_C = \frac{N(N-1)}{2} \times t_{en} + \frac{N(N-1)}{2} \times t_{de} + (N-1) \times t_{sh} \quad (1)$$

3.2 Mixing Efficiency of Hierarchical Mechanism

The CoinShuffle mechanism requires all the participants to be online at the same time, and the later participant can only start to operate after the previous one completes the shuffling and address transmission. Therefore, the coin mixing process of CoinShuffle mechanism is lengthy and inefficient. If a malicious node deliberately delays or destroys the coin mixing process, all the completed operations of other users will be meaningless.

In order to improve the operation efficiency of the CoinShuffle mechanism and solve the problem of single point of failure, all of the participants in the coin mixing transaction can be organized into multiple groups and complete the coin mixing operation parallel. Spouse the total number of participants in the coin mixing transaction is N, they are divided into m groups, and each group has n members, that is, $N = m \times n$. When using this kind of hierarchical mechanism, the users within the same group perform layered encryption and shuffling operation on their output addresses according to the CoinShuffle

mechanism. All the groups complete the above operations in parallel, which can be regarded as the first layer of shuffling. Subsequently, the last node of each group, as the representative of the group, performs layered encryption and shuffling again, which can be regarded as the second layer of shuffling. The hierarchical mechanism greatly reduces the number of nodes participate in the shuffling on each layer. On the one hand, it speeds up the efficiency of single layer of shuffling, and on the other hand, it also reduces the impact of single point of failure on the overall coin mixing efficiency.

According to the analysis in Section 3.1, the time to complete the first layer of shuffling in a group with n participants can be expressed as:

$$T_1 = \frac{n\,(n-1)}{2} \times t_{en} + \frac{n\,(n-1)}{2} \times t_{de} + (n-1) \times t_{sh} \quad (2)$$

The time to finish the second layer of shuffling among m groups can be expressed as:

$$\begin{aligned} T_2 \quad = \quad & \frac{m\,(m-1)}{2} \times t_{en} + \frac{m\,(m-1)}{2} \times t_{de} \\ & +(m-1) \times t_{sh} \end{aligned} \quad (3)$$

Therefore, the total time to complete the coin mixing transaction after using the hierarchical mechanism can be expressed by the sum of the time for the two layers of shuffling:

$$T_H = T_1 + T_2 \quad (4)$$

Taking Formula (2) and Formula (3) into the above equation and reorganize the result, we can obtain:

$$\begin{aligned} T_H = & \left[\frac{(m+n)^2 - (m+n) - 2mn}{2}\right] \times t_{en} \\ & + \left[\frac{(m+n)^2 - (m+n) - 2mn}{2}\right] \times t_{de} \\ & + (m+n-2) \times t_{sh} \end{aligned} \quad (5)$$

It can be observed from Formula (5) that the total time of hierarchical mechanism is related to the additive and multiply result of m and n. The total number of participants is $N = m \times n$. According to the constraint relationship between the addition and multiplication results of two variables depicted in Figure 1, the following conclusions can be obtained:

1) When the difference between m and n is significant, the value of N is small and the overall running time of coin mixing is longer;

2) When the difference between m and n is small, the value of N is large and the overall running time of coin mixing is shorter;

3) When m is equal to n, N will achieve its maximum value and the overall running time of coin mixing is the shortest.

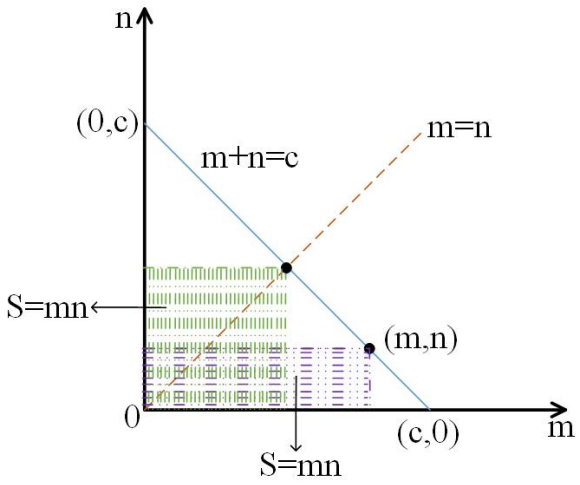


Figure 1: Addition and multiplication relationship of two variables

Based on the above analysis, the collaborative coin mixing scheme proposed in this paper mainly adopts the following hierarchical strategies.

When the total number of participants (represented by N) is relatively small, in order to ensure the security of coin mixing transaction, the number of members (represented by n) in each small group should be as large as possible, while the number of small groups (represented by m) can be small. When the total number of participants is relatively large, the number of small groups and the number of members in each group should be as similar as possible to achieve better coin mixing efficiency.

4 GroupShuffle

The proposed blockchain collaborative coin mixing scheme based on the hierarchical mechanism (i.e., GroupShuffle) mainly includes four stages: announcement stage, shuffling stage, confirmation stage, and blame stage. All participants of the coin mixing transaction will negotiate and agree on the amount of each transaction in advance, and determine the Bitcoin address held by the participant as the input address of the coin mixing transaction.

4.1 Announcement Stage

At this stage, all the participants of coin mixing transaction are first randomly organized into three large groups G_i, $i \in \{1,2,3\}$ with approximately the same number of members. The participants in each large group are then randomly arranged into several small groups S_j, $j \in \{1,2,3,...,m\}$ with roughly the same number of members. Make sure that each of the participant is aware of his large group number G_i and small group number S_j.

Each of the participant n_i, $i \in \{1,2,3,...,N\}$ announces his input address vk_i, and generates a temporary

key pair (ek_i, dk_i) for the encryption and decryption operation in the shuffling stage. The participant broadcasts ek_i and the signature $\sigma_{i,1} = Sig\,(sk_i, 1, \tau)$ to other nodes within the group, where sk_i is the signing key of node n_i and τ is the session identifier.

For the participant n_j within the same small group, after receiving the broadcast message from n_i, he uses the received public key ek_i to verify the signature $\sigma_{i,1}$ to determine whether n_i is a malicious node. Meanwhile, participant n_j checks that the address vk_i holds enough money and is available to carry out the mixing transaction. If the signature verification fails or the input address vk_i does not have enough available transaction amount, it will enter the Blame stage.

4.2 Shuffling Stage

The proposed GroupShuffle mechanism organizes all the participants into large groups and small groups randomly, so as to perform a hierarchical shuffling operation and speed up the execution efficiency of the coin mixing transaction.

4.2.1 First Layer of Shuffling within Small Group

The first layer of shuffling in the GroupShuffle mechanism is completed in parallel within each small group, where the basic principle is consistent with the CoinShuffle mechanism (as shown in Figure 2). Suppose each small group contains n nodes, starting from the first participant node u_1, layered encryption is performed on its output address $vk_1^{'}$ by obtaining the public keys $\{ek_2, ek_3, ..., ek_n\}$ of other nodes within the same small group to get C_1=Enc(ek_2,Enc(ek_3,...,Enc(ek_n,$vk_1^{'}$))). Then, node u_1 signs with its own private key to get $\sigma_{1,2} = Sig\,(sk_1, (ek_1, 2, \tau))$. Subsequently, node u_1 sends the encrypted output address information C_1 together with the signature $\sigma_{1,2}$ to the second node u_2 in this small group.

Node u_2 receives and verifies the validity of the signature $\sigma_{1,2}$. If the signature verification fails, turns to the Blame stage. Otherwise, node u_2 decrypts the received encrypted message C_1 with its own private key dk_2 and obtains D_1=Enc(ek_3,...,Enc(ek_n,$vk_1^{'}$))). Then, node u_2 also uses the public keys of other nodes in this small group to perform layered encryption on its own output address $vk_2^{'}$ to obtain C_2=Enc(ek_3,Enc(ek_4,...,Enc(ek_n,$vk_2^{'}$))). Next, node u_2 randomly shuffles the decrypted information D_1 and its own encrypted information C_2, and sends the result $Shuffle\,(D_1, C_2)$ together with the signature $\sigma_{2,2} = Sig\,(sk_2, (ek_2, 2, \tau))$ to the next node u_3.

Subsequent node in the same small group receives and verifies the signature sent by the previous node in the same way. Then uses its own private key to decrypt the output address sent by the previous node, uses the public keys of other nodes remaining in this group to perform layered encryption on its own output address, and randomly shuffles these messages to get $Shuffle\,(D_{i-1}, C_i)$. The same operation will be done analogously until the last node in this small group uses its own private key dk_n to decrypt and recover the output addresses $\left\{vk_1^{'}, vk_2^{'}, ..., vk_{n-1}^{'}\right\}$ of other nodes in the same group. Then, he will insert his own output address $vk_n^{'}$ and randomly perturb the output addresses of this small group through a shuffling operation to get the complete output address list out_s. Then, the first layer of shuffling will be ended.

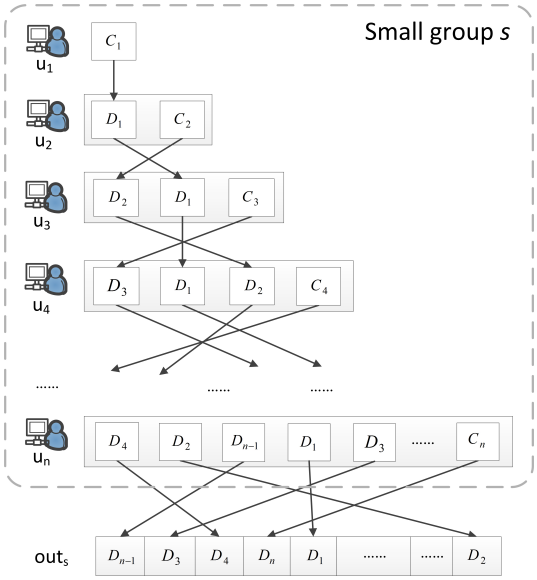


Figure 2: Schematic diagram of the first layer of shuffling within the small group

4.2.2 Second Layer of Shuffling within Large Group

Each large group contains several small groups s_j, $j \in \{1, 2, ..., m\}$, and the last node of each small group will participate in the second layer of shuffling on behalf of the small group. The basic principle is the same as the first layer of shuffling (as shown in Figure 3). The terminal node s_j of each small group announces its own public key ek_{S_j}. Starting from terminal node s_1, it will use the public keys of other terminal node to create layered encryption for the output address out_{s_1} of its small group. The signature $\sigma_{s_1,2} = Sig\,(sk_{s_1}, (ek_{s_1}, 2, \tau))$ together with the layered encryption result will be sent to the terminal node s_2 of the second small group.

The terminal node s_2 of the second small group receives and verifies the legitimacy of the signature $\sigma_{s_1,2}$. If the signature verification fails, turns to the Blame stage. Otherwise, terminal node s_2 decrypts the encrypted message with its own private key, and uses the public keys of the last node of other small groups to perform layered encryption on the output address out_{s_2}. Randomly perturbs the output addresses of two small groups through a shuffling operation $Shuffle\,(out_{S_1}, out_{S_2})$, and then continues to transmit the result to terminal node s_3 of the third small group.

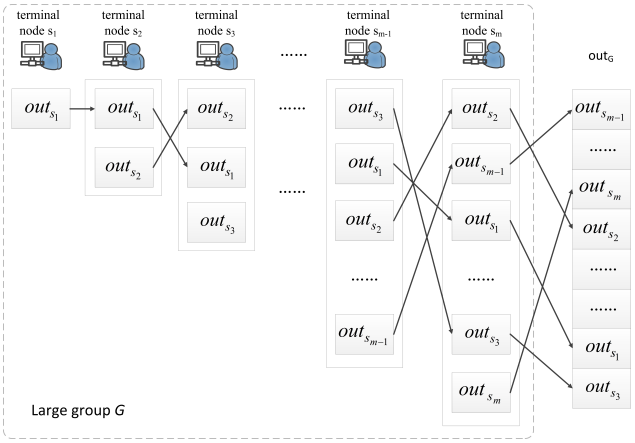


Figure 3: Schematic diagram of the second layer of shuffling within the large group

The last node of each subsequent small group follows the same method to receive and verify the signature sent by the last node of the previous small group. After verifying the legitimacy of signature, the last node of each small group decrypts the message to obtain the output address. Then it will use the public key of the last node of other small groups to perform layered encryption on the output address for its small group, and randomly perturb it with the decrypted output address to form a shuffling result $Shuffle\,(out_{si-1}, out_{s_i})$. The same operation will be done analogously until the terminal node s_m of the last small group decrypts and recovers the output addresses $\{out_{s_1}, out_{s_2}, ..., out_{s_m-1}\}$ of other small groups, then it inserts the output address out_{s_m} of the last small group, and randomly perturb all of the output addresses through a shuffling operation to obtain the complete output address list out_G of the current large group. Then, the second layer of shuffling will be ended.

4.2.3 Third Layer of Shuffling between Large Groups

Finally, the last node of the three large groups will participate in the third layer of shuffling on behalf of their group. The basic principle keeps the same as that of the first two layers of shuffling (as portrayed in Figure 4). The terminal node G_1 of the first large group uses the public keys of the last node of the other two large groups to perform a layered encryption on the output address list out_{G_1} of the first large group. This message will be sent to the terminal node G_2 of the second large group together with the signature $\sigma_{G_1,2} = Sig\,(sk_{G_1}, (ek_{G_1}, 2, \tau))$. Terminal node G_2 receives and verifies the validity of the signature $\sigma_{G_1,2}$. If the signature verification fails, enters the Blame stage. Otherwise, terminal node G_2 uses its own private key to decrypt the message it received, and inserts the encrypted result of output address list out_{G_2} of the second large group to perform a shuffling operation. The result after shuffling will be sent to the terminal node G_3 of the third large group together with the signature $\sigma_{G_2,2} = Sig\,(sk_{G_2}, (ek_{G_2}, 2, \tau))$. Terminal node G_3 receives and verifies the validity of the signature $\sigma_{G_2,2}$. If the verification fails, turns to the Blame stage. Otherwise, terminal node G_3 decrypts the encrypted message, inserts the output address list out_{G_3} of the third large group, and perform shuffling to obtain the output address list $AddressList$ of all the participants. After that, the complete output address list $AddressList$ will be broadcast to all the participants of the coin mixing transaction, and then enters the verification stage.

4.3 Verification Stage

After receiving the output address list $AddressList$, all the participants of the coin mixing transaction will check whether their output address are in the list. If it exists, corresponding participant will use his own private key to sign and broadcast. After each participant receives the output address list confirmed by the signatures of all other participants, the coin mixing transaction takes effect. Then the coin mixing transaction will be submitted to the Blockchain network to complete subsequent operations. If the participant can not find his output address in the list, he will refuse to sign. Then, the blame stage will be started to find the offending node.

4.4 Blame Stage

The main task of the blame stage is to find and eliminate the offending node(s) in the coin mixing transaction process, so as to restart the new coin mixing process. In the proposed GroupShuffle mechanism, participants need to sign for their operations on each stage to ensure that their operations are traceable. When illegal operations are found, the large group where the node is belonging can be quickly located through the signature, and then further traced back to the small group where the node is located, so that the offending node can be quickly found. The following three types of violations are mainly involved:

1) Illegal operations occurred during the Announcement stage. The main task of this stage is that the participants broadcast their input addresses and public keys, and negotiate grouping to clarify the large group and small group they belongs to. Illegal operations at this stage mainly include refusing to broadcast the input address and public key or intentionally publishing wrong address. When there is a violation operation, other honest nodes can find out the violation node by verifying the signature and broadcast it to other nodes. Then, the honest nodes can restart the Announcement process after eliminating the violation node(s).

2) Illegal operations occurred during the Shuffling stage. The main task of this stage is to perform layered encryption and shuffling of the output addresses for all the participants. During the three-layer shuffling

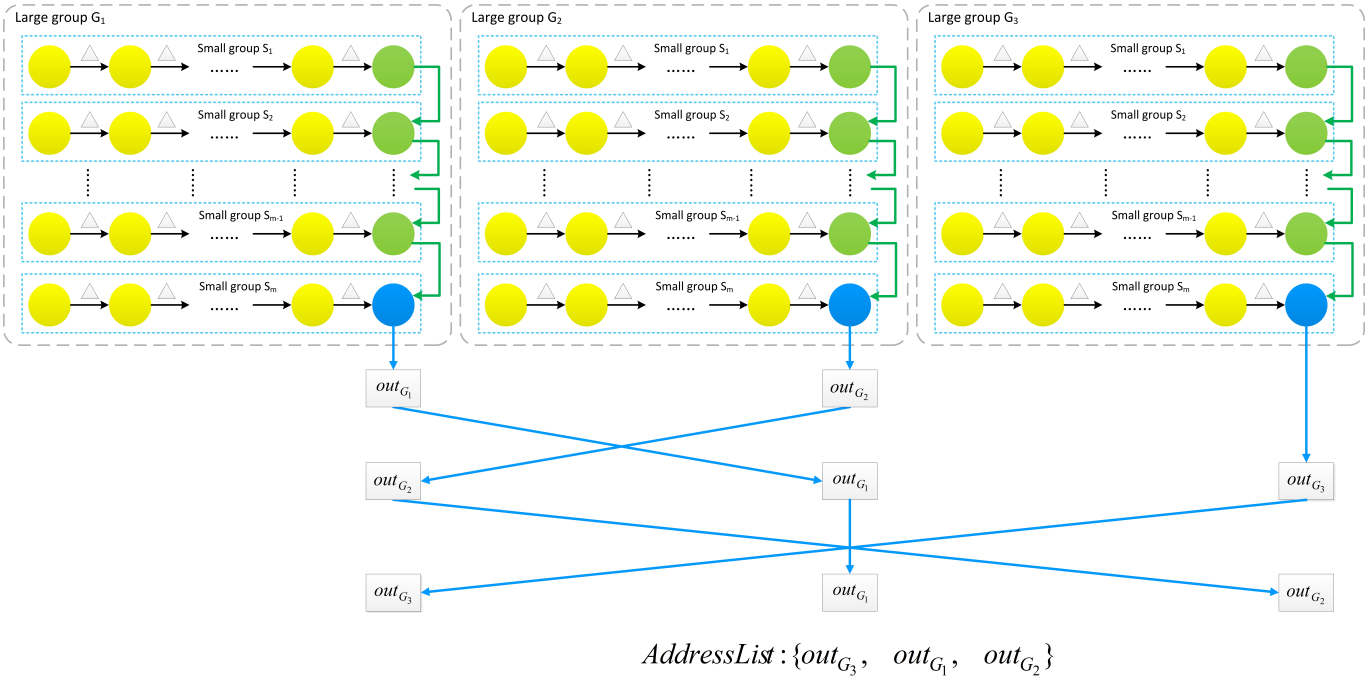


Figure 4: Schematic diagram of the third layer of shuffling between large groups

process of the GroupShuffle mechanism, illegal operations may include output address layered encryption errors, discarding or withholding received address encryption information, etc. When the above-mentioned violations occur, it is necessary to determine the large group that the offending node belongs based on the wrong or missing output address signature. Then, the last node of the large group traces back to the small group where the offending node locates based on the address. Finally, the last node of the small group determines the corresponding offending node by verifying the signature. The offending node(s) will be broadcast to other coin mixing participants, and the honest nodes can restart the shuffling process after removing the offending node(s).

3) Illegal operations occurred during the Confirmation stage. The main task of this stage is to verify the output addresses of all the participants are in the output address list and then to sign and confirm them. Violations at this stage are primarily signature denials. Other participants can use the missing signature to discover the participants who refused to sign, and then remove their corresponding output addresses from the output address list.

5 Experiments and Security Analysis

In order to verify the performance of the GroupShuffle mechanism, this section will compare and analyze the results of the proposed method with some decentralized coin mixing mechanisms, such as CoinShuffle [16], TTShuffle [5], and IMShuffle [18] in terms of the overall running time of the coin mixing mechanism, the average running time of nodes, the impact of the number of groups on running efficiency, and the impact of the number of malicious nodes. Multiple groups of experiments will be Set up to compare and analyze the factors that affect the performance of the coin mixing mechanism, such as the number of participants and the number of malicious nodes.

5.1 Experimental Environment

The hardware platform is the 64-bit windows 10 operating system, 4GB internal storage, Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz processor. A P2P network has been set up to generate multiple participant nodes in the coin mixing transaction. The CBC mode of AES algorithm is used for encryption operations between nodes, and the Elliptic Curve Digital Signature is selected to complete the signature confirmation operation between nodes.

5.2 Analysis of Mixing Efficiency

Suppose that there is no malicious node, set the total number of participants in the coin mixing transaction to be 9, 18, 27...,108. The experiments of the CoinShuffle mechanism, TTShuffle mechanism, IMShuffle mechanism, and GroupShuffle mechanism have been repeated several times to get the averaged overall running time of the mechanism and the running time of a single node. Experimental results are depicted in Figure 5 and Figure 6.

It can be observed from Figure 5 that the overall run-

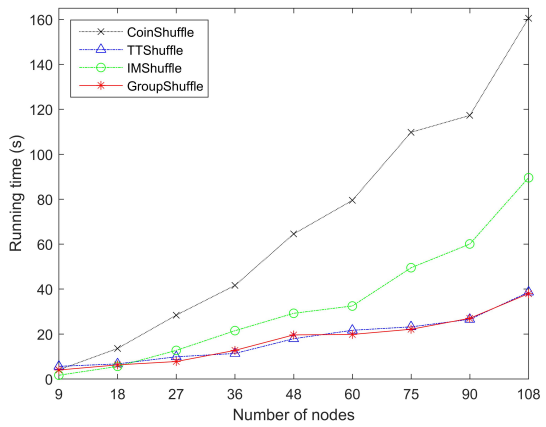


Figure 5: Running time of different mixcoin mechanisms

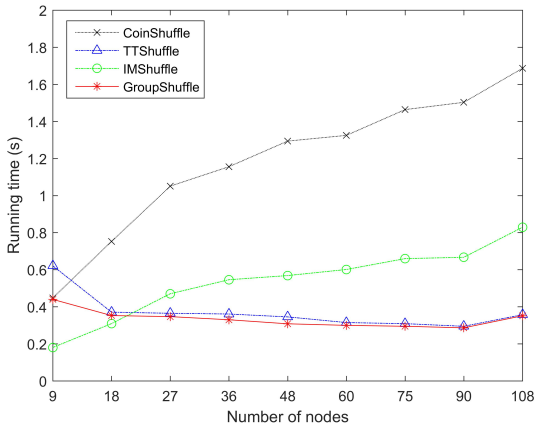


Figure 6: Average running time of node

ning time of various coin mixing mechanisms increases to varying degrees with the increase in the number of coin mixing transaction participants. The CoinShuffle mechanism, IMShuffle mechanism and TTShuffle mechanism take about 28 seconds, 13 seconds and 10 seconds respectively when the total number of participants is 27; while the GroupShuffle mechanism only takes about 8 seconds for the coin mixing process under the same circumstances, which is significantly lower than other mechanisms. When the number of mixing transaction participants increases to 108, the running time of the GroupShuffle mechanism is only 38 seconds, which is far less than 160 seconds of the CoinShuffle mechanism. With the increase of the number of mixing participants, the increase in the running time of the GroupShuffle and TTShuffle mechanisms using the layered mixing strategy is relatively flat, which is significantly smaller than that of other mechanisms. The reason is that the participants in the middle position of the CoinShuffle mechanism need to decrypt the encrypted information sent by the previous participant to obtain the output address of the former from it, and then they encrypt all the information again after adding their own output address and send it to the next participant. As the number of participants increases, this repeated encryption and decryption operations take a lot of time, resulting in lengthy mixing time and low efficiency. The IMShuffle mechanism groups the participants of the mixing transaction, although it can improve the efficiency of the mixing transaction to a certain extent, each user needs to find a "middleman" and transmit its encrypted output address to it. Therefore, the overall operational efficiency is significantly slower when there are a large number of participants. At the same time, the results also prove that the blockchain layered coin mixing scheme designed in this paper can significantly improve the efficiency of coin mixing and save the time of the coin mixing transaction.

According to the result of the average running time of node portrayed in Figure 6, as the number of mixed coin participants increases, the average running time of CoinShuffle mechanism nodes increases significantly, and the growth rate of the IMShuffle mechanism is second. Because as the number of nodes participating in mixing transaction increases, each node in the above mechanism needs to perform output address encryption and decryption operations more times. The average running time of the node of the GroupShuffle and TTShuffle mechanisms using the layered coin mixing strategy is relatively stable, and the average running time of the node of the GroupShuffle mechanism is significantly lower than other mechanisms. The reason is that the increase in the number of participating nodes in the mixing transaction appears as more groups on the first layer, which has no effect on the parallel shuffling speed of each group on the first layer. The newly added groups have increased the number of nodes participating in the second layer shuffling, resulting in an insignificant increase in the running time of the overall coin mixing mechanism. Based on the above results, it can be seen that the GroupShuffle mechanism proposed in this paper is more efficient in mixing coin when there are no malicious nodes.

5.3 Quantity of Members vs. Mixing Efficiency

In order to further discuss the impact of the number of groups and the number of members in the group on the efficiency of coin mixing while using a hierarchical coin mixing mechanism, the total number of participants in the coin mixing transaction will be fixed at 90 and the members of each group are set to be 3, 5, 6, 10 and 15, under the assumption that there is no malicious node. The experiments of the CoinShuffle mechanism, TTShuffle mechanism, IMShuffle mechanism, and GroupShuffle mechanism have been repeated several times to get the averaged overall running time of the mechanism and the running time of a single node. Wherein, the number of groups formed by TTShuffle and IMShuffle mechanism are 30, 18, 15, 9 and 6 respectively. The number of groups formed by GroupShuffle mechanism are 10, 6, 5, 3 and 2

respectively. Experimental results are portrayed in Figure 7 and Figure 8.

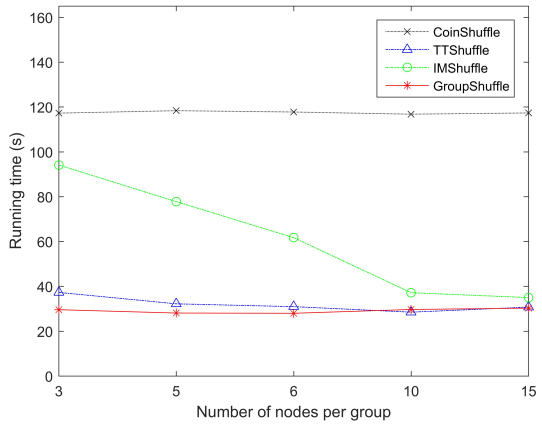


Figure 7: Comparison of running time with 90 participants

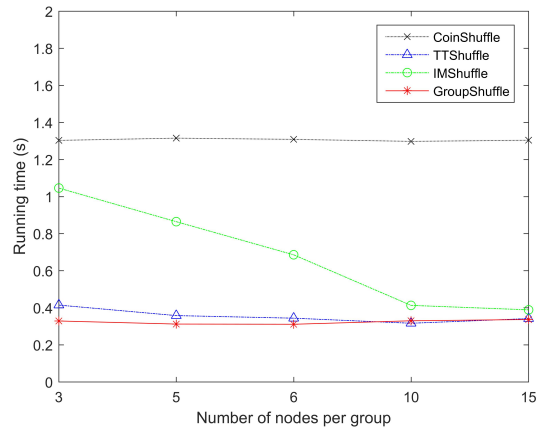


Figure 8: Average running time of node with 90 participants

Compare the results in Figure 7 and Figure 8 it can be observed that, the overall running time of the CoinShuffle mechanism and the running time of a single node have nothing to do with the number of groups and the number of members in each group. Because the CoinShuffle mechanism does not involve grouping, all nodes participating in coin shuffling will layer encryption their own output addresses according to the rules and send them to the next participant. Therefore, when the total number of participating nodes remains unchanged, the overall running time of the CoinShuffle mechanism and the running time of a single node are almost a horizontal straight line. With the increase of the number of members in the IMShuffle mechanism, the overall time required to complete coin mixing and the running time of a single node are gradually reduced, and the efficiency of coin mixing is gradually improved. Because under the premise that the total number of participating nodes is fixed, the increase in the number of members in the small group leads to a reduction in the number of small groups, and the time consumed to find a "middleman" to forward the output address is also reduced. Therefore, the overall time spent on coin mixing and the time spent on a single node are reduced accordingly. In the GroupShuffle and TTShuffle mechanisms that adopt a hierarchical coin mixing strategy, the greater the difference between the number of members in the small group and the number of small groups, the longer the overall running time and the running time of a single node to complete the mixing; When the number of members in small group and the number of small groups are roughly equal, the overall running time to complete the coin mixing and the running time of a single node are shorter. Among them, the running time of the GroupShuffle mechanism proposed in this paper is shorter than other coin mixing mechanisms. Based on the above results, it can be seen that the stratification and grouping strategy of the GroupShuffle mechanism can achieve better coin mixing efficiency when there are no malicious nodes among the coin mixing participants.

5.4 Proportion of Malicious Nodes vs. Mixing Efficiency

Denial of service attack is a common problem for the coin mixing transaction. Malicious participants can destroy the realization of the entire mixed coin transaction by sending false transaction addresses, wrong encryption, signature results or refusing to complete the relevant operations of each stage. In order to analyze the influence of the number of malicious nodes on the efficiency of coin mixing transaction, the total number of participants is fixed at 90, and 10%, 20%, 30%, 40%, and 50% of the participants are randomly selected as malicious nodes. The experiments of the CoinShuffle mechanism, TTShuffle mechanism, IMShuffle mechanism, and GroupShuffle mechanism have been repeated several times to compare the running time of various mechanisms with different proportion of malicious nodes. Experimental results are portrayed in Figure 9.

Compare the results of Figure 9 with Figure 5, it can be found that when there are malicious nodes among the participants in the coin mixing transaction, the running time of the coin mixing mechanism increases significantly. When the total number of participating nodes remains unchanged, as the proportion of malicious nodes increases, the running time of various coin mixing mechanisms increases significantly. This is because after a malicious node is discovered, the coin mixing mechanism needs to remove the malicious node and restart execution from the Announcement stage, so the overall running time is significantly increased. The GroupShuffle mechanism proposed in this paper takes less time to complete coin mixing than other mechanisms when the proportion of malicious nodes continues to increase, which shows that the two-layer grouping coin mixing mechanism designed in this paper can improve the efficiency of coin mixing and

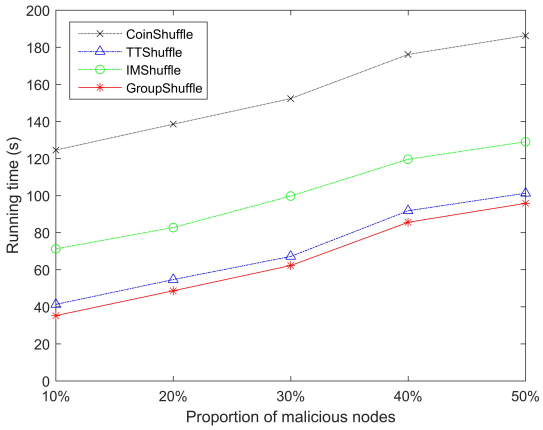


Figure 9: The impact of proportion of malicious nodes on running time

greatly reduce the risk of encountering denial of service attacks during the transaction process. The reason is that no matter whether the malicious node is a member of a certain group or the last node, the denial of service attack initiated by it will only affect the mixing process of the group, and will not affect the mixing process of other groups. Therefore, the scope of impact of the denial of service attack on the overall coin mixing efficiency is limited to the first layer of shuffling stage, which will not cause the second and third layer of shuffling speed to decline. Through the layering and grouping mechanism, the denial of service attack of a malicious node in a group will not affect the coin mixing process of other groups, thereby reducing the risk of denial of service attacks for other coin mixing transaction participants, and at the same time improving the coin mixing process overall efficiency.

5.5 Security Analysis

First of all, the proposed GroupShuffle mechanism can protect the corresponding relationship between each participant and its own output address, ensuring the transaction privacy of the coin mixing participants. The GroupShuffle mechanism uses the CBC mode of the AES algorithm for encrypted transmission between nodes. Each node uses the public key of other nodes in the group to perform layering encryption on its own output address. In addition to the output address, it avoids the leakage of other personal information. When the first layer of shuffling is performed in the small group, the next node can only use its own key to decrypt the outermost information of the message sent by the previous node, until the last node in the small group decrypts to obtain all the output address information of the small group. However, each node performs a shuffling operation while adding its own output address, disrupting the order of the addresses, so that the last node cannot match the output address information of the participating nodes in the group with their identities one by one, cutting off the linkability between the input address and the output address. The subsequent second layer of shuffling between the last nodes of each small group and the third layer of shuffling between the last nodes of the large group will randomly disrupt the relationship between the small group and the large group through layering encryption and shuffling. The corresponding relationship between the output addresses further ensures the transaction privacy security of the coin mixing participants.

Second, the proposed GroupShuffle mechanism reduces the impact of denial of service attack. Denial of service attack may occur at all stages of the coin mixing transaction, the later the stage, the longer the wasted time and the greater the impact on the coin mixing transaction. Taking the CoinShuffle mechanism as an example, if a malicious node refuses to confirm that its own output address exists in the output address list at the final stage, the coin mixing mechanism needs to start from scratch after identifying and eliminating the malicious node, which wastes a lot of time, and damages the user experience of participation. The GroupShuffle mechanism reduces the risk of denial of service attack during the transaction process through layering and grouping. Denial of service attack by malicious nodes in one group will not affect the coin mixing process of other groups. The first layer of shuffling in the small group can detect and filter out malicious nodes in time when decrypting and confirming output addresses to reduce the impact on coin mixing results of other participants. It is foreseeable that when the total number of coin mixing participants increases significantly, the layering and grouping strategy of the GroupShuffle mechanism will have more obvious advantages in resisting denial of service attack and execution efficiency.

6 Conclusion

The decentralized coin mixing mechanism based on CoinShuffle uses layered encryption to complete the packaging and transmission of the output addresses, which requires a large amount of calculation and the efficiency of low. In order to solve the above problems, this paper proposes a blockchain collaborative coin mixing mechanism based on hierarchical mechanism (GroupShuffle). The Participants of coin mixing transaction are randomly organized to form a hierarchical structure with comparable size of large group and small group. The privacy of coin mixing participants is protected by the layered shuffling within the small group, with the large group, and between large groups. Meanwhile, the efficiency of the overall coin mixing mechanism has been improved. The experimental comparison results prove that regardless of whether there are malicious nodes among the participants, the coin mixing efficiency of the proposed GroupShuffle mechanism is better than that of the existing CoinShuffle, TTShuffle, and IMShuffle mechanisms. Limited by the level of knowledge, the proposed GroupShuffle mechanism also has some shortcomings. For example, the last node of

each small group and the last node of each large group need to undertake more cumbersome computing tasks than other participants in the second and third layer of shuffling, and pay more computing power resources. Therefore, it may reduce the enthusiasm of the corresponding nodes to participate in the coin mixing transaction, and even refuse to become the last node, resulting in the restart of the coin mixing process. Therefore, in the subsequent improvement work, corresponding incentive schemes will be considered to increase the enthusiasm of participating nodes and improve the integrity of the GroupShuffle mechanism.

Acknowledgments

This study is supported by the National Nature Science Foundation of China (No. 62361036), and the Nature Science Foundation of Gansu Province (No. 22JR5RA279).

References

[1] F. Barbàra and C. Schifanella, "Dmix: decentralized mixer for unlinkability," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pp. 1–8. IEEE, 2020.

[2] R. Beck, M. Avital, M. Rossi, and J. B. Thatcher. "Blockchain technology in business and information systems research,", 2017.

[3] G. Bissias, A. P. Ozisik, B. N. Levine, and M. Liberatore, "Sybil-resistant mixing for bitcoin," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pp. 149–158, 2014.

[4] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18*, pp. 486–504. Springer, 2014.

[5] Q. L. Cheng and Y. Jin, "Ttshuffle: privacy protection mechanism based on two-tier shuffling in blockchain," *Application Research of Computers*, vol. 38, no. 2, pp. 363–371, 2021.

[6] Y. Q. Diao, A. Y. Ye, J. M. Zhang, H. N. Deng, Q. Zhang, and B. R. Cheng, "A dual privacy protection method based on group signature and homomorphic encryption for alliance blockchain," *Journal of Computer Research and Development*, vol. 59, no. 1, p. 172, 2022.

[7] J. S. Fan, J. H. Chen, R. Shen, Z. G. Liu, Q. M. He, and B. T. Huang, "Sgx based privacy and security protection method for blockchain transactions," *Journal of Applied science*, vol. 39, no. 1, pp. 17–28, 2021.

[8] Q. Feng, D. B. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, 2019.

[9] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 169–178, 2009.

[10] J. Guan, "A review of research on the application fields and existing problems of blockchain technology," *Technology Innovation and Application*, vol. 11, no. 12, pp. 134–136+139, 2021.

[11] M. C. K. Khalilov and A. Levi, "A survey on anonymity and privacy in bitcoin-like digital cash systems," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2543–2585, 2018.

[12] C. P. Li, L. E. Wang, Q. T. Xu, D. C. Li, P. Liu, and X. X. Li, "Groupchain: A blockchain model with privacy-preservation and supervision," in *Proceedings of the 2020 4th International Conference on High Performance Compilation, Computing and Communications*, pp. 42–49, 2020.

[13] G. Maxwell, "Coinjoin: Bitcoin privacy for the real world," in *Post on Bitcoin forum*, vol. 3, p. 110, 2013.

[14] M. H. Nie and Y. Y. Ou, "A digital currency decentralized obfuscation scheme with customizable amounts," *Journal of Guangdong University of Technology*, vol. 38, no. 01, pp. 64–68, 2020.

[15] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7*, pp. 552–565. Springer, 2001.

[16] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "Coinshuffle: Practical decentralized coin mixing for bitcoin," in *Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II 19*, pp. 345–364. Springer, 2014.

[17] T. Shen, C. Qing, and J. P. Yu, "A blind-mixing scheme for bitcoin based on an elliptic curve cryptography blind digital signature algorithm," *arXiv preprint arXiv:1510.05833*, 2015.

[18] J. H. Song, Z. K. Li, and B. C. Zhang, "Coin mixing mechanism in blockchain based on intermediator," *Application Research of Computers*, vol. 39, no. 3, pp. 868–873, 2022.

[19] L. Valenta and B. Rowan, "Blindcoin: Blinded, accountable mixes for bitcoin," in *Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers*, pp. 112–126. Springer, 2015.

[20] Z. Y. Wang, J. W. Liu, Z. Y. Zhang, and H. Yu, "Full anonymous blockchain based on aggregate signature and confidential transaction," *Journal of Computer Research and Development*, vol. 55, no. 10, pp. 2185–2198, 2018.

[21] R. Y. Xiao, W. Ren, T. Q. Zhu, and K. R. Choo, "A mixing scheme using a decentralized signature protocol for privacy protection in bitcoin blockchain," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1793–1803, 2019.

[22] G. X. Xu, J. N. Dong, C. Ma, J. Liu, and U. G. O. Cliff, "A certificateless signcryption mechanism based on blockchain for edge computing," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[23] G. X. Xu, Y. Liu, and P. W. Khan, "Improvement of the dpos consensus mechanism in blockchain based on vague sets," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4252–4259, 2020.

[24] A. Zhang and X. Y. Bai, "Survey of research and practices on blockchain privacy protection," *Journal of Software*, vol. 31, no. 5, pp. 1406–1434, 2020.

[25] L. H. Zhu, H. DONG, and M. SHEN, "Privacy protection mechanism for blockchain transaction data," *Big Data Research*, vol. 4, no. 1, pp. 46–56, 2018.

[26] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, and k. Wehrle, "Coinparty: Secure multi-party mixing of bitcoins," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, pp. 75–86, 2015.

Biography

Yan Yan received the Ph.D. degree in control theory and control engineering from Lanzhou University of Technology, China. She is currently an Associate Professor at School of Computer and Communication, Lanzhou University of Technology. Her research interests include information security, privacy preserving technology, and differential privacy. She is a member of the IEEE and the China Computer Federation.

Qing Liu is currently a master student of the School of Computer and Communication, Lanzhou University of Technology, China. She received the B.Eng. degree from Longqiao College of Lanzhou University of Finance and Economics in 2020. Her research interests include blockchain security and privacy protection.

Jingjing Li is currently a master student of the School of Computer and Communication, Lanzhou University of Technology, China. She received the B.Eng. degree from Gansu University of Political Science and Law in 2020. Her research interests include blockchain security and privacy protection.

Image Encryption Scheme for Deniable Authentication Based on Chaos Theory

Qiu-Yu Zhang, Yi-Lin Liu, and Guo-Rui Wu
(Corresponding author: Qiu-yu Zhang)

School of Computer and communication, Lanzhou University of Technology
No. 287, Lan-Gong-Ping Road, Lanzhou 730050, China
Email: zhangqy@lut.edu.cn, lyilin1024@163.com, vigdis_r@163.com
(Received May 8, 2023; Revised and Accepted Sept. 24, 2023; First Online Apr. 25, 2024)

Abstract

Aiming at the complex structure of the existing deniable authentication image encryption methods based on public key cryptography and the high computational cost caused by many bilinear and modular power operations, an image encryption scheme for deniable authentication based on chaos theory was proposed. Firstly, the data owner uses the pseudo-random number generator based on the key encryption technology of Caesar's password to generate the session key and conduct a secure exchange between the data owner and the data user by generating a hash code. Then, the improved Arnold chaotic hashing parallel algorithm is used to encrypt and decrypt the original image iteratively, and the corresponding hash values are generated during the encryption and decryption process. Finally, the data user judges the hash value to achieve deniable authentication. Experimental results show that the proposed scheme simplifies the design of the repudiation authentication cryptographic protocols and ensures the privacy and security of data user identities. The improved Arnold chaotic hashing parallel algorithm is used to improve image encryption algorithms' security and computational efficiency.

Keywords: Chaotic Hashing Parallel Algorithm; Deniable Authentication; Image Encryption; Key Negotiation; Privacy Protection

1 Introduction

With the rapid development of the Internet and cloud computing technology, the secure sharing of image data can further exploit the high value of image data while protecting the privacy of users [23]. The privacy protection issue has been hindering the development of image data sharing, making it urgent to solve the problem of anonymization in secure image sharing [26]. Therefore, a new cryptographic mechanism, deniable authentication protocol, has been developed to meet some new security requirements in Internet applications.

Deniable authentication protocols are a unique style of contemporary cryptographic authentication protocols [1, 3]. A deniable authentication protocol has three basic features: the receiver can verify a given message at any time; the receiver cannot prove to a third party that the message originated from a particular sender; and the sender can deny the origin of a message if the receiver intends to reveal the source to a third party. This feature of deniable authentication makes it widely used in areas such as secure email, medical research, and electronic voting [14]. In recent years, scholars have introduced public key cryptography into the field of deniable authentication in order to improve the security strength of deniable authentication, which is a new deniable authentication encryption (DAE) system with the security properties of both encryption and deniable authentication [4].

DAE is a cryptographic primitive that can simultaneously perform public key encryption and deniable authentication at a lower cost than the deniable method of authentication followed by encryption, enabling deniable authentication and confidentiality at the same time [16], while deniable encryption enables the confidentiality of communication messages to be guaranteed even if the sender/receiver is later forced to reveal the plaintext, random number or private key [17]. There are many DAE techniques available, but most are based on public key cryptosystems [4,18], with a large number of bilinear pairings, modulo power operations, and complex structures and operations [8, 13]. The key to achieving DAE functionality lies in ensuring that the receiver can identify the source of a particular message it receives on the premise that somehow the receiver of the message is able to mimic all the communication processes between the sender and the receiver. Therefore, there is a need for a shared key between the sender and the receiver along with the security of the key, as well as a good performance encryption and decryption algorithm and a hashing algorithm to encrypt and authenticate the message.

In this paper, a relatively simple structure, relatively low computational cost and secure deniable authenticated

image encryption scheme is constructed by combining a secure Caesar cipher-based key encryption technique with an improved Arnold chaotic hashing parallel algorithm. The main contributions are as follows:

1) By using a key encryption technique based on the Caesar cipher, a hash code for the session key is generated using a hash code generator, which is then transformed into the corresponding session key using a key generator after passing through a secure transmission channel, ensuring the secure exchange of keys between the data owner and the data user.

2) A chaotic hashing parallel algorithm is proposed using a group cipher algorithm and an improved Arnold chaotic mapping. The improved Arnold chaotic mapping solves the problems of simple low-dimensional chaotic structure, insufficient key space and short period, and has better chaotic performance, chaotic range and complexity, which improves the security performance of image encryption.

3) A simple structured DAE scheme is designed by combining the Caesar cipher-based key encryption technique and the chaotic hashing parallel algorithm. Due to the parallelism of the chaotic hashing parallel algorithm, the computational efficiency is improved while achieving user identity privacy protection.

The rest of the paper is organized as follows: Section 2 reviews related research work. Section 3 gives details of the implementation principles and processing flow of the proposed scheme. Section 4 gives experimental simulations as well as a detailed security performance analysis and a comparative performance analysis with existing methods. Section 5 concludes the work in this paper.

2 Related Works

In recent years, cryptographic researchers have proposed a number of novel and efficient deniable authentication protocols. The earliest deniable authentication protocol was proposed by Dwork *et al.* in 1998 based on zero-knowledge proofs, but the proof of knowledge in this protocol is very time-consuming and the scheme is time-limited. Shi *et al.* [22] proposed a quantum deniable authentication protocol based on the continuous quantum theory of bimodal compressed quantum states. Li *et al.* [20] proposed two HDA protocols in a pervasive computing environment based on bilinear pairs. Ahene *et al.* proposed two HDA schemes, the HDA-I scheme allowing communication sessions from CLC to Identity-Based Cryptography (IBC) settings, and the HDA-II scheme operating in reverse. Xu *et al.* [25] propose a recipient-repudiable steganography scheme that uses deep neural networks to handle the coercive attacks encountered by the receiver. Abd *et al.* [1] design a non-interactive secure deniable authentication protocol whose core security is based on factorization and the discrete logarithm difficulty problem. Gupta *et al.* [11] propose an identity-based deniable authentication protocol that is protocol is efficient enough to resist many possible attacks and can be used in resource-limited mobile Ad Hoc network environments. Zeng *et al.* [28] propose a deniable authentication protocol with source hiding that does not reveal any private information and protects the privacy of participants.

Traditional encryption schemes do not consider the case where an adversary eavesdrops on the ciphertext and then coerces the sender or receiver to account for the public key, random number, plaintext or decryption key used in encryption, so Canetti *et al.* introduced the concept of Deniable Encryption (DE) in 1997, which allows the sender/receiver to generate a "deniable" (but indistinguishable from the real value) random number/private key to open the ciphertext as a different plaintext even after encrypting the communication. Kar *et al.* [19] propose an efficient and lightweight DAE scheme that avoids heavy pairing operations and enables lower computational cost, communication overhead and energy consumption. Jiang *et al.* [12] designed a new message reconciliation system to support deniable and responsible message reconciliation to avoid abusive reporting. Jin *et al.* [15] used a hybrid encryption approach to design an HDAE scheme by combining a tag key encapsulation mechanism with a data encapsulation mechanism. Yang *et al.* [17] proposed a specific and efficient DE scheme based on single-key function encryption, focusing on the design of a specific and efficient dual-solution DE scheme in a private key scenario. Chen *et al.* [7] designed a cross-layer DE system that is reasonably applicable to mobile devices. Existing DE schemes cannot support mutual authentication. Cao *et al.* [4] propose an efficient public key authentication DE scheme that supports both deniability and mutual authentication, while protecting the receiver's identity privacy under coercion. Agrawal *et al.* [2] define and construct a Learning With Errors (LWE) polynomial based difficulty. Kar *et al.* [18] designed a DAE scheme based on a certificate-free setting. Jin *et al.* [13] proposed a DAE scheme based on bilinear pairing operations for email. Chi *et al.* [8] proposed a DAE scheme based on the LWE problem by increasing the probability of decryption failure so that the user can deceive the coercer with a false message. Lin *et al.* [21] proposed a protection method for Learning Encryption with Deniability (LED), which forces third parties to believe in false data and protects user privacy. Kang *et al.* [17] proposed a key editing scheme that the receiver can deny, reducing the length of ciphertext and key, and simplifying the construction of decryption algorithms. Chakraborty *et al.* [6] proposed a public key encryption scheme with multiple specified receiver signatures that provides stronger deniability. Collins *et al.* [9] constructed a deniable authentication model that considers the real world of the entire message system. Xu *et al.* [24] proposed a sender-repudiation image hiding scheme based on reversible networks and a sender-repudiation image steganography scheme based

on deniable encryption. Cao *et al.* [5] constructed a new public-key sender DE scheme in a fully deniable framework. Cui *et al.* [10] proposed a chaotic mapping-based full-session key negotiation scheme applied to in-vehicle Ad hoc networks, using a chaotic mapping The DAE is implemented using a combination of chaotic mapping and hashing function encryption algorithms, which has better security and improved computational performance.

In summary, most of the existing DAE techniques are based on public-key cryptosystems, which have complex structures and high computational costs. Therefore, the deniability, authentication and confidentiality of the image encryption scheme are achieved by using the key encryption technique based on Caesar cipher and chaotic hashing parallel algorithm in this paper. Meanwhile, the design of the cryptographic protocol is simplified and the computational cost is reduced.

3 The Proposed Scheme

3.1 DAE Model

Figure 1 shows the DAE model used. The model consists of four different entities: Trusted Third Party (TA), Data Owner (DO), Data User (DU) and Coercer (CO).

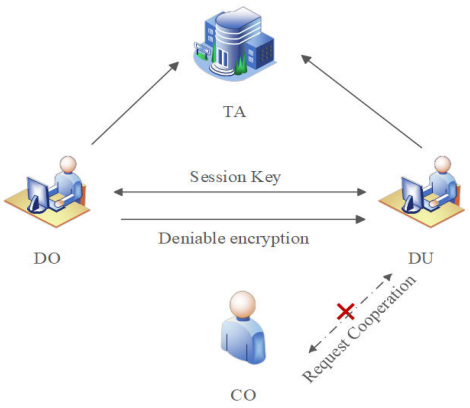


Figure 1: DAE model.

1) Trusted Third Party (TA): The TA is assumed to be a powerful trusted third party with a high level of communication, computational and storage capabilities.

2) Data Owners (DO): Generate session keys, encrypt the raw image data and transmit them to the data user.

3) Data users (DU): Decryption of ciphertext image data and the ability to identify the source of the ciphertext data, but not to prove its true origin to a third party.

4) Coercer (CO): The CO forces the DU to obtain the source of the message. However the DU is unable to confirm the true source of the data as it can completely falsify the entire communication process.

3.2 Key Encryption Processing Based on Caesar's Cipher

Encrypting the session key k using a Caesar cipher-based key encryption method enables double-secure encryption of the session key, which guarantees the secure transmission of k and allows the sender and receiver to have a common session key. The sender of this method transmits the hash code instead of the session key, and the hash code can be transformed into the key corresponding to the hash code (session key k) for the receiver to decrypt the message. Figure 2 shows the encryption process for a session key.

As shown in Figure 2, the session key encryption process consists of five main components: a pseudo-random number generator, a session key, a hash code generator, a hash code, and a key generator.

1) Pseudo-random number generator: used to generate the session key and guarantee the randomness of the session key.

2) Session key: used by the encryption algorithm to convert plaintext data into ciphertext data, while passing through the hash generator to generate the hash code.

3) Hash code generator: used to generate the hash code for the session key.

4) Key generator: used to generate the session key corresponding to the transmitted hash code.

5) Hash Code: is an integer value associated with a session key. The ¡*hash code*, *session key length*¿ is transmitted over a secure transmission channel and a key generator on the receiving end is used to generate the session key corresponding to the hash code.

The pseudo-random number generator uses the generated session key in the image encryption algorithm, while the sender uses the hash generator to generate a hash code corresponding to the session key to be sent to the receiver via the transmission channel, the hash code generation algorithm is implemented as shown in Algorithm 1.

Algorithm 1 Hash code generation

1: **Input:** Unsigned long integer key *key*, Unsigned long integer key length *key_len*
2: **Output:** *hash_value*
3: // Unsigned long integer hash code generation
4: Unsigned long integer hashes $hash\_value = 0, i$
5: **for** $i = 0; i < key\_len; i++$ **do**
6: $hash\_value \leftarrow hash\_value + (\text{pow}(31,\text{i}) \times (key\%10)$
7: $key \leftarrow key/10$;
8: **end for**
9: **return** *hash_value*

After the receiver receives the hash code, it generates the session key k corresponding to the hash code for ciphertext image decryption through the key generator, and

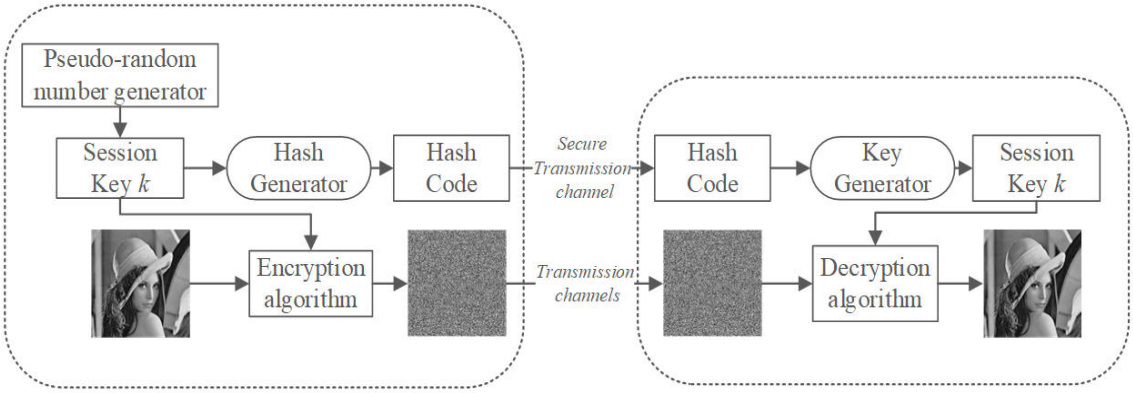


Figure 2: Session key encryption.

the key generation algorithm is implemented as shown in Algorithm 2.

Algorithm 2 Key generation

Input: Unsigned long integer hash codes *hash_code*, Unsigned long integer key length *key_len*
Output: *key*
// Unsigned long integer session key decryptor
Unsigned long integer $key = 0, quotient$
// i takes integers
int i;
for $i = key\_len - 1; i >= 0; i++$ **do**
  $quotient \leftarrow hash\_code/(pow(31, i))$
  $key \leftarrow 10 \times key + quotient$;
  $key \leftarrow key/10$;
  $hash\_code \leftarrow hash\_code\%pow(31, i)$
end for
return *key*

3.3 Deniable Authentication Image Encryption Algorithm

3.3.1 Deniable Authentication Image Encryption

Figure 3 shows the proposed deniable authenticated image encryption processing flow. Defining k as the key, m as the plaintext, C as the round function and E as the chaotic hashing parallel algorithm, the encryption is $E_k(m)$ and the decryption is $D_k(E_k(m))$. The whole encryption and decryption process is essentially a specific number of iterations, where the encryption of the plaintext image is done in parallel with the hashing of the plaintext, and the decryption process also generates the corresponding hash values.

The specific encryption processing steps are as follows:

Step 1: DO generates the session key using the pseudo-random number generator, and then generates the hash value through the hash code generator. The hash value is transmitted to DU through the secure transmission channel, and DU generates the corresponding session key using the key generator of Section 3.2 to realize the secure exchange of the session key.

Step 2: DO utilizes the chaotic hashing parallel algorithm, which generates the sub key, message block, and position corresponding to the session key through the sub key generation algorithm. The input value of the round function C is generated through the $f_k()$ function.

Step 3: According to the chaotic hashing parallel algorithm, the ciphertext $E_k(m)$ is generated by iterating through the grouping cipher algorithm and the round function, and the hash value $H(m)$ of m can be obtained simultaneously by the parallelism of algorithm E. $E_k()$ and $H(m)$ will be sent to DU at the same time.

Step 4: DU receives $E_k(m)$ and $H(m)$ and performs the decryption calculation $D_k(E_k(m)) = m'$, while a new hash value $H(m')$ can be obtained due to the parallelism of algorithm E.

Step 5: Compare $H(m)$ with the received $H(m')$. If $H(m) = H(m')$, the message is accepted, otherwise it is rejected.

3.3.2 The Chaotic Hashing Parallel Algorithm

This paper uses the hashing parallel algorithm based on block cipher algorithm and improved Arnold chaotic map to encrypt the plaintext image. Among them, the parallel mode is reflected in two aspects: first, the plaintext image is processed in parallel into message blocks $M_1, M_2, ..., M_s$ using a structure similar to DM(DM is a DM database, which is a partitioned logical storage structure), and then processed in parallel through $f_k()$; Secondly, through the tree structure, two adjacent blocks Out_i and Out_{i+1} are combined into one, and the hash value output by the round function is combined through the round function C to generate the final hash value. The specific design details are as follows:

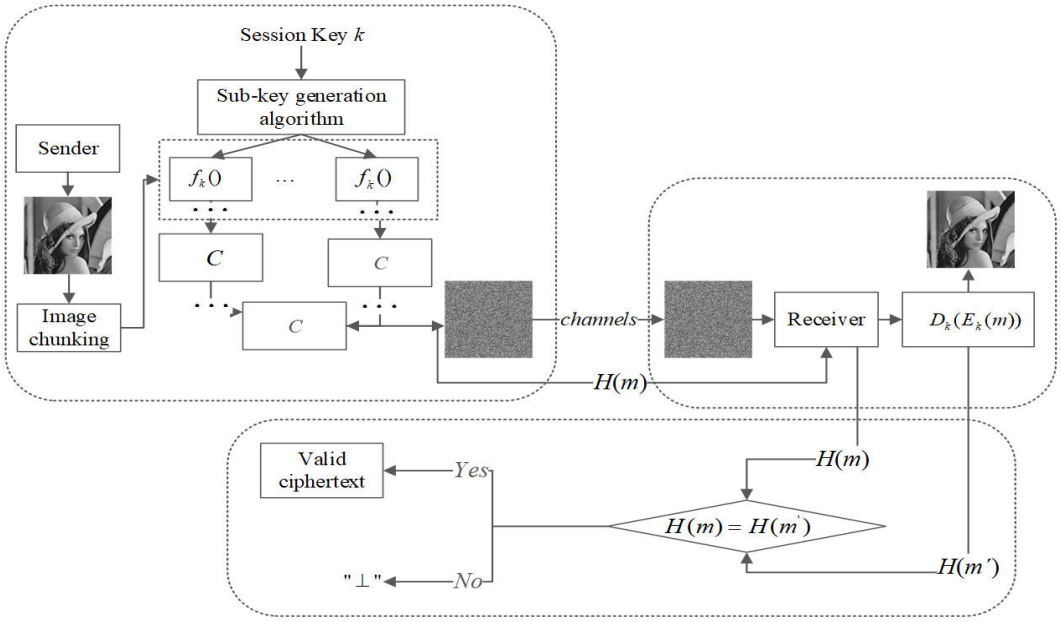


Figure 3: Deniable Authentication Image Encryption Process.

1) Block cipher algorithm: Block cipher has simple structure and strong applicability. Using the iteration of the round function to achieve "diffusion" and "disturbance" of data, providing data confidentiality and facilitating parallel operations. Its encryption E and decryption D can be represented as a mapping on F_2^n : S_k is the key space, and $S_k \subseteq F_2^n$. The detailed definition is shown in Eq. (1):

$$E : F_2^n \times S_k \to F_2^n \quad D : F_2^n \times S_k \to F_2^n \tag{1}$$

where, for any key $k \in S_k$, its encryption function $E(\cdot, k)$ and decryption function $D(\cdot, k)$ are permutations on F_2^n , and for the same key k, the encryption and decryption functions are inverse permutations with each other. It can be seen that the plaintext length and ciphertext length in Eq. (1) is both n.

2) Improved Arnold mapping: The parameters in the traditional two-dimensional Arnold model shown in Eq. (2) is fixed, and there are problems with low dimensional chaotic structures, insufficient key space, short cycles, and vulnerabilities in grayscale and color image encryption. After multiple rounds of transformation, there will be periodic phenomena.

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = A_j \begin{bmatrix} x_n \\ y_n \end{bmatrix} (mod\ L) \tag{2}$$

where $\begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix}$, p and q are the control parameters of the chaotic system, and once they are assigned, they are determined. When $p = q = 1$, the system is a classic Arnold transform, which is equivalent to image cutting and stitching. (x_n, y_n) and (x_{n+1}, y_{n+1}) are the positional coordinates of pixels before and after conversion.

In order to overcome the problem of small key space and short period, the parameters of the traditional two-dimensional Arnold model of Eq. (2) has been improved and are mathematically defined in Eq. (3):

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = A_j' \begin{bmatrix} x_n \\ y_n \end{bmatrix} (mod\ L) \tag{3}$$

where $A_j' = \begin{bmatrix} 1 & x(p) \\ x(q) & x(p) * x(q) + 1 \end{bmatrix}$, and is represented by the following matrix of parameters.

$$A_0' = \begin{bmatrix} 1 & x(p) \\ x(q) & x(p) * x(q) + 1 \end{bmatrix},$$

$$A_1' = \begin{bmatrix} x(p) * x(q) + 1 & x(p) \\ x(q) & 1 \end{bmatrix},$$

$$A_2' = \begin{bmatrix} 1 & x(q) \\ x(p) & x(p) * x(q) + 1 \end{bmatrix},$$

$$A_3' = \begin{bmatrix} x(p) * x(q) + 1 & x(q) \\ x(p) & 1 \end{bmatrix},$$

$$A_4' = \begin{bmatrix} x(p) & 1 \\ x(p) * x(q) - 1 & x(q) \end{bmatrix},$$

$$A_5' = \begin{bmatrix} x(q) & 1 \\ x(p) * x(q) - 1 & x(p) \end{bmatrix},$$

$$A_6' = \begin{bmatrix} x(p) & x(p) * x(q) - 1 \\ 1 & x(q) \end{bmatrix},$$

$$A_7^{'} = \begin{bmatrix} x(q) & x(p) * x(q) - 1 \\ 1 & x(p) \end{bmatrix}.$$

The two pixels $x(p)$ and $x(q)$ are two pixel values from plaintext. The positions of two pixels are relatively fixed in clear text. However, after each round of perturbation, the encrypted object will change, and the pixel values at p and q will also change. So $x(p)$ and $x(q)$ will also change. This results in a rapid increase in key space and period, ultimately achieving the effect of One Time One Secret (OTP).

3) Key hashing function: Hashing function: Hashing function refers to mapping an input of any size to a hash value of a fixed size, mathematically defined as Eq. (4):

$$H_i = h(M_i, H_{i-1}) \tag{4}$$

where $h()$ and $M_i(i$=1, 2,...,$n)$ represent the hashing loop function and the i-th message block, respectively. H_{i-1} and H_i are the previous and current state values of the hashing function, respectively.

According to Eq. (4), the definition of Eq. (5) can be obtained.

$$\begin{aligned} H_n &= h(M_n, H_{n-1}) \\ &= h(M_n, h(M_{n-1}, M_{n-2}) \\ &= \cdots \\ &= h(M_n, h(M_{n-1}, h(M_{n-2}, \cdots, h(M_1, M_0)))) \end{aligned} \tag{5}$$

Key hashing function: The key hashing function $H_n = h_k(M_n, H_{n-1})$ can be executed in parallel if the loop function h_k satisfies the following conditions.

$$h_k(x, y) = f_k(x) \odot g_k(y) \tag{6}$$

$$g_k(x \odot y) = g_k(x) \odot g_k(y) \tag{7}$$

where k is the key, $f_k()$ and $g_k()$ are two keying functions, and $\odot$ is an operator.

The corresponding parallel hash value H_n is calculated by Eq. (8):

$$\begin{aligned} H_n = {} & f_k(M_n) \odot g_k(f_k(M_{n-1})) \odot \cdots \odot g_k^i(f(M_{n-i})) \\ & \odot \cdots \odot g_k^{n-1}(f_k(M_1)) \odot g_k^n(H_0) \end{aligned} \tag{8}$$

4) Chaotic hashing parallel algorithm structure: Figure 4 shows the parallel structure of the designed chaotic cryptographic hashing function. The hashing function has strong diffusion and obfuscation properties and is very sensitive to messages and has good resistance to statistical attacks. The specific steps are as follows:

Structure of $f_k()$: The return value of $f_k()$ in Figure 4 should be determined not only by the key and the message block, but also by the position of the block in the whole message. $f_k()$ consists of a grouping cipher and a logical operation, and the return value depends on the key k, the message block M_i and the position corresponding to the message block, as defined in Eq. (9):

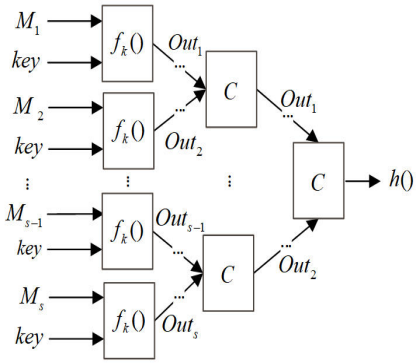


Figure 4: Chaotic cryptographic hashing function parallel structure.

$$\begin{cases} K_i^{'} = E_{M_i}(i \oplus k) \oplus i \oplus k \\ M_i^{'} = E_i(M_i \oplus k) \oplus M_i \oplus k \\ Out_i = f_k(M_i) = E_{M_i^{'}}(K_i^{'}) \oplus K_i^{'} \end{cases} \tag{9}$$

where the parameters M_i, i and k represent the ith message block, the position of M_i in the whole message and the key, respectively. Out_i is the output value of $f_k(M_i)$. The function $E_k(p)$ is a packet cipher, and the output is a ciphertext p encrypted with the key k, $\oplus$ denotes a heterogeneous operation, where the function $E_k(p)$ uses the Advanced Encryption Standard (AES).

Step 1: Define l as the length of the message block, which is set to 128 bits. The padding message M ensures that its length is a multiple of l; the number of padding bits $(100...0)_2$ is n, satisfying $(m+n)\ mod\ l = l - 64$, $1 \le n \le l$. In addition, the last 64 bits of the padding part are used to save the length of M. If M is greater than 2^{64}, let $m = m \bmod 2^{64}$.

Step 2: The padding message is divided into blocks $M_1, M_2, ..., M_s$. Each block contains l bits.

Step 3: Calculate all $Out_i(i = 1, 2, ..., s)$ values in parallel according to Eq. (9).Combine all Out_i by tree structure to obtain the final hash value. Algorithm 3 is a chaotic hashing parallel algorithm implementation.

Round function $C(a, b)$: Due to the low security of simple operators such as XOR and multiplication, chaotic mapping has the advantages of sensitivity to initial conditions, chaos and ergodicity, and its iteration has good one-way. Define a one-way round function $C(a, b)$ using an improved discrete Arnold chaotic map, where a, b are input values. Function $C(a, b)$ compresses two message blocks into one,

Algorithm 3 Chaotic hashing parallel algorithm

Input: $Out_1, Out_2, \cdots, Out_s$
Output: Out_1
$R = [log_2 S]$
$count = S$;
for i=1 to R **do**
  $k = \lfloor count/2 \rfloor$
  for j=1 to k **do**
    $Out_j = C(Out_{2j-1}, Out_{2j})$
  end for
  if $2k < count$ **then**
    $Out_{k+1} = Out_{count}$
  end if
  $count = k$
end for
return Out_1

which has good diffusivity, compressibility and unidirectionality. Algorithm 4 is implemented as a round function generation algorithm.

Algorithm 4 $C(a, b)$ generation algorithm

Input: two message block a **and** b
Output: Out
The message blocks and are partitioned into 8-bit sub-blocks $a_1, a_2, \cdots, a_t$ **and** $b_1, b_2, \cdots, b_t$ **respectively**
$x^{'} = a_1 \oplus a_2 \oplus \cdots \oplus a_t$
$y^{'} = b_1 \oplus b_2 \oplus \cdots \oplus b_t$
for m=1 to n **do**
  for i=1 to t **do**
    $p = x$
    $q = y$
    $x = (a_i + \lfloor x^{'}/37 \rfloor)\ mod\ 256$
    $y = (b_i + \lfloor y^{'}/37 \rfloor)\ mod\ 256$
    if $x = 0$ and $y = 0$ **then**
      $x = 1$
      $y = (a_i + b_i)\ mod\ 256$
    end if
    if $(x + y) > 2$ **then**
      $j = (x + y)\ mod\ 8$
    else
      $j = (x + y)\ mod\ 4$
    end if
    $\begin{bmatrix} x^{'} \\ y^{'} \end{bmatrix} = A_j^{'} \begin{bmatrix} x \\ y \end{bmatrix}\ mod\ 256$
    $a_i = x^{'}$
    $b_i = y^{'}$
  end for
end for
$Out = (a_1 \oplus b_1)||(a_2 \oplus b_2)|| \cdots ||(a_t \oplus b_t)$

4 Experimental Results and Performance Analysis

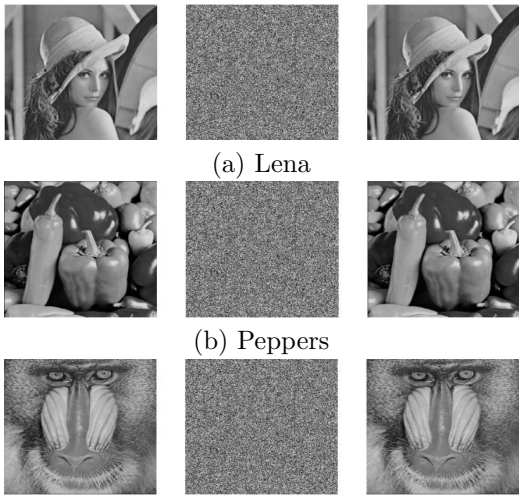
4.1 Image Encryption Performance Analysis

All the experiments in this paper are done in a laptop with Windows 10 operating system, hardware environment of Intel Core i5-4210H 2.9GHz and running memory of 32GB using MATLAB 2015, Visual C++ experimental platform to complete the experimental simulation. Three grayscale images of Lena, Peppers, and Baboon with the size of 256×256 are selected as the test images. Figure 5 shows the test results of the original image, encrypted image, and decrypted image of the three test images. From the test results, we can see that the encrypted ciphertext images do not have any visual information leakage, and the decryption process can highly restore the plaintext images. It shows that the proposed scheme can effectively ensure the visual security of the ciphertext image.

4.1.1 Histogram Analysis

The grayscale histogram shows the distribution of all grayscale values of an image, where the horizontal coordinate indicates the grayscale value and the vertical coordinate indicates the number of times the pixel with each grayscale value appears in the image. Figure 6 shows the histograms corresponding to the plaintext and ciphertext images of the three test images.

From Figure 6, it can be seen that the gray value distribution of the plaintext image histogram is very uneven, while the corresponding gray value distribution of the ciphertext image histogram is more uniform. It indicates that the proposed scheme can well mask the statistical characteristics of grayscale images, and has very good diffusion characteristics and resistance to statistical attacks.

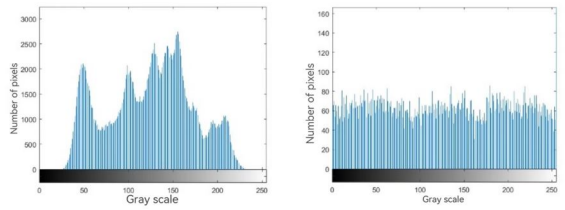


(a) Lena

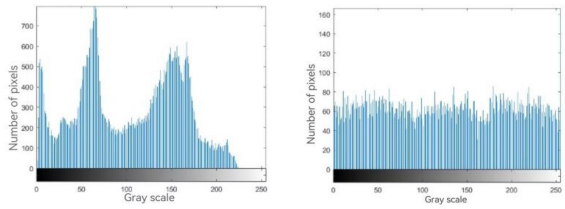
(b) Peppers

(c) Baboon

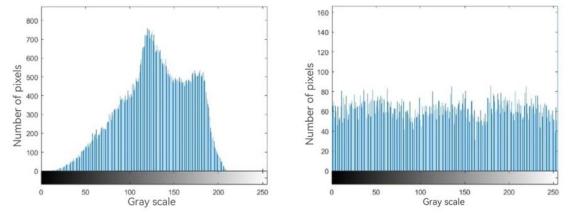
Figure 5: Test results of original image, encrypted image, and decrypted image.



(a) Lena



(b) Peppers



(c) Baboon

Figure 6: Histogram comparison of plaintext and ciphertext images.

4.1.2 Correlation Analysis

Correlation refers to the degree of correlation between adjacent pixel grayscale values in an image. The smaller the correlation coefficient, the lower the pixel correlation, and the stronger the ability to resist statistical analysis. Figure 7 shows the correlation analysis of adjacent pixels between Lena plaintext image and ciphertext image. The definition of correlation coefficient is shown in Eqs. (10)-(13):

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)D(y)}} \tag{10}$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{11}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{12}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{13}$$

where r_{xy} is the correlation coefficient; $cov(x,y)$ is the covariance; $D(x)$ is the variance; $E(x)$ is the mean; x, y are the grayscale values of the neighboring pixels.

As can be seen from Figure 7 that the adjacent pixel correlation between plaintext and ciphertext images is calculated for Lena images using 1500 pairs of pixel points, and the adjacent pixel correlation is stronger for plaintext images and the distribution of adjacent pixels is more uniform for ciphertext images, indicating that the proposed scheme satisfies zero correlation and has better encryption effect, and ciphertext images have lower pixel correlation. Table 1 shows the comparison between the correlation coefficients between adjacent pixels of plaintext and ciphertext images for the proposed scheme and existing methods [27, 29].

As can be seen from Table 1, the correlation coefficients between the adjacent pixels of the ciphertext image and the pixels in the horizontal, vertical and diagonal directions in the original pixel plaintext image are relatively large, and the correlation coefficients are close to 1; while in the corresponding ciphertext image, the correlation coefficients are close to 0. The experimental results are slightly better than those in the scheme [27, 29]. Therefore, the proposed scheme can better eliminate the correlation between adjacent pixels and can mask the original features of the image.

4.1.3 Information Entropy

The information entropy reflects the uncertainty of image information, and generally the higher the entropy, the more information and the less visible information. For a grayscale image with L=256, the theoretical value of information entropy is 8. Table 2 shows the comparison of information entropy between the proposed scheme and existing methods [27,29]. The information entropy is calculated by Eq. (14) as follows:

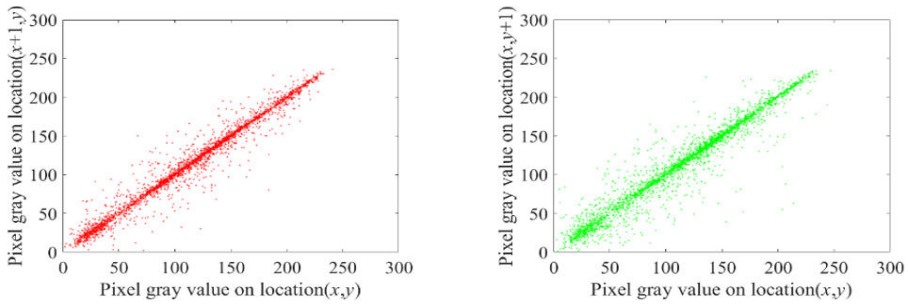
$$H(m) = -\sum_{i=1}^{L} P(m_i) log_2 P(m_i) \tag{14}$$

where L denotes the number of gray levels in the image, (m_i) denotes the pixel value, and $P(m_i)$ denotes the probability of occurrence of the gray value m_i.

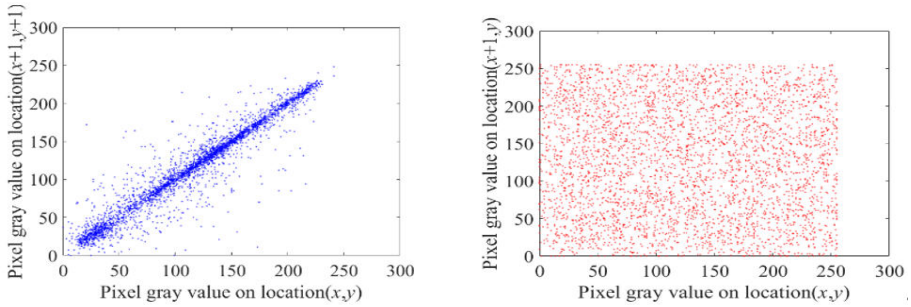
As can be seen from Table 2, the information entropy of the encrypted image is close to the theoretical value8 and its grayscale values are uniformly distributed, which can effectively resist statistical attacks. Compared with the scheme [27, 29], the information entropy value of the ciphertext image in the proposed scheme is closer to the ideal value, and the proposed chaotic hashing parallel algorithm is very sensitive to messages due to its strong diffusion and obfuscation properties. Therefore, the proposed scheme has better performance in resisting statistical attacks.

4.1.4 Differential Attack

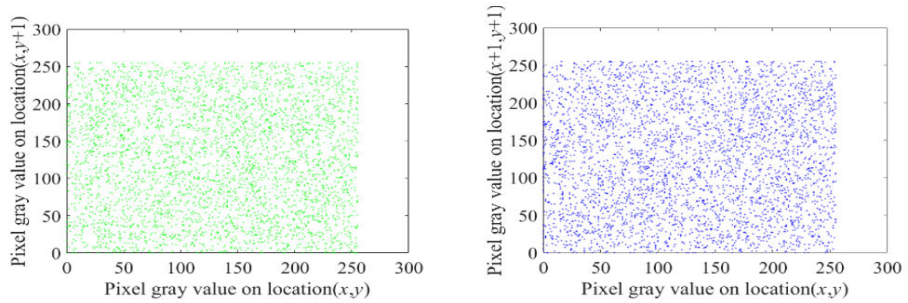
Differential attack refers to an attacker who, after slightly changing the plaintext, compares the differences between the corresponding ciphertext before and after the change, in order to find the corresponding relationship between the plaintext image and the ciphertext image. The number of pixels change rate (NPCR) and normalized average change intensity (UACI) are usually used to evaluate the



(a) Horizontal orientation of plaintext images (b) Vertical orientation of plaintext images



(c) Diagonal orientation of plaintext images (d) Horizontal orientation of the ciphertext image



(e) Vertical orientation of the ciphertext image (f) Diagonal orientation of the ciphertext image

Figure 7: Neighboring pixel correlation analysis of Lena plaintext image and ciphertext image.

algorithm's ability to resist differential attacks. When the values of NPCR and UACI are close to the ideal values of 99.6094% and 33.4635%, it indicates that the encryption has a strong ability to resist differential attacks. Table 3 shows the comparison of NPCR and UACI between our proposed approach and existing methods [27, 29]. The relevant calculation formulas are shown in Eqs. (15)-(17):

$$C(i,j) = \begin{cases} 0, \ if \ P_1(i,j) = P_2(i,j) \\ 1, \ if \ P_1(i,j) \neq P_2(i,j) \end{cases} \tag{15}$$

$$NPCR = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} C(i,j)}{M \times N} \times 100\% \tag{16}$$

$$UACI = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} |P_1(i,j) - P_2(i,j)|}{255 \times M \times N} \times 100\% \tag{17}$$

where M and N represent the length and width of the image, respectively, and $P_1(i,j)$ and $P_2(i,j)$ represent the corresponding ciphertext pixel grayscale values before and after the plaintext change, respectively.

As can be seen from Table 3, the NPCR values of the proposed scheme are slightly lower than those in the scheme [27, 29], and the UACI values are slightly higher than those in the scheme [27] and slightly lower than those in the scheme [27], but the NPCR and UACI values of all images are closer to the ideal values than those in the scheme [27, 29]. Therefore, the proposed scheme has better performance in resisting differential attacks.

4.2 Performance Analysis of Deniable Encryption Schemes

Deniability: Because the data owner and data user already jointly own the session key k, and the chaotic encryption hashing parallel algorithm $E_k()$ is stored in a trusted third party (TA), both parties can obtain the algorithm $E_k()$ through TA. Therefore, data users can completely imitate all the communication processes between the data owner and the data user. It can provide an attacker with a false message that cannot be detected by the attacker, so that the data user cannot prove to the attacker that the message m

Table 1: Correlation coefficients between adjacent pixels

Images	Direction	Plaintext images	Cipher images	Ref. [29]	Ref. [27]
Lena	Level	0.97898	-0.0036822	0.00902660	-0.007100
	Vertical	0.98784	-0.0021478	-0.00592550	0.008500
	Diagonal	0.96547	0.0032566	0.00552270	0.000200
Peppers	Level	0.97979	-0.0028623	0.00088813	0.003800
	Vertical	0.98075	-0.0022578	0.00060857	0.002500
	Diagonal	0.96631	0.0032000	0.00434800	0.003200
Baboon	Level	0.86829	-0.0036002	0.00424650	-0.004800
	Vertical	0.77354	-0.0021458	-0.00740770	-0.001700
	Diagonal	0.74757	0.0037566	-0.00456560	0.006800

Table 2: Comparison of information entropy

Images	Plaintex	Ciphertext	Ref. [29]	Ref. [27]
Lena	7.445121	7.997656	7.9976948	7.996800
Peppers	7.581900	7.997268	7.9971601	7.996800
Baboon	7.593800	7.997153	7.9970808	7.997100

comes from the data owner, and the data owner can also deny that the message m comes from themselves.

Message authenticity: This article proposes to transmit the key by generating a hash code instead of directly transmitting the key, which has good security. Therefore, only the session key k of algorithm E is known between the data owner and the data user. Data users know that only the data owner can generate ciphertext with the key k, which can confirm the true identity of the user from whom the message m originates. At the same time, data users can also confirm the integrity of the message transmission process through hash values and achieve user identity authentication.

Safety analysis: During message transmission, the key is transmitted by generating a hash code, which achieves double-secure encryption of the session key. The security by algorithm E ensures that it is impossible for a malicious attacker to intercept $E_k(m)$ and $H(m)$ and replace them with their own values without knowing the key, i.e. the scheme is resistant to man-in-the-middle attacks, and the use of a chaotic hashing parallel encryption algorithm based on the packet cipher algorithm and the improved Arnold chaos mapping-based chaotic hashing parallel encryption algorithm for encrypting images, making the scheme resistant to statistical attacks and to differential attacks.

Efficiency analysis: The proposed scheme adopts the improved Arnold chaotic cryptographic hashing parallel algorithm to perform encryption, decryption and hashing operations in parallel, which is a significant improvement in efficiency compared to traditional encryption and hashing algorithms, and most of the existing DAE schemes based on public key cryptosystems are based on bilinear pairing and modulo power operations, which are computationally expensive.

4.3 Performance Comparison with Existing DAE Solutions

4.3.1 Efficiency Analysis

Compared to the bilinear mapping in the public key encryption regime and the ECC encryption algorithm, the chaotic encryption algorithm avoids scalar multiplication and modulo power operations, which improve efficiency. is negligible compared to other operations. Thus, a general conclusion is drawn as follows:

$$T_p \approx 10T_m, T_m \approx 3T_c, T_c \approx 2.42T_s, T_s \approx 17.4T_h$$

A summary of these formulas reflects a link between the time taken by the algorithms:

$$T_p \approx 10T_m \approx 30T_c \approx 72.6T_s \approx 1263.24T_h$$

where T_p is the bilinear pairing operation time; Tm is the scalar dot product operation time; T_c is the execution time of $T_n(x)$ mod p in the Chebyshev polynomial; T_s is the symmetric encryption operation time; and T_h is the hashing operation time.

The proposed scheme generates the hash value in two stages. (1) The complexity of the computation is mainly contributed by $f_k()$, which encrypts the message block three times using AES (AES is a simple integer operation). These operations use very few CPU resources during the computation. (2) The computational complexity is mainly contributed by the round function $C()$, and the floating-point operations in the chaotic mapping can be omitted. At the same time, due to the parallelism of the chaotic hashing parallel algorithm, its encryption and decryption as well as the operations on the hash values are executed in parallel. Figure 8 shows the running time of the chaotic hashing parallel algorithm corresponding to varying the file size for different numbers of threads.

As can be seen from Figure 8, the execution time of the entire chaotic hashing function (including the time to read the file) decreases as the number of threads increases.

Table 3: Comparison of NPCR and UACI

Images	NPCR (%)			UACI (%)		
	Proposed	Ref. [29]	Ref. [27]	Proposed	Ref. [29]	Ref. [27]
Lena	99.6135	99.6170	99.6140	33.4384	33.4199	33.5463
Peppers	99.6245	99.6399	99.6262	33.4505	33.3027	33.4768
Baboon	99.6102	99.6185	-	33.4312	33.4211	-

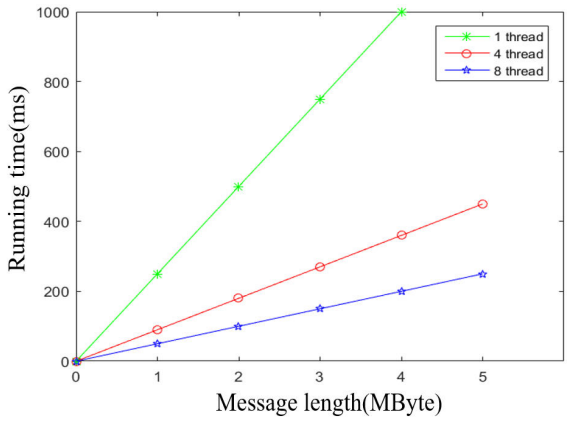


Figure 8: Running time of the chaotic hash parallel algorithm

Therefore, the chaotic hashing function proposed in this paper can achieve higher operational efficiency on a parallel platform and the constructed deniable authenticated image encryption scheme costs less time.

4.3.2 Safety Analysis

Table 4 shows the performance of the proposed scheme compared to existing DAE schemes in terms of security.

It can be seen from Table 4 that the DAE scheme constructed in this paper meets all the security requirements in the table, and compared with the scheme [5], the proposed scheme provides mutual authentication by generating the key into a hash code for transmission, so that only the session key of the algorithm is known between the DO and the DU, and the DU knows that only the DO can generate the ciphertext with the key k, so that it can confirm that the message m comes from the DO. At the same time, compared with the scheme [4] and scheme [5], the proposed scheme is resistant to statistical attacks and differential attacks, and the experimental results are close to the ideal values according to the two metrics of information entropy value and differential attacks in the experimental analysis. Therefore, the proposed scheme has better security performance.

5 Conclusions

A deniable authenticated image encryption scheme based on chaos theory is proposed. The scheme addresses the complexity of existing key exchange protocols and the lack of security in the key transmission process by generating a hash code in the form of a session key for transmission, which achieves double encryption of the session key and completes the secure exchange between the two parties; in addition, using an improved Arnold chaotic hashing parallel algorithm, i.e. encrypting the original image by iteration of the group cipher algorithm and the round function The security of the ciphertext image is guaranteed, as only the session key is common between the data owner and the data user, and the encryption function is known. As a result, the data user can fully simulate the entire communication process, enabling a deniable authenticated image encryption process; the efficiency of the encryption operation is also improved due to the parallelism of the encryption algorithm. The experimental results show that the scheme is highly operable, secure and has low computing costs, and facilitates the privacy protection of user identity through the deniability implementation.

The next research plan is to combine it with related techniques such as searchable encryption, thus enhancing the security performance of the DAE scheme.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61862041). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] N. N. Abd and E. S. Ismail, "Deniable authentication protocol based on factoring and discrete logarithm problems," in *2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pp. 1–6. IEEE, 2022.

[2] S. Agrawal, S. Goldwasser, and S. Mossel, "Deniable fully homomorphic encryption from learning with errors," in *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part II 41*, pp. 641–670. Springer, 2021.

[3] E. Ahene, J. Walker, E. Ahene, J. Walker, I. Ali, and KO. Peasah, "Efficient deniable authentication and its application in location-based services," *Computers and Electrical Engineering*, vol. 100, p. 107958, 2022.

[4] Y. Cao, J. Wei, F. Zhang, Y. Xiang, and X. Chen, "Efficient public-key authenticated deniable encryp-

Table 4: Comparison of information entropy

Safety requirements	Ref. [4]	Ref. [5]	**Proposed**
Providing user anonymity	Yes	Yes	Yes
Provide mutual authentication	Yes	No	Yes
Provides forward security	Yes	Yes	Yes
Deniability	Yes	Yes	Yes
Resistant to replay attacks	Yes	Yes	Yes
Resistant to man-in-the-middle attacks	Yes	Yes	Yes
Resistant to offline keyword guessing attacks	Yes	Yes	Yes
Resistant to impersonation attacks	Yes	Yes	Yes
Resistant to statistical attacks	-	-	Yes
Resistant to differential attacks	-	-	Yes

tion schemes," *Computer Standards & Interfaces*, vol. 82, p. 103620, 2022.

[5] Y. Cao, F. Zhang, C. Gao, and X. Chen, "New practical public-key deniable encryption," in *Information and Communications Security: 22nd International Conference, ICICS 2020, Copenhagen, Denmark, August 24–26, 2020, Proceedings 22*, pp. 147–163. Springer, 2020.

[6] S. Chakraborty, D. Hofheinz, U. Maurer, and G. Rito, "Deniable authentication when signing keys leak," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 69–100. Springer, 2023.

[7] N. Chen, B. Chen, and W. Shi, "A Cross-layer Plausibly Deniable Encryption System for Mobile Devices," in *International Conference on Security and Privacy in Communication Systems*, pp. 150–169. Springer, 2022.

[8] P. W. Chi, M. H. Wang, and Y. H. Chuang, "A LWE-Based Receiver-Deniable Encryption Scheme," in *2021 International Conference on Security and Information Technologies with AI, Internet Computing and Big-data Applications*, pp. 124–133. Springer, 2022.

[9] D. Collins and S. Colombo, "Real World Deniability in Messaging," *Cryptology ePrint Archive*, 2023.

[10] J. Cui, Y. Wang, J. Zhang, and Y. Xu, "Full session key agreement scheme based on chaotic map in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8914–8924, 2020.

[11] D. S. Gupta, S. K. H. Islam, and M. S. Obaidat, "A Novel Identity-based Deniable Authentication Protocol Using Bilinear Pairings for Mobile Ad Hoc Networks," *Adhoc & Sensor Wireless Networks*, vol. 47, 2020.

[12] P. Jiang, B. Qiu, and L. Zhu, "Report when malicious: Deniable and accountable searchable message-moderation system," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1597–1609, 2022.

[13] C. Jin, G. Chen, C. Yu, and J. Zhao, "Deniable authenticated encryption for e-mail applications," *International Journal of Computers and Applications*, vol. 42, no. 5, pp. 429–438, 2020.

[14] C. Jin, G. Chen, C. Yu, J. Zhao, Y. Jin, and J. Shan, "Heterogeneous deniable authentication and its application to e-voting systems," *Journal of Information Security and Applications*, vol. 47, pp. 104–111, 2019.

[15] C. Jin, G. Kan, G. Chen, C. Yu, Y. Jin, and C. Xu, "Heterogeneous deniable authenticated encryption for location-based services," *Plos one*, vol. 16, no. 1, p. e0244978, 2021.

[16] C. V. Joe and J. S. Raj, "Deniable authentication encryption for privacy protection using blockchain," *Journal of Artificial Intelligence and Capsule Networks*, vol. 3, no. 3, pp. 259–271, 2021.

[17] Y. Kang and Z. Jiang, "Concretely Efficient Deniable Encryption Scheme from Single-key Functional Encryption," *Journal of Cryptologic Research*, vol. 9, no. 2, pp. 353–378, 2022.

[18] J. Kar and J. Kar, "Provably secure certificateless deniable authenticated encryption scheme," *Journal of Information Security and Applications*, vol. 54, p. 102581, 2020.

[19] J. Kar, K. Naik, and T. Abdelkader, "An efficient and lightweight deniably authenticated encryption scheme for e-mail security," *IEEE Access*, vol. 7, pp. 184207–184220, 2019.

[20] F. Li, J. Hong, and A. A. Omala, "Practical deniable authentication for pervasive computing environments," *Wireless Networks*, vol. 24, pp. 139–149, 2018.

[21] Z. W. Lin, T. H. Liu, and P. W. Chi, "LED: Learnable Encryption with Deniability," in *International Computer Symposium*, pp. 649–660. Springer, 2022.

[22] W. M. Shi, B. B. Chen, Y. M. Wang, Y. H. Zhou, and Y. G. Yang, "A quantum deniable authentication protocol based on two-mode squeezed quantum states," *Optik*, vol. 220, p. 165146, 2020.

[23] C. E. J. Singh and C. A. Sunitha, "Chaotic and Paillier secure image data sharing based on blockchain and cloud security," *Expert Systems with Applications*, vol. 198, p. 116874, 2022.

[24] Y. Xu, Z. Xia, Z. Wang, X. Zhang, and J. Weng, "Sender-deniable image steganography," *Journal of image and graphics*, pp. 760–774, 2023.

[25] Y. Xu, Z. H. Xia, Z. C. Wang, X. P. Zhang, and J. Weng, "Deniable steganography," *arXiv preprint arXiv:2205.12587*, 2022.

[26] P. Yan, K. Wang, and X. Jia, "A Covert Communication Method Combining Secret Sharing and Steganography," in *2022 IEEE 2nd International Conference on Computer Systems (ICCS)*, pp. 140–148. IEEE, 2022.

[27] X. Yan, X. Wang, and Y. Xian, "Chaotic image encryption algorithm based on arithmetic sequence scrambling model and DNA encoding operation," *Multimedia Tools and applications*, vol. 80, pp. 10949–10983, 2021.

[28] S. Zeng, Y. Mu, H. Zhang, and M. He, "A practical and communication-efficient deniable authentication with source-hiding and its application on Wi-Fi privacy," *Information Sciences*, vol. 516, pp. 331–345, 2020.

[29] S. Zhu, X. Deng, W. Zhang, and C. Zhu, "Image encryption scheme based on newly designed chaotic map and parallel DNA coding," vol. 11, no. 1, p. 231, 2023.

Biography

Qiu-yu Zhang Researcher/Ph.D. supervisor, graduated from Gansu University of Technology in 1986, and then worked at school of computer and communication in Lanzhou University of Technology. He is vice dean of Gansu manufacturing information engineering research center, a CCF senior member, a member of IEEE and ACM. His research interests include network and information security, information hiding and steganalysis, multimedia communication technology.

Yi-lin Liu is currently a master student of the School of Computer and Communication, Lanzhou University of Technology, China. She received the BS degrees in software engineering from Lanzhou University of Technology, in 2020. Her research interests include network and information security, multimedia data security and research on lightweight image encryption methods.

Guo-rui Wu is currently a master student of the School of Computer and Communication, Lanzhou University of Technology, China. She received the BS degrees in network engineering from Lanzhou Institute of Technology, Gansu, China, in 2020. Her research interests include network and information security, multimedia data security, blockchain.

Doc2vec-GRU: A Behavior Classification Method for Malicious Code

Haiming Wang[1], Yuntao Zhao[1], and Zijun Wang[2]
(Corresponding author: Yuntao Zhao)

School of Information Science and Engineering, Shenyang Ligong University[1]
Science and Technology Department, Shenyang Ligong University[2]
Shenyang 110159, China
Email: zhaoyuntao2014@163.com

(Received Apr. 29, 2023; Revised and Accepted Sept. 23, 2023; First Online Apr. 25, 2024)

Abstract

Due to the highly developed state of today's cyberspace, many malware programs appear on the internet. Although many detection software programs for malicious code are currently available on the network, the analysis of such code still heavily relies on manual inspection. Manual inspection requires expertise to analyze the behavior of malware. This paper presents a pattern for classifying the behavioral characteristics of malicious code. The assembler code statements are regarded as text. A regular expression is designed following the API call statements in the assembler code. Then, the regularization expression extracts the API calls, which are used as the input of the Doc2vec model and vectorized. The result of it is used as the input of the classification model. A model containing three GRU layers is built for multi-classification, and an attention mechanism is added. Classification results are evaluated by loss value and accuracy. Finally, the classification result shows it can achieve high accuracy, and the loss becomes stable. Demonstrated the model understands and fits the data.

Keywords: API Calls; Attention Mechanism; Doc2vec; GRU Model; Malicious Behavior

1 Introduction

Malicious code is a type of malicious software intentionally written and designed to attack computer systems, steal data, and damage hardware or software. It may enter a computer system through email, malicious websites, or other means to cause different levels of threat and harm to the user's computer system and data. There are many kinds of malicious codes, including viruses, worms, Trojan horses, malicious adware, spyware, ransomware, mining software, etc. They attack with different means and purposes, but all pose threats and hazards to computer systems and users' privacy. With the continuous development of computer technology, malicious codes are also upgrading and evolving, and the attack methods are becoming more and more covert and difficult to defend. Hackers can use various techniques to bypass antivirus software, such as code mutation, code obfuscation, and exploiting vulnerabilities. By mutating malicious code and changing its characteristics, hackers make it difficult to be detected and identified by antivirus software; or using code obfuscation tools to obfuscate malicious code, making it difficult to be detected and identified by static analysis and dynamic analysis techniques; they can also exploit vulnerabilities in software such as operating systems, browsers and applications to bypass the defense mechanism of antivirus software and make malicious code successfully invade computer systems. Therefore, protecting computer and network security against malicious code attacks and invasions has become an issue that cannot be ignored. Effective security measures and technical means are needed to improve their security defenses and protect the computer and network security of individuals and organizations.

When detecting malicious code, the key issue is how to detect the behavior of the malicious code, which does not necessarily paralyze the computer system but behaves with certain malicious intent, such as: recording the input of the computer user's keyboard; using the infected computer to send information outside the local area network; modifying registry information or deleting files in the computer. The malicious code is accompanied by system API calls during the execution of the above actions. Accurately classifying malware families remains a difficult problem, as malware continues to evolve and mutate.

To address this challenge, researchers have proposed several approaches to detect and classify malware. These include The literature [11]proposes a technique to analyze malware by running malware samples in the environment and monitoring the activities caused by the malware samples. The literature [18]uses recurrent neural networks (RNN) and long short-term memory (LSTM) networks to generate uniform feature embeddings for each binary

file and computes similarity measures for binaries using concatenated neural networks. The literature [12]uses a model that uses a bidirectional encoder representation from Transformers (BERT) to detect malware. The literature [14]proposes a novel feature representation method for malware detection that presents a malware classification and detection system using a hybrid approach of migration learning and texture features. The literature [3] reveals the importance of step size in malicious code classification using RNN and the highest AUC using Word2Vec feature vectors found by comparing different feature vectors. The literature [17]introduced an accurate static malware detection system designed specifically for Windows environments that effectively identifies and categorizes portable executable (PE) malware as benign or malicious. The literature [7]proposes an incremental malware detection model for meta-feature APIs and system call sequences. The literature [8]develops an accurate RNN model that utilizes information gain-based feature selection to identify the most relevant features for malware detection and effectively classify malware and benign software, outperforming other machine learning methods. The contribution of the paper [6]is the introduction of a convolutional Recurrent neural Network (CRNN) method for malware detection using opcode sequences, which shows improved accuracy compared to traditional machine learning models

In the field of malicious code analysis, Natural Language Processing (NLP) plays a critical role in feature extraction and identifying novel, unfamiliar malicious code. NLP methods convert code into text, enabling text mining to extract features. For example, N-gram models can capture common API sequence patterns and convert them into feature vectors. In addition, NLP helps process large amounts of log data from dynamic analysis to extract structured features from keywords in API call sequences. In machine learning, NLP helps in text classification, employing techniques such as bag-of-words and deep learning models to detect new, unrecognizable malicious code.

The literature [5]proposes a framework for early-stage malware detection and mitigation by leveraging NLP techniques and machine learning algorithms. The literature [16]places API call sequences as the subject of purification and optimization processes. The literature [2]uses common subsequences of API calls to study association rule-based malware classification. The literature [1]proposes embedding modules to convert Windows API function parameters, registries, filenames, and URLs into low-dimensional vectors while still retaining proximity properties. The approach proposed in [4]can be applied to real-time malware threat search, especially for security-critical systems. The literature [13]proposes an intelligent detection system for binary code vulnerabilities based on program slicing. The literature [9]proposes a method for constructing malware variants and API sequence datasets and provides a pre-trained malware detection model based on BERT. The literature [10] uses Convolutional Neural Networks (CNNs) to extract features from API call sequences and train classifiers for malware detection. The literature [15]proposes a malware detection method that combines the use of API calls, TF-IDF, and RNNs.

Current behavioral analysis against malicious code is performed in two main ways. One is static analysis, where the binary code is disassembled and the analyst analyzes it manually or using a static analysis tool such as IDA Pro by analyzing the code logic and identifying the controls between blocks of code. The other is dynamic analysis, which involves running the malware in a controlled environment to observe its behavior and interaction with the system. Both static and dynamic analysis have limitations. Static analysis of malicious code in assembly form requires a combination of technical skills, knowledge of the behavior of the malicious code, and the use of specialized tools and environments. Dynamic analysis involves running malware in a controlled environment, and runtime detection techniques involve modifying the behavior of runtime malware by injecting code into the process. This can be used to intercept and log system calls, network traffic, and other events triggered by malware. Both static and dynamic analysis of malicious code require a significant investment of effort to perform, and a method is needed to be able to classify behavior without running malicious code against its API calls.

Therefore, this paper proposes a method to extract API sequences from malicious code in assembly form and then classify the behavior. The main contributions of this paper include:

1) Propose a regular expression to extract API from assembly code. The IAT (Import Address Table) is a data structure in the PE file that contains the names of dynamic link libraries (DLLS) and function names on which the executable depends. During disassembly, function names in the IAT can be mapped to the corresponding API. The API call sequence is extracted by pattern matching of assembly code with a regular expression. Regular expression is a method based on pattern matching, so it can quickly extract all API call sequences from assembly files, greatly improving efficiency. You can customize regular expressions as needed to extract specific sequences of API calls to better suit different application scenarios. This method can extract the API call sequence in assembly files quickly and efficiently, reduce the workload of manual analysis, and thus save time and cost;

2) Build Doc2vec model. The sequence of API calls extracted from the assembly code is taken as input and output vectorized. The advantage of the Doc2vec model is that it can transform unknown terms into vector representations and calculate the relationships between word vectors to better express the semantics of documents. The traditional word bag model or TF-IDF model regards each document as a set of discrete words, ignoring the context information between words. The Doc2vec model takes into ac-

count contextual information between words so that the semantics of a document can be more accurately expressed;

3) A classification model with three GRU layers and one attention layer is established. GRU is a recursive neural network that can process variable length sequences efficiently and extract important feature information from them. GRU can model contextual information in sequences to better capture semantic information in API sequences. Compared with the traditional classification method of the word bag model, GRU can more accurately understand the relationship between words in the input sequence. The use of the attention mechanism can make the model pay more attention to the important sequence parts, to improve the accuracy of classification;

2 Related Research

Malicious code behavior analysis refers to the process of in-depth analysis of malicious code (e.g. viruses, Trojans, worms, etc.) to understand its attack behavior, propagation methods, functions, characteristics, and other information. In the field of computer security, malicious code behavior analysis is an important technique that can help security experts identify and respond to various threats. Malicious code analysis technology refers to a technology that analyzes and researches malicious code samples through various means and methods, to obtain its relevant information and behavioral laws. These techniques include static analysis, dynamic analysis, obfuscated code analysis, machine learning, etc.

RNN and LSTM are two important neural network structures in the field of deep learning. They can be used for modeling and predicting sequence data and hence are useful in analyzing sequence data such as API sequences. GRU is a recurrent neural network similar to LSTM for solving long-term dependency problems. They are both designed to solve the problem of vanishing or exploding gradients that RNNs encounter when dealing with long sequence data, while also allowing the network to be able to selectively forget some information.

2.1 Malware Analysis

Static analysis of malicious code is the reverse analysis of malicious code through disassembly and decompilation to obtain the logic and function of the malicious code and the behavior that may be generated. This method can be used to analyze the executable file of the malicious code. Dynamic analysis, on the other hand, involves running the malicious code in an infected system or virtual environment, observing its behavior and impact on the system, collecting relevant data, and analyzing it to gain a more comprehensive understanding of the malicious code's behavior. This approach can be used to analyze the behavior of malware and the process of cyber attacks. Data mining: Analyze the data collected by the malicious code, system logs, network traffic, and other data to understand the behavior of the malicious code, the propagation path, the target of the attack, the degree of victimization, and other information.

The Malicious code static analysis technique is a technique to identify malicious behavior by analyzing the code structure and features of malicious code. Malicious code disassembly analysis is a technical means to reverse engineer malware. By disassembling malicious code, the assembly code of its underlying machine instructions can be obtained to gain insight into the implementation principle and function of malicious code. By analyzing the disassembled code and understanding the implementation logic and specific functions of the malicious code, it is possible to understand whether there are vulnerabilities or backdoors, as well as information about its association with other malware. This technique is performed without running the malicious code, thus further damage to the computer system can be avoided. Static binary analysis is a technique that determines malicious behavior and operating system calls by performing reverse analysis of binary files. This technique can also be performed without running malicious code, thus avoiding additional risks to the system. The main steps of static binary analysis include file analysis (identifying the format and structure of the binary file, including headers, segments, sections, etc.), disassembly (converting binary code to assembly code to better understand its structure and functionality), analysis of import and export tables (looking at import and export tables to understand the libraries and API calls that the program relies on), control flow analysis (analyzing the program's control flow to identify code blocks and functions), data flow analysis (tracing the flow of data in a program to determine the values of variables and constants), symbolic execution (using symbolic variables to model various paths of code execution to identify potential vulnerabilities and attack vectors), and behavioral analysis (analyzing the behavior of a program to identify potentially malicious behavior and types of attacks). Through these analysis techniques, static binary analysis can help security researchers better understand the structure and function of binaries to identify malicious behavior and attack vectors.

Dynamic analysis of malicious code is a technique that runs malicious code on an infected system and monitors its behavior to identify malicious behavior. Unlike static analysis, dynamic analysis can provide more information, such as how the malicious code affects the system and how it interacts with other programs. Dynamic analysis typically involves the following steps: running malicious code in a secure environment, monitoring its behavior and collecting behavioral data, then analyzing the data to identify malicious behavior and attack vectors, and taking timely action to prevent the malicious code from impacting the system. For example, antivirus software, firewalls, and quarantine of infected systems can be used to secure systems and data. Dynamic analysis techniques

can help security researchers gain insight into the behavior and functionality of malicious code to better identify and prevent malicious behaviors and attacks. However, it is important to note that the process of dynamic analysis may expose infected systems to risk, and therefore appropriate preventive and security measures need to be taken to secure systems and data.

2.2 PE File Reverse Analysis

The import table in the PE file records the names and addresses of external functions or variables that the program needs to refer to. When the program executes the code that needs to call the external function or variable, it will look up the corresponding name and address in the import table, and then jump to that address to execute the corresponding code. The name in the import table is usually the function name or serial number of the API. the function name or serial number of the API is recorded in the export table of the operating system, which provides many API functions for the program to call. When a program calls an API function, it looks up the name or serial number of the function in the export table of the operating system and then jumps to the address of the function to execute the corresponding code. Therefore, there is a mapping relationship between the import table in the PE file and the export table of the operating system. The API name or serial number in the import table will be mapped to the corresponding function address in the export table of the operating system for the program to call the API function correctly. This mapping relationship is an important link between PE files and APIs.

The export table lists the functions and variables in the program that can be called by external programs, which is important for dynamic link libraries (DLLs). The Resources table contains the various resources used by the program, such as icons, bitmaps, strings, menus, etc., which can be used internally or externally by the program. The relocation table contains information about the relocation that needs to be done when the program is loaded. The relocation table can be used to correct address references in the program when the program is loaded at a non-default address. These data structures can help the analysis to better understand the structure and function of the program for more in-depth analysis and disassembly. As it is shown in Figure 1

The IAT (Import Address Table) in the PE file records the names and addresses of external functions and dynamic link libraries that the program needs to use to dynamically load and call these functions at program runtime. The function names in the IAT can be mapped to the corresponding APIs. Reverse analysis of the PE file to obtain the names and addresses of the external functions recorded in the IAT. The code segments that use these functions are then found in the reverse process and disassembled into assembly code. There is a strong connection between the behavioral characteristics of the malicious code and the API calls. Malicious code usually

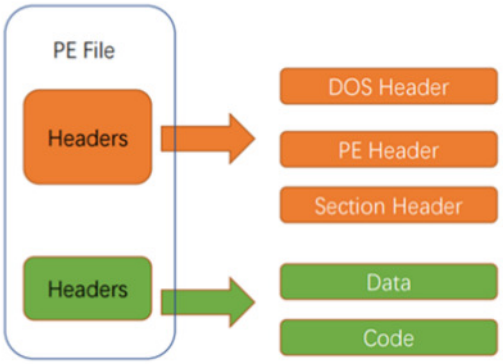


Figure 1: PE file

performs its malicious behavior through API calls, which are a set of predefined functions or methods provided by the operating system or third-party libraries that can be used to perform various operations, such as file reading and writing, process control, network communication, etc. The reverse analysis process of malicious code through the API calls helps to determine its behavioral characteristics. When malicious code calls API functions related to network communication, file operations, etc., it is likely that the malicious code is designed for network attacks, data theft, file tampering or stealing, and other behaviors. By analyzing the API functions called by the malicious code, the main functional and behavioral characteristics can be understood.

When IDA Pro reverse analyzes a binary file, it uses feature information from the FLIRT database to identify known library functions. If IDA Pro finds feature information in the FLIRT database in the binary file, it will automatically add the function to the disassembly code and display the name and parameter information of the function in the disassembly code.

2.3 Gate Recurrent Unit

GRU is a variant of recurrent neural networks that solves the gradient vanishing problem in standard RNNs and better captures long-term dependencies in sequential data by including reset and update gates, such as speech recognition, machine translation, and natural language processing. GRU is better than standard RNNs. It also has a more simplified memory unit than LSTM, and thus is faster to compute with comparable performance to LSTM.

In malicious code classification, the API call sequence is sequence data, and the GRU model can encode each element in the sequence and predict the subsequent elements based on the previous encoding information. Compared with other traditional sequence models, the GRU model can effectively reduce the gradient disappearance and gradient explosion problems when learning, thus im-

proving the performance and generalization ability of the model. In addition, the GRU model has fewer parameters, making it relatively fast to train and capable of handling variable-length sequence data, which makes it more suitable for applications in NLP fields such as malicious code classification.

The structure of Reset and update gates is shown in Figure 2. For a given time step t, assume that the input is a small batch $x_t \in R^{n*d}$ while the hidden state of the previous time step is $H_{t-1} \in R^{n*h}$ and then the reset gate $R_t \in R^{n*h}$ and the update gate z $z_t \in R^{n*h}$ are calculated as follows:

$$R_t = \sigma(X_t W_{x_r} + H_{t-1} W_{h_r} + b_r) \tag{1}$$

$$Z_t = \sigma(X_t W_{x_z} + H_{t-1} W_{h_z} + b_z) \tag{2}$$

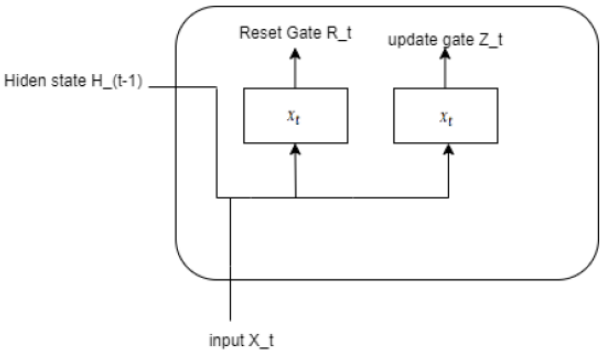


Figure 2: Reset and update gates in the GRU model

Figure 3 shows Computing the hidden state in the GRU model. The GRU contains two gating units: A reset gate and an update gate. The activation functions of these gates are sigmoid functions that are used to limit the output value of the gate between 0 and 1. The reset gate controls the degree of influence of the previous state on the current state, while the update gate controls the degree of influence of the previous state and the current input on the next state.

$$H_t = Z_t \bigodot H_{t-1} + (1 - Z_t) \bigodot \tilde{H}_t \tag{3}$$

3 Design of Classification Method

The experimental scheme is as follows,

Step 1: Read the API call sequences in each ASM file in the training set using regularization rules, tag them according to the names of the ASM files, and save the results to a CSV file.

Step 2: Feed the API call sequences obtained in the previous step into Doc2vec, and represent the API call sequences with a 50-dimensional vectorization.

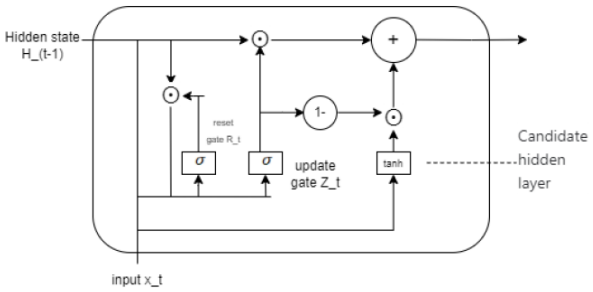


Figure 3: Computing the hidden state in the GRU model

Step 3: The vectorized API sequences obtained in Step 2 are fed into the GRU network with an attention mechanism for the multi-classification task, and the classification effect of the model is evaluated using loss value and accuracy. The overall procedure is illustrated by Figure 4.

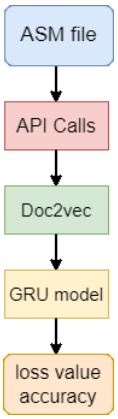


Figure 4: Experimental flow graph

3.1 Malware Dataset

To study and classify malware, the Microsoft Malware Classification Challenge (BIG 2015) dataset was used. The dataset contains labeled data for training and unlabeled data for testing, totaling about 500 GB. The dataset contains malware from nine different families, including Ramnit, Vundo, Tracur, and Obfuscator.ACY, which belongs to the worm category, as well as Gatak, which belongs to the Trojan category, Kelihos_ver3 and Kelihos_ver1, which belong to the Botnet category, and Simda, which belongs to the backdoor category. Kelihos_ver3 and Kelihos_ver1 are in the Botnet category, and Simda is in the Backdoor category. The total number of malware files in the dataset is 10,868, and these files are divided into byte files and ASM files. Unlike binary files, the ASM file section contains intuitive assembly code. In the experiments, only training data was used, of which 80% was used for training and 20% for testing. The

amount of malware varies between families, with families 4, 5, 6, and 7 having lower amounts. To avoid excessive computation, this study used only some of the six malware families with high data volumes for training. Each family of the training data and the corresponding number are shown in Table 1.

Table 1: Malware train data

label	Family name	The number of ASM file
1	Ramnit	908
2	Lollipop	1375
3	Kelihos_ver3	1806
6	Tracur	150
8	Obfuscator.ACY	219
9	Gatak	623

3.2 Reverse Analysis API Extraction

Binary reverse analysis is the process of converting hexadecimal numbers into human-readable assembly instructions. This paper takes the form of static analysis to study the behavioral characteristics of malicious code. Static analysis is performed on captured malicious code samples without running them, and malicious code samples are reverse analyzed to readable assembly code. The information in the malicious code is analyzed to infer the logic and behavior of the program. To deeply analyze and understand the function, behavior, and attack method of the malicious code. Malicious code disassembly can reveal the true intent, command and control flow of malicious code, and can reveal hidden features, encryption algorithms, and defense mechanisms of malicious code. Figure 5 shows an example of the malware disassembles fragments.

```
.text:00401000 ; =============== S U B R O U T I N E =======================================
.text:00401000
.text:00401000 ; Attributes: bp-based frame
.text:00401000
.text:00401000 sub_401000      proc near               ; CODE XREF: _main+13↓p
.text:00401000
.text:00401000 phkResult       = dword ptr -8
.text:00401000 var_4           = dword ptr -4
.text:00401000
.text:00401000                 push    ebp
.text:00401001                 mov     ebp, esp
.text:00401003                 sub     esp, 8
.text:00401006                 lea     eax, [ebp+phkResult]
.text:00401009                 push    eax             ; phkResult
.text:0040100A                 push    0F003Fh         ; samDesired
.text:0040100F                 push    0               ; ulOptions
.text:00401011                 push    offset SubKey   ; "SOFTWARE\\Microsoft \\XPS"
.text:00401016                 push    80000002h       ; hKey
.text:0040101B                 call    ds:RegOpenKeyExA
.text:00401021                 test    eax, eax
.text:00401023                 jz      short loc_401029
.text:00401025                 xor     eax, eax
.text:00401027                 jmp     short loc_401066
.text:00401029 ; ---------------------------------------------------------------------------
.text:00401029
.text:00401029 loc_401029:                             ; CODE XREF: sub_401000+23↑j
.text:00401029                 push    0               ; lpcbData
.text:0040102B                 push    0               ; lpData
0000101B 0040101B: sub_401000+1B (Synchronized with Hex View-1)
```

Figure 5: Malicious code disassembles fragments

Disassembling malicious code can help analyze the instructions and flow of binary files and can determine the functions and system calls used by the program, among others. It is used to examine, manipulate, and modify executable files. When it comes to reverse analysis of malicious code, behavioral characteristics are very important. The assembly code contains information such as the machine instruction that operates, the marker that marks the location, and the memory cell that stores the data or performs the operation. The instruction is usually a basic operation, such as moving data, performing arithmetic operations, comparing values, etc. The operands are the inputs and outputs of the instruction. Instructions and operands in assembly code usually manipulate memory addresses, which can be code segments, data segments, stacks, etc. of a program. Assembly code usually contains function and call instructions that divide the code of a program into reusable modules. Assembly code usually contains system API call instructions that allow the program to access operating system functions such as file reading and writing, network communication, process control, etc.

Regular Expressions, consisting of word symbols and operators, are a means of pattern matching and text processing. Regular expressions can be used to find specific strings, URLs, or words in text. The principle of searching strings with regular expressions is based on the matching rules of regular expressions and the structure of strings. To search for a string with a regular expression, you first define a matching pattern, which consists of a regular expression and a specific flag. Then the matching pattern is matched against the string to be searched, checking whether there are.

In this paper, we propose a regular expression to extract APIs from assembly code based on the way APIs are called in assembly code. the regularized expression used to extract API sequences: r"\s+call\s+ds:(?P<api_name>[_a-zA-Z0-9]+)", the expression '\s' means match one or more space characters (e.g.space, character newline, etc.). 'call' matches the call, and 'ds:' matches the character ds.(?P<api_name>[_a-zA-Z0-9]+) is a named capture group. This pattern matches any space character before or after the statement, from the capture API name in the assembly statement (which can only contain letters, numbers, and underscores). By applying this regular expression pattern to each line of code in the individual assembly file of the read dataset, a series of API calls can be extracted.

3.3 API Sequence Vectorization

In the previous step, the sequence of API calls is extracted from the assembly code and treated as text. The API call sequences are then trained using the Doc2vec method, which is an unsupervised algorithm that learns a fixed length feature representation, enabling the classification of API call sequences of malicious code in the next step.

Using the Doc2vec algorithm, sentences, paragraphs, and documents can be represented as vectors. This is an extended version of the Word2Vec algorithm with several

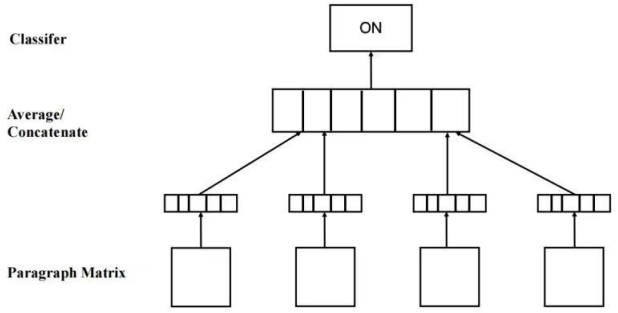


Figure 6: Doc2vec model

advantages, such as not requiring a fixed sentence length and accepting sentences of different lengths as training samples. Using the Doc2vec algorithm, malicious codes can be classified more accurately because the structure of the model can overcome the drawbacks of the bag-of-words model. Figure 6 illustrates the structure of the doc2vec model.

3.4 GRU Model

The reason why the GRU model performs well in malicious code API classification is that the GRU model has the memory capacity to process sequential data. after the API sequences are fed into the vectorized output of the Doc2vec model, the output vector is fed as input to the GRU network for the multi-classification task. Figure 7 illustrates the structure of the classification model.

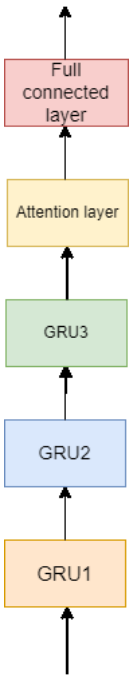


Figure 7: Classification model

Adding an attention mechanism to the GRU model improves its performance of the GRU model because it allows the model to focus more on key information when processing sequential data. The attention mechanism allows the model to visualize the features it focuses on, thus improving the interpretability of the model. It can be easier to understand the model's decisions and identify which features have the most influence on the model's predictions.

A model containing three GRU layers and an attention layer with 32 neurons per GRU layer to prevent overfitting was built through the Keras deep learning framework. The shape of the input tensor consists of two dimensions of data, the first dimension is the time step, the second dimension is the number of features, and the last dimension is 1 (representing a univariate time series). Additionally, an attention mechanism is applied in the model to improve the mining of the key information of the time series. Ultimately, the returned sequence model can return the output sequence for each time step as needed. The model is compiled using an Adam optimizer with a loss function of categorical cross-entropy and an evaluation metric of accuracy. The attention mechanism is then used to enhance the expressiveness of the GRU layer, and finally, the prediction results are output using a fully connected layer.

To address the problem of overfitting during model training, a 20% Dropout is applied after each GRU layer, and L1 regularization is also added to the Dense layer before the output layer. Dropout is a regularization technique that ignores randomly selected neurons during training. L1 regularization is a technique used in machine learning that works by adding a penalty term to the loss function that is proportional to the absolute value of the weight,.s. The model uses smaller weights, which leads to sparsity in the weight matrix. Smaller weights reduce the complexity of the model and can help prevent overfitting by limiting the ability of the model to fit noisy or irrelevant features in the training data.

4 Result

4.1 Experimental Environment

The experiment was conducted on a Windows 11 computer system with i5-11400H CPU and 16G RAM. The development environment was Python 3.7 and Pycharm Professional, TensorFlow 2.10.0 was used to build the GRU model, gensim was used to build the doc2vec model, and matplotlib was used to plot the results.

4.2 Extracting API Call Sequences from Assembly Code

This experiment imports the necessary libraries, such as "os" for OS-related functions, "re" for regular expressions, "chardet" for detecting file encoding, and "CSV" for reading and writing CSV files for reading and writing CSV files. Then define the regular expression pattern to

match the API calls in the ASM file content. Specify the path to the folder where the ASM file is located. Next, open the CSV file in write mode and create a DictWriter object with the field names. The code loops through the files in the folder with the ASM extension detects the encoding for each file using the "chardet. detect()" function, decodes the file content using the detected encoding, and reads the content. API calls are extracted from the content using regular expression patterns and stored in a list. The file ID and the extracted API call are written to a CSV file. This process is repeated for all ASM files in the folder. If there is a CSV file in the dataset, each file is tagged with a category by file name and, the two CSV files are merged by file name. The result is saved as a CSV file. Figure 8 shows some of the results.

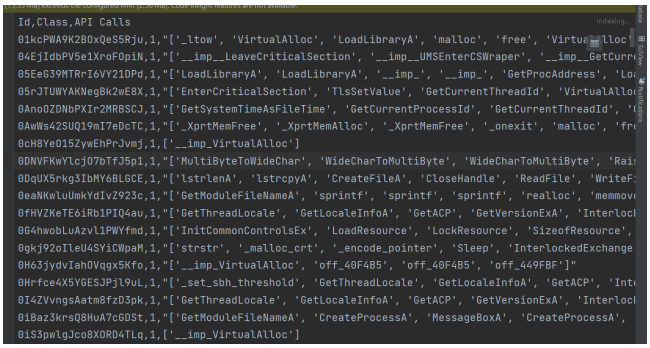


Figure 8: Extract the API sequence and label it

4.3 API Calls for Sequence Vectorization

The extracted API sequence was sent to the Doc2vec model for vectorization, and the input API call sequence was represented by a 100-dimensional vector. The Gensim library was used to build the Doc2vec model, the generator was transformed into a repeatable sequence, and the doc2vec model was trained to represent the API sequence. The vectorized representation of one of the API sequences is shown below in Figure 9.

```
C:\Users\12513\python.exe D:\ML\GRU+attention\doc.py
[ 0.00704034  0.00566528 -0.00306355  0.00987099  0.00248704 -0.00436699
  0.0044907   0.00096673  0.00463677 -0.00385698 -0.00745628  0.00733446
  0.00030128 -0.00329935  0.00952603 -0.00030466 -0.00159476 -0.00362364
  0.00728839 -0.00362473 -0.00550567 -0.00825513 -0.00340968 -0.00713346
  0.00241553  0.00427818 -0.00990101 -0.0030985  -0.0049099  -0.00301215
 -0.0014658  -0.00452034 -0.00585548 -0.00695248 -0.00158894  0.00548736
-0.00263824  0.00242334 -0.00222537 -0.00241051 -0.00856188  0.00302902
  0.00169214  0.00081569 -0.00841346  0.00059554  0.00087606 -0.00339213
  0.00196027  0.00307339]
```

Figure 9: API sequence vector representation

4.4 GRU Model Classification Results

The data set used in this experiment is six types of data, and the data are divided into training and testing sets according to the ratio of 8:2. In this paper, the accuracy of the model multi-classification is evaluated in terms of accuracy and loss values. The Num_epoch value of the training process is set to 1000, the batch_size value is 32, and the L1 regularization strength is set to 0.001.

In Keras, the optimizer, loss function, and evaluation metric can be set using the model's compile() function. Among them, accuracy is a common evaluation metric that measures the classification accuracy of the model on both the training and test sets. After the model is trained, the model. evaluate() function can be used to calculate the accuracy and loss values of the model on the test set. the first argument of the model. evaluate() function is the input data of the test set, and the second argument is the label data of the test set. The first value in the return value of the function is the loss value on the test set, and the second value is the accuracy value on the test set.

By training the model and recording the training process, the loss and accuracy curves can be plotted. These plots can be used to evaluate the effect of model training. The effect of model training is shown in Figure 10.

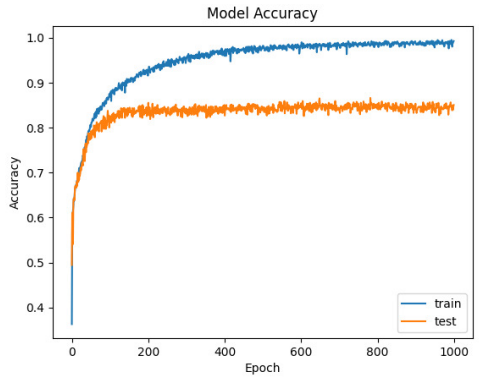


Figure 10: GRU model training accuracy curve

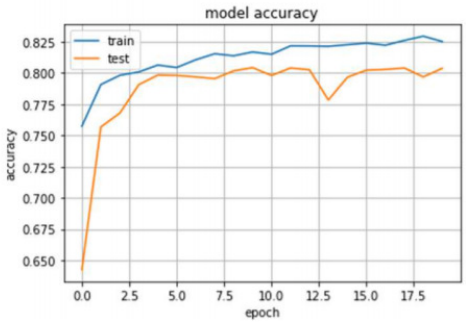


Figure 11: LSTM model training accuracy curve

In Figure 10, it can be seen that the accuracy of the

model classification is low at the beginning, and the accuracy of the model training increases after a certain number of training sessions, and the value of the accuracy leveled off after 200 training sessions. In contrast, as shown in Figure 11, the LSTM accuracy is not only lower than that of GRU but also the accuracy fluctuation does not converge to a fixed value during the training process.

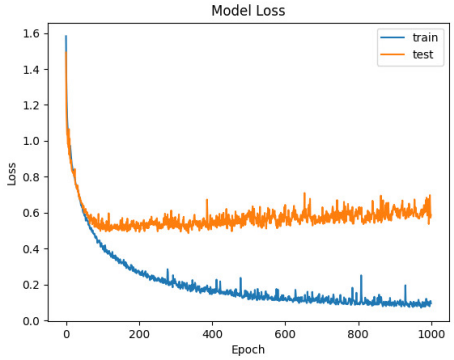


Figure 12: GRU model training loss curve

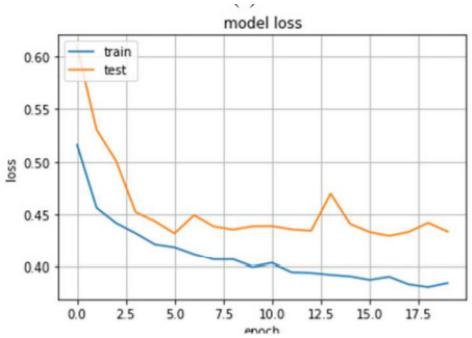


Figure 13: LSTM model training loss curve

Figure 12 shows that the model shows a decreasing trend in loss on both the training and test sets and converges after a certain number of training sessions. This indicates that the model has a strong learning ability and can fit the data well. In contrast, the LSTM model shown in Figure 13 shows a decreasing loss at the beginning but later shows some fluctuations, which indicates that the model may be affected by certain factors when classifying.

5 Conclusions

Observing the graphs, we can find that the model achieves good performance on both the training and test sets. After reaching a certain number of training times, the classification accuracy of the model reaches a certain value and starts to converge to a fixed value, and the loss of the model on both the training and test sets shows a decreasing trend, indicating that the model has a strong learning ability and can fit the data well.

This study uses static analysis to examine the behavioral features in the assembly code and extract useful information from them. However, static analysis has some shortcomings. Specifically, static analysis can only analyze the static structure of the malicious code, while it cannot obtain information about the dynamic behavior of the code. This means that it may not be able to obtain the complete runtime of the malicious code, and thus may miss some critical information. In addition, many malicious code authors use various techniques such as shelling and code obfuscation to protect their code from being detected. Some malicious code may use coding techniques similar to those of normal programs, which may cause static analysis to misidentify it as a normal program. Subsequent analysis of the behavioral characteristics of malicious code should consider the inclusion of dynamic analysis, combining dynamic and static analysis of malicious code to analyze the behavior of malicious code as accurately as possible.

Acknowledgments

This study was supported by the 2023 Liaoning Province Applied Basic Research Program Project, 2023JH2/101300203. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] Z. Cheng, T. Xie, P. Shi, C. Li, R. Nadkarni, Y. Hu, C. Xiong, D. Radev, M. Ostendorf, L. Zettlemoyer *et al.*, "Binding language models in symbolic languages," *arXiv preprint arXiv:2210.02875*, 2022.

[2] G. D'Angelo, M. Ficco, and F. Palmieri, "Association rule-based malware classification using common subsequences of api calls," *Applied Soft Computing*, vol. 105, p. 107234, 2021.

[3] S. Jha, D. Prashar, H. V. Long, and D. Taniar, "Recurrent neural network for detecting malware," *computers & security*, vol. 99, p. 102037, 2020.

[4] A. N. Jahromi, S. Hashemi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "An enhanced stacked lstm method with no random initialization for malware threat hunting in safety and time-critical systems," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 630–640, 2020.

[5] Z. Jamadi and A. G. Aghdam, "Early malware detection and next-action prediction," *arXiv preprint arXiv:2306.06255*, 2023.

[6] S. Jeon and J. Moon, "Malware-detection method with a convolutional recurrent neural network using opcode sequences," *Information Sciences*, vol. 535, pp. 1–15, 2020.

[7] P. Kishore, S. K. Barisal, and D. P. Mohapatra, "An incremental malware detection model for meta-feature api and system call sequence," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2020, pp. 629–638.

[8] P. Kumar, U. S. B, I. Mishra, S. S, D. R. Tripathi, and S. Rama Krishna T., "Malware detection classification using recurrent neural network," in *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, 2022, pp. 876–880.

[9] F. Lu, Z. Cai, Z. Lin, Y. Bao, and M. Tang, "Research on the construction of malware variant datasets and their detection method," *Applied Sciences*, vol. 12, no. 15, p. 7546, 2022.

[10] T. Mu, H. Chen, J. Du, and A. Xu, "An android malware detection method using deep learning based on api calls," in *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. IEEE, 2019, pp. 2001–2004.

[11] A. F. Muhtadi and A. Almaarif, "Analysis of malware impact on network traffic using behavior-based detection technique," *International Journal of Advances in Data and Information Systems*, vol. 1, no. 1, pp. 17–25, 2020.

[12] A. Rahali and M. A. Akhloufi, "Malbert: Malware detection using bidirectional encoder representations from transformers," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 3226–3231.

[13] J. Tian, W. Xing, and Z. Li, "Bvdetector: A program slice-based binary code vulnerability intelligent detection system," *Information and Software Technology*, vol. 123, p. 106289, 2020.

[14] F. Ullah, G. Srivastava, and S. Ullah, "A malware detection system using a hybrid approach of multi-heads attention-based control flow traces and image visualization," *Journal of Cloud Computing*, vol. 11, no. 1, pp. 1–21, 2022.

[15] B. A. V. Vidyapeetham, "Api call based malware detection approach using recurrent neural network—lstm," in *Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 1*, vol. 940. Springer, 2019, p. 87.

[16] S. Yesir and İ. Soğukpinar, "Malware detection and classification using fasttext and bert," in *2021 9th International Symposium on Digital Forensics and Security (ISDFS)*. IEEE, 2021, pp. 1–6.

[17] M. I. Yousuf, I. Anwer, A. Riasat, K. T. Zia, and S. Kim, "Windows malware detection based on static analysis with multiple features," *PeerJ Computer Science*, vol. 9, p. e1319, 2023.

[18] Z. Yu, R. Cao, Q. Tang, S. Nie, J. Huang, and S. Wu, "Order matters: Semantic-aware neural networks for binary code similarity detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1145–1152.

Biography

Haiming Wang was born in Ningde, Fujian, China, in 1999. He received a B.S. degree in telecommunication from Shenyang LiGong University, China, in 2021. He is currently pursuing an M.S. degree in communication engineering at Shenyang Ligong University, Shenyang, China. His research interests include network communications and security, deep learning, and malicious code classification technology.

Yuntao Zhao received a Ph.D. degree in control science and engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2013. He is currently a Postdoctoral Researcher in pattern recognition and artificial intelligence with Northeastern University, Shenyang, China, in 2015. He is also a Professor at the Communication and Network Institute and the School of Information Science and Engineering, at Shenyang Ligong University, Shenyang. He has authored over 30 papers published in related international conference proceedings and journals. He is the holder of ten patents and software copyrights. His main research interests include deep learning, AI algorithms, cyberspace security, protocol analysis, and data mining.

Zijun Wang was born in Shenyang, Liaoning Province, China, in 1984. He received an M.E degree in artillery, automatic gun, and ammunition engineering from Shenyang Ligong University, in 2009. He is currently pursuing a Ph.D. in armament science and technology at Shenyang Ligong University. His research interests include military communications, navigation, guidance, and materials science.

A Study on Influence Maximizing Based on Two Rounds of Filtration Metric in Social Networks

Yang Li[1] and Zhiqiang Wang[2]
(Corresponding author: Zhiqiang Wang)
State Information Center, Beijing 100045, China[1]
Email: liyang_cas@163.com
Beijing Electronic Science and Technology Institute, Beijing 100070, China[2]
Email: wangzq@besti.edu.cn
(Received May 9, 2023; Revised and Accepted Sept. 23, 2023; First Online Apr. 25, 2024)

Abstract

The influence maximization problem is discovering a seed set of nodes in a social network and making the spread as large as possible based on influence propagation. The current related algorithm based on the greedy strategy maintains a better influence propagation but has high time complexity and is not very scalable. This paper proposes a new method to solve the influence maximization problem by reducing the time complexity, called the Two Rounds of Filtration Metric (TRFM) algorithm. The main work is as follows:(1) A regional node metric is proposed based on the local topology to measure the nodes, which reduces the evaluation time. (2) The submodular characteristic is applied to discover the TOP-K seed node set from the candidate node set; meanwhile, the evaluation measurement in the whole network maintains a better influence propagation. The experimental results on the actual data set verify the effectiveness of the TRFM algorithm.

Keywords: Community Division; Greedy Strategy; Independent Cascade Model; Influence Maximization; Social Network

1 Introduction

1.1 Overview of Influence Maximization

Social networks are increasingly integrated into every aspect of our working life by the new generation of information technology. Users can follow the star, make friends, release information, and promote products through social networks such as Weibo, WeChat, Twitter, and Facebook. For example, a company develops a new cell phone and hopes to promote it through some stars to influence more people; As well as a company develops a new APP or a new cell phone and hopes to promote it through some famous bloggers to attract more users to participate by word of mouth, etc. Ultimately, we hope to maximize the influence of other users on social networks.

1.2 Problem of Influence Maximization

The applications mentioned above can be summarized as the influence maximization problem, i.e., we can take a social network graph, for example, where the graph nodes represent users in the social network, the edges of the graph represent user relationships in the social network, which can be described as a problem how to discover the set of k initial nodes in the graph that maximizes the spread of the final influence by given a specified propagation model. Several researchers have carried out extensive research work based on this problem. Kempe first represented the influence maximization study through a discrete optimization problem and proved it to be an NP-Hard problem with the simple greedy algorithm to achieve the optimal solution of (1-1/e). In subsequent research, some researchers keep optimizing the greedy algorithm to improve the performance further. Others propose some heuristic algorithms from scalability and keep advancing to deepen the research.

1.3 The Main Work of The Method in This Paper

In this paper, the search space is reduced by two rounds of node filtration, significantly reducing the running time. The experimental results on the public dataset verify the effectiveness of this paper, and the main work of this paper is as follows:

1) Propose a two-round node filtration method: Through the two-round filtration from the community evaluation and node evaluation method, the node search space is reduced, and the propagation coverage is narrowed.

2) Propose the regional metric of nodes: The local metric of nodes is formed by integrating and evaluating

nodes' neighborhood, radiation, and connectivity attributes.

3) Proposed the greedy algorithm based on submodularity property: Based on the submodularity property, the set of candidate nodes in two rounds is evaluated by the whole network metric, the set of Top-k nodes is found, and which can substantially reduce the time complexity.

The remaining sections in this paper are as follows: Section 2 addresses the review of related research works. Section 3 focuses on the greedy algorithm based on the two-round filtration metric in this paper. Section 4 shows the experimental comparison and result analysis. The final section presents the related conclusions and prospects.

2 Related Work

In the early research process, node degree became the preferred influence node criterion in terms of network structure topology, and it was believed that nodes that might be in the central position in the network or have specific linking properties tend to bring better influence, such as node degree, node centrality, and so on. However, the generative characteristics of scale-free networks determine that such nodes tend to be linked together preferentially, leading to more extensive duplicate coverage of influence propagation.

With the deepening of further research, based on earlier sociological analysis and marketing-related studies, influence propagation models (independent cascade model and linear threshold model) for interactions between users are constructed to evaluate influence, which can get a whole network quantitative perspective by portraying the activation states between nodes. The current research is mainly divided into greedy algorithms and heuristic algorithms.

2.1 Introduction to The Progress of Greedy-Based Algorithms

Kempe [17] represented the influence maximization problem as a discrete optimization problem for the first time and obtained the maximized influence propagation by a greedy algorithm. On this basis, Leskovec [21] proposed the celfGreedy algorithm to reduce the number of Monte Carlo simulations by submodular characteristics, reducing the time complexity to a more significant extent. Still, because its search space is the nodes of the entire network topology, the computational performance is affected by the data set. Its worst-case time complexity is approximately equal to that of the original greedy algorithm. In response, Goyal [13] proposed the celfPlusGreedy algorithm to evaluate the influence gain of nodes by further reducing the number of Monte Carlo simulations. Still, the reduced time complexity is more limited.

Subsequently, researchers continued to optimize the algorisms from the topology; Chen [6] proposed the new greedy algorithm to improve the efficiency by pre-deleting edges, which was compared with the celfGreedy algorithm and found to be advantageous only during the first round of computation. Wang *et al.* [28] proposed the CGA and OASNET methods using a greedy algorithm and dynamic programming approach to find the seed nodes. However, the simulation scope is limited to within the community, which reduces the network-wide influence metric.

Later, researchers proposed optimization schemes with different perspectives. Borgs *et al.* [4] proposed a hypergraph-based influence propagation estimation method, which still needs the validation of scene data. Cohen *et al.* [8] proposed to reduce the time complexity by selecting the node with the highest information gain for every round. Laya *et al.* [2]proposed a fuzzy propagation model to deal with the influence maximization (IM) problem. Yang *et al.* [29]proposed an exchange improvement algorithm to improve further the quality of the solution to the non-submodular influence maximization problem. Jie *et al.* [24]proposed a novel influence maximization algorithm of node avoidance based on user interest. Tang *et al.* [23] performed influence evaluation by measuring the lowest boundary of the propagation scope, and the running time was better than the CELF++ algorithm. The running time is better than the CELF++ algorithm. Wang *et al.* [27] proposed the IV-Greedy algorithm based on the multi-path asynchronous threshold model MAT, which can achieve better experimental results on the dataset. Zhou *et al.* [30] reduced the number of Monte Carlo simulations for influence calculation by constructing an upper bound function for the greedy strategy. The experimental results showed that when the size of the seed node set is small, the time complexity is better than the CELF algorithm.

2.2 Introduction to The Progress of Heuristic-Based Algorithms

To further improve the scalability of influence maximization algorithms and better apply them to large-scale social networks, researchers have also proposed some heuristic algorithms, such as Median centrality [3] and k-core [20], etc. Regarding topology, Chen *et al.* [6] proposed the DegreeDiscount method based on the first-order neighborhood influence of nodes, which works better experimentally when the propagation probability is small. Subsequently, Chen *et al.* proposed the LDAG [7] method to select seed nodes by updating the local topology to improve scalability. Still, the experimental results are easily affected by the network topology [15]. Cordasco *et al.* [9] proposed an efficient heuristic for the network structure of the tree, annular graphs, and complete graphs algorithm. They extended it to conduct influence calculations in directed graph network structures [10].

Then propagation paths became the focus of research; for example, Kimura *et al.* [19] proposed SPM/SP1M

method based on the shortest path, and Narayanam *et al.* [22] proposed Shapley value-based method, but both algorithms are weak in scalability. Goyal *et al.* [12] offered a way to find the shortest path from the node adjacency region. Galhotra *et al.* [11] proposed a heuristic algorithm based on adjacency paths that reduce the memory overhead compared to the CELF++ algorithm.

Around the relational perspective among nodes, Agha *et al.* [18] studied variable propagation probabilities based on node heterogeneity. They proposed an optimization model that simultaneously constrains the seed set and propagation scope. Wang *et al.* [26] argued for enhancing the consideration of group influence on nodes in multi-relational social networks. Chen *et al.* [5] used reinforcement learning based on the Markov decision process to model the influence problem. Later, the research perspective was gradually expanded, Jiang *et al.* [14] proposed a simulated annealing algorithm to optimize the influence problem; Jung *et al.* [16] performed incremental influence measurement on seed nodes, which can effectively reduce memory overhead and running time.

3 The Proposed Method of This Paper

3.1 Main Ideas

Some improved versions of the greedy algorithm by researchers have reduced the time complexity to some extent. However, the running time is relatively high and needs further improvement in real large-scale social networks. In this paper, we hope to provide a filtering mechanism to evaluate nodes from the community and topological perspectives, which can reduce the more extensive repeated coverage of the propagation scope of the preferentially linked nodes.

3.2 Variable Representation

A social network is modeled using an undirected graph G= (V, E), where node v represents the users in the network and edge e represents the association between users. Table **??** lists the important variables used in this paper; In this paper, S is used as the set of nodes selected to maximize its influence propagation, which also becomes the seed set. $simCas(S)$ represents a stochastic process based on the spread of the node set S's influence; therefore, the result of its influence is also a random set of nodes. The algorithm in this paper uses the graph G and the number k as input to generate a seed set S. The aim is to maximize the influence of other nodes based on the selected seed set.

3.3 Node Evaluation

Community division will help us to filtrate meaningful and dispersed communities, which can avoid repeated coverage of the propagation scope due to the preferential linking of nodes. Hence, the research in this paper involves the work related to community division, and to improve the performance further, this paper adopts Raghavan's [25] label propagation method, which can be achieved in linear time, as the method of community calculation in this paper.

There is some difficulty in evaluating the global attribute metric values of nodes in the whole network, which will consume a lot of running time, so we want to provide a fine-grained method to measure the attributes of nodes. In this paper, we consider candidate node benchmark metrics (BM) by the following three factors: node's adjacency attribute Lv, node's radiation attribute Rv, and node's connectivity attribute Cv.

$$BM(v) = \frac{Lv + Rv}{2} * Cv \tag{1}$$

The variable is described as follows:

- Dv: The node's degree, reflecting the node's number of neighbors.
- Lv: The neighboring metrics of a node, i.e., the ratio of the node's degree to the sum of its neighbor's degree, reflects the strength of the node's influence on neighboring nodes;
- Rv: The radiation metrics of a node, i.e., the ratio of the sum of the node's neighboring degrees to the sum of the community nodes, reflects the radiation strength of the node in the region;
- Cv: The connectivity metrics of a node, i.e., the ratio of the node's betweenness centrality and the sum of the node's betweenness centrality in the community, reflects the node's connectivity strength in the region.

$$Lv = \frac{Dv}{\sum_{W \in N(v)} D(w)} \tag{2}$$

$$Rv = \frac{\sum_{W \in N(v)} D(w)}{\sum_{W \in Com(v)} D(w)} \tag{3}$$

$$Cv = \frac{Bet(v)}{\sum_{W \in Com(v)} Bet(w)} \tag{4}$$

where b_{vw} represents whether node v is connected to node w, 1 if connected, and 0 otherwise. Dv represents the degree of node w, $N(v)$ represents the set of neighboring nodes of node v, $Com(v)$ represents the set of community nodes of node v, and $Bet(v)$ represents the betweenness centrality of node v.

Algorithm 1 The Two Rounds of Filtration Metric (TRFM) Algorithm

Input: Graph G, the amount of seeds k
Output: Top-k vertices

1: initialize $S = \emptyset, S_G = \emptyset$
2: community partition and get z candidate communities: $C_1, C_2, \ldots, C_z$
3: **for** $i = 1$ to z **do**
4: in community C_i, compute the local value based on Equation (2)
5: in community C_i, compute the radiation value based on Equation (3)
6: in community C_i, compute the connection value based on Equation (4)
7: compute benchmark value bv based on Equation (1), sort the candidate vertex and continually add to the set S_G
8: **end for**
9: **for** each vertex $v \in S_G \backslash S$ **do**
10: $MG_v = 0$
11: **for** $i = 1$ to R **do**
12: $MG_v += |SimCas(S \cup \{v\})|$
13: **end for**
14: $MG_v = MG_v / R$
15: store the vertex v with MG_v into the *Queue* Q
16: **end for**
17: sort the *Queue* Q in the descending order
18: $S = S \cup \{$first vertex in $Q\}$ and remove first vertex from Q
19: **for** $i = 2$ to k **do**
20: **while** true **do**
21: $vf =$first vertex, $vs =$second vertex in Q
22: **if** vs has not be evaluated in current round **then**
23: **if** MG_{vf} in current round $>= MG_{vs}$ in **previous / current** round **then**
24: $S = S \cup \{vf\}$ and remove vf from Q
25: **break**
26: **else**
27: insert the vf into the Q based on its marginal gain MG_{vf}
28: **end if**
29: **end if**
30: **end while**
31: **end for**
32: **return** S

3.4 Metrics and Algorithm Execution

3.4.1 Regional Metrics for Two Rounds

In this paper, we propose Two Rounds of Region Metric (TRFM); we hope to reduce the time complexity by searching the range at the whole network level, and how to locate the nodes' candidate solutions becomes the key. In the first round, we select some scaled subcommunities as candidate communities through community partitioning, which reduces the influence of repeated propagation due to the nodes are often linked together preferentially in the scale-free network; in the second round, we calculate the local attribute metrics through candidate communities to select some nodes with a higher ranking of benchmark metric to join the candidate node set continuously, and then always reduce the search scope to reduce time complexity.

Based on the "diminishing returns" property of the submodular function, the Marginal Gain obtained by adding a node v to the set S cannot be smaller than the marginal gain obtained by adding the same element v to the parent set T of S, denoted as $f(S \cup \{v\}) - f(S) >= f(T \cup \{v\}) - f(T)$. Based on the property of "diminishing returns," the number of evaluations is reduced, and the computational performance is improved by comparing the marginal gain value of the current round with the previous round. The time complexity is $O(N)$ in the optimal case and $O(KNRM)$ in the worst case. Therefore, in this paper, the algorithm also introduces the idea of submodular characteristics [21] to reduce the number of Monte Carlo simulations, which can be in the algorithm in two steps:

Find the first seed node: In the first round of calculation, the influence gain value is calculated for the set of filtered nodes and stored in the queue in reverse order, and the first node of the line is the first seed node we found, and then the node is removed at the head of the queue.

Find the remaining set of $k-1$ seed nodes: Continue to evaluate the marginal gain of nodes in each round and update the queue by comparing the influence gain of the first node in the current round with the influence gain of the second node in the previous round, if the gain value of the first node is more significant, we select the first node as the seed node. Otherwise, we insert the node into the corresponding position in the queue according to its influence gain value. Then we iterate the comparison of influence gain and update the node queue until the remaining $k-1$ seed nodes are found.

3.4.2 The Execution Process of The Algorithm

The algorithm in this paper is shown in Algorithm 1. Line 2 performs community division. Lines 4, 5, and 6 calculate the nodes' neighboring metrics, radiative metrics, and connectivity metrics in the current community. The baseline attributes of the nodes in the current community are calculated in line 7, sorted, and added to the candidate node set. In lines 9-18, the evaluation of influence gain is computed for each node in the candidate node-set, and the 1st seed node is found. In lines 19-29, the evaluation calculation of the influence gain values of the nodes stored in the queue is iterated until the $k-1th$ seed node is found.

Complexity analysis: Line 2 community division consumes $O(M)$. Lines 3-8 consume $O(MCNC)$. Lines 9-18 consume $O(zRNC)$, and lines 18-32 compute statistically for the remaining $k-1$ node sets with optimal time complexity of $O(NC)$ and worst time complexity of

$O(kzRNC)$. Overall, the optimal time complexity is thus $= O(M) + O(MCNC) + O(zRNC) + O(k) = O(zRNC)$ and the worst time complexity is $= O(M)+O(MCNC)+O(zRNC)+O(kzRNC) = O(kzRNC)$.

4 Experimental Analysis

We conduct our experiments on publicly available datasets and compare them with current heuristics and greedy algorithms to verify the effectiveness of the proposed method in two aspects: the range of influence and the running time.

4.1 Experimental Setup

Operating system: Windows 10, processor: Intel (R) i5 1.8GHz, memory: 32G.

4.1.1 Experimental Data Set

The experimental data were obtained from the dataset of Arxiv, a paper collaboration network [1], where a node represents that the user published a paper and an edge represents those two users co-authored the paper. The dataset is as follows:

- Dblp: DBLP academic paper collaboration dataset, where the number of nodes is about 14,485 and the number of edges is 37,026.
- GrQc: A collaborative dataset of papers in general relativity and quantum cosmology, where the number of nodes is about 5,242 and the number of edges is 14,485.
- Hep: A combined dataset of articles in high energy physics, where the number of nodes is about 15,233 and the number of advantages is 31,380.
- Phy: A collaborative dataset of papers in the field of physics, where the number of nodes is about 14,997 and the number of edges is 57,866.

Table 2: Statistics of four real-world networks

DataSet	#Vertice	#Edge
Dblp	14,485	37,026
GrQc	5,242	14,485
Hep	15,233	31,380
Phy	14,997	57,866

We extracted the structure of four types of paper collaboration networks from the arXiv paper literature, each node in the network represents an author, and each edge represents the existence of two authors collaborating on a paper. The structure of the four types of networks is shown in Table 2.

4.1.2 Experimental Model

The goal of our algorithm is to perform validation in the Independent Cascade (IC) model, so we use the following two models to generate non-uniform information propagation probabilities:

- UIC: i.e., Uniform Independent Cascade Model (UIC) On each edge (v, w), we uniformly choose the probability at random in the set 0.1,0.01, which corresponds to the level of influence;
- WIC:i.e., Weighted Independent Cascade Model (WIC), in which the probability of influence on each edge (v, w) is $1/d_w$, where d_w is the number of degrees of entry of node w. However, the model can generate asymmetric, non-uniform propagation probabilities even if the original graph is undirected.

4.1.3 Comparison Method

- Random: As a basic comparison algorithm, k nodes are randomly selected in graph G. The graph is referred to as Rand;
- MaxDegree: as a comparison algorithm, one that selects k nodes with a maximum degree based on their topology, abbreviated as HeuMD in the figure, with time complexity of $O(N)$;
- DegreeDiscount: proposed in the literature [6], a degree discount heuristic, abbreviated as HeuDD in the figure, with a running time of $O(klogN + M)$;
- BetweennessCentrality: proposed in [3], a heuristic based on the betweenness centrality of nodes, abbreviated as HeuBet in the figure, and the optimal running time is $O(MN)$;
- CelfGreedy: proposed in [13] a greedy algorithm optimization scheme based on submodular properties, abbreviated as Celf in the figure, and the optimal running time is $O(RMN)$.
- The method of this paper: The greedy algorithm based on two rounds of filtration proposed in this paper, referred to as TRFM in the figure, has an optimal running time of $O(zRNC)$;

4.1.4 Evaluation Indicators

Influence range: To better obtain the influence propagation scope of these algorithms, for each node seed set, we got a more stable propagation scope by Monte Carlo simulations 2000 times on UIC and WIC models, respectively. The larger the influence propagation value, the better the algorithms perform.

Runtime: We also compare the runtime of influence propagation for the set of k=50 nodes. The smaller the running time, the better the algorithm performs.

4.2 Analysis of Results

4.2.1 Experimental Results Demonstration

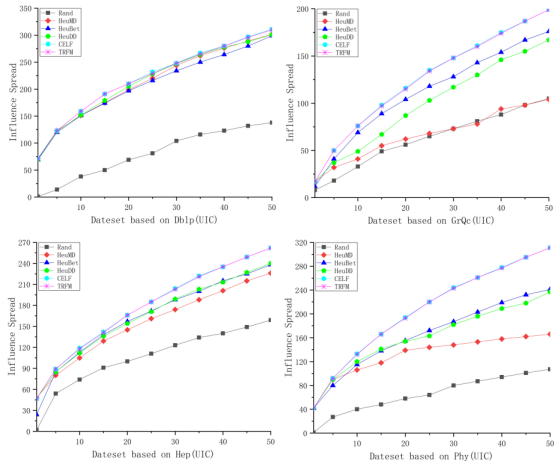


Figure 1: Experimental comparison of algorithms based on different data sets under the UIC model

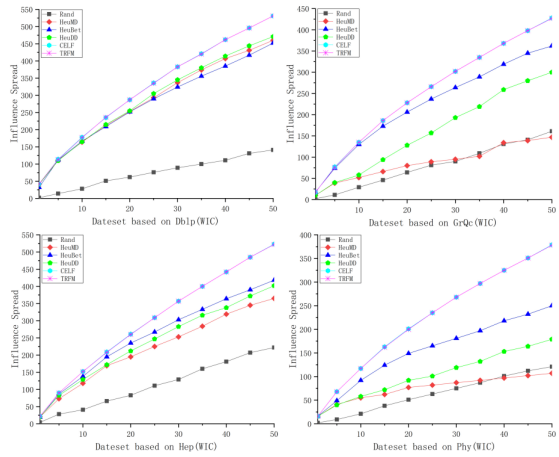


Figure 2: Experimental comparison of algorithms based on different data sets under the WIC model

Influence propagation: Figure 1 (based on the UIC model) and Figure 2 (based on the WIC model) show the scope of influence propagation based on various algorithms on four different datasets. For straightforward reading, in all the influence propagation legends, the legends rank the algorithms from the direction down according to the scope of influence propagation (k=50). Figure 3 shows the running time comparison of the Celf algorithm and TRFM algorithm when the k=50 seed set.

4.2.2 Analysis of Propagation Scope

First, the influence propagation scope based on the UIC model and the WIC model is shown in Figure 1 and Figure 2, where the CELF algorithm shown in cyan as the optimal coverage guarantees an approximate optimal solution over the four data sets, and we used as the target for the benchmark test and marked as 100%.

Secondly, the random strategy shown in black shows the practical significance of the strategy selection that must be employed. The maximum degree strategy is shown in red; However, it offers a specific propagation scope on the dblp dataset and Hep dataset; due to the generative characteristics of the scale-free network, the nodes with more significant degrees are often linked together preferentially, which quickly causes repeated coverage of the propagation scope and cancels out part of the propagation effect; therefore it has to perform poorly on the GrQc dataset, Phy dataset, even inferior to the random strategy.

Then, the betweenness centrality shown in blue and the degree discounting algorithm shown in green, either based on the UIC or WIC models, reflect a better and more stable propagation scope as heuristic strategies. However, there is still some distance to improve the propagation scope compared to the optimal one.

Finally, the CELF algorithm under the greedy strategy shown in cyan, and the TRFM algorithm based on two rounds of filtration proposed in this paper shown in pink, maintain excellent propagation scope on the four data sets, and the performance can remain stable on both the UIC model and the WIC model. The TRFM algorithm proposed in this paper can significantly approximate the optimal solution of the CELF algorithm. Benchmark comparisons of the propagation scope of different algorithms are shown in Table 3 and Table 4.

4.2.3 Running Time Analysis

Current algorithms: As shown in Figure 3 and Figure 4, the CELF algorithm corresponds to the cyan histogram, and the TRFM algorithm corresponds to the pink histogram. From the comparison of the data based on the UIC model (as shown in Figure 3), compared with CELF, the TRFM algorithm saves 96%, 89.3%, 92.9%, and 93.5% of the running time; from the comparison of the data based on WIC model (as shown in Figure 4), compared with CELF, TRFM algorithm saves 97.1%, 91.5%, 92.2%, 95.8% of the running time; the TRFM algorithm proposed in this paper substantially reduces the running time and improves the running time by about 10 times to 30 times compared with the CELF algorithm of greedy strategy and maintains good stability.

Experiments on four publicly available datasets show that the TRFM method proposed in this paper can obtain a propagation scope that can approach the optimal solution of CELF, whether based on the UIC model or the WIC model, and, at the same time, substantially reduces the computation time and maintains good stability.

Table 3: Comparison of the propagation scope of different algorithms based on the UIC model

Data\Algorithm	Rand	HeuMD	HeuMD	HeuDD	TRFM	CRLF
Dblp	44.40%	97.10%	96.10%	96.80%	99.70%	100%
GrQc	52.80%	52.30%	88.40%	83.90%	100%	100%
Hep	60.70%	86.30%	90.80%	91.60%	100%	100%
Phy	34.40%	52.40%	77.50%	76.20%	100%	100%

Table 4: Comparison of the propagation scope of different algorithms based on the WIC model

Data\Algorithm	Rand	HeuMD	HeuMD	HeuDD	TRFM	CRLF
Dblp	26.60%	86.60%	85.10%	88.70%	100%	100%
GrQc	37.60%	34.30%	84.60%	70.10%	99.80%	100%
Hep	42.40%	69.80%	79.90%	76.90%	99.80%	100%
Phy	31.90%	28.20%	66.00%	47.20%	99.70%	100%

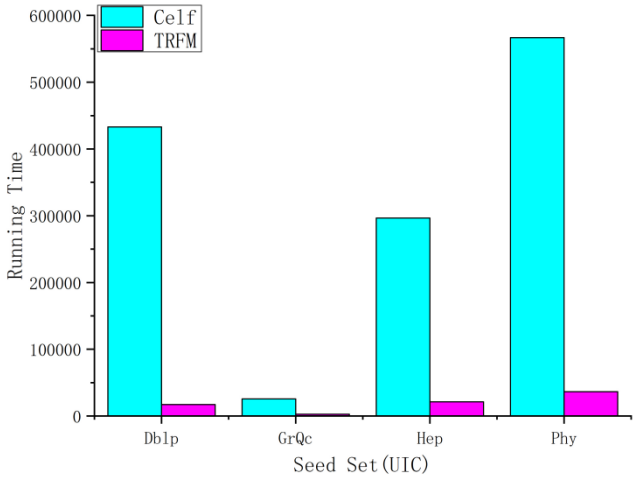


Figure 3: Comparison of the algorithm running time for different data sets based on the UIC model

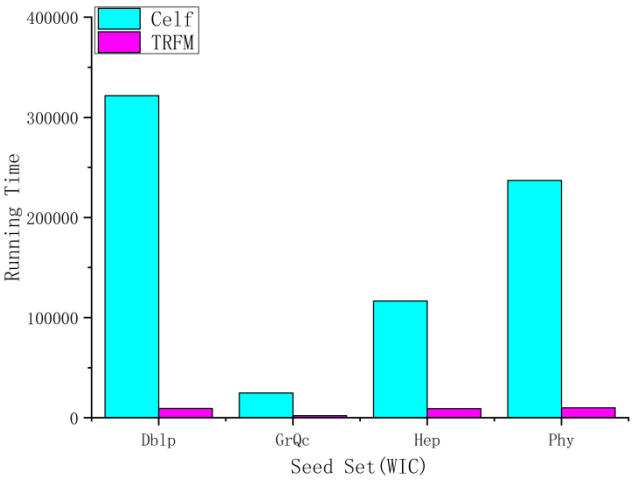


Figure 4: Comparison of the algorithm running time for different data sets based on the WIC model

5 Conclusion and Outlook

As the problem of maximizing the influence of social networks is a hot research topic, this paper proposes the TRFM algorithm with two rounds of filtration metrics, which significantly reduces the time complexity compared with current methods and approaches the optimal propagation scope on four different datasets and has stable performance. Nevertheless, there is still much room for future research work; for example, solving the influence maximization problem based on "topic semantic modeling," "large-scale social networks," "dynamic online computing," etc. may provide ideas for the application of social networks.

Acknowledgments

The research of this paper was supported by a grant from the National Social Science Foundation of China under the General Project (19BXW107). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] "High energy physics [eb/ol]," in *Available online: http://www.arxiv.org/archive/hep*, accessed on 2003.

[2] L. Aliahmadipour and E. Valipour, "A new fuzzy propagation model for influence maximization in social networks," *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, vol. 30, pp. 279–292, 2022.

[3] M. Barthelemy, "Betweenness centrality in large complex networks," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 163–168, 2004.

[4] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time,"

in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SIAM)*, pp. 946–957, 2014.

[5] M. Chen, Q. P. Zheng, V. Boginski, and E. L. Pasiliao, "Influence maximization in social media networks concerning dynamic user behaviors via reinforcement learning," *Comput Soc Netw*, pp. 8–9, 2021.

[6] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (ACM SIGKDD)*, 2009.

[7] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM)*, pp. 88–97, 2010.

[8] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Sketch-based influence maximization and computation: Scaling up with guarantees," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (ACM)*, pp. 629–638, 2014.

[9] G. Cordasco, L. Gargano, and A. A. Rescigno, "Active spreading in networks," in *Proceedings of the ICTCS*, pp. 149–162, 2016.

[10] G. Cordasco, L. Gargano, and A. A. Rescigno, "Influence propagation over large scale social networks," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 ,ACM*, pp. 1531–1538, 2015.

[11] S. Galhotra, A. Arora, S. Virinchi, and S. Roy, "ASIM: A scalable algorithm for influence maximization under the independent cascade model," in *Proceedings of the 24th International Conference on World Wide Web, ACM*, pp. 35–36, 2015.

[12] A. Goyal, W. Lu, and L. V. Lakshmanan, "Simpath: An efficient algorithm for influence maximization under the linear threshold model," in *Proceedings of the 2011 IEEE 11th International Conference on Data Mining (ICDM)*, 2011.

[13] A. Goyal, W. Lu, and L. Lakshmanan, "Celf++: optimizing the greedy algorithm for influence maximization in social networks," in *Proceedings of the 20th international conference companion on World wide web, ACM*, pp. 47–48, 2011.

[14] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, and K. Xie, "Simulated annealing based influence maximization in social networks," in *Proceedings of the 35th AAAI Conference on Artificial Intelligencet (AAAI)*, 2011.

[15] K. Jung, W. Heo, and W. Chen, "Irie: Scalable and robust influence maximization in social networks," in *arXiv, 2011, preprint arXiv:1111.4795*, 2011.

[16] K. Jung, W. Heo, and W. Chen, "Irie: Scalable and robust influence maximization in social networks," in *2012 IEEE 12th International Conference on Data Mining (ICDM)*, pp. 918–923, 2012.

[17] D. Kempe, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (ACM SIGKDD)*, 2003.

[18] A. Kermani, R. Ghesmati, and M. S. Pishvaee, "A robust optimization model for influence maximization in social network with heterogeneous nodes," *Comput Soc Netw*, pp. 8–17, 2021.

[19] M. Kimura and K. Saito, "Tractable models for information diffusion in social networks," *Knowledge Discovery in Databases: PKDD 2006*, pp. Springer: 259–271, 2006.

[20] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, "Identification of influential spreaders in complex networks," *Nature Physics*, vol. 6, no. 11, pp. 888–893, 2010.

[21] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (ACM SIGKDD)*, 2007.

[22] R. Narayanam and Y. Narahari, "A shapley value-based approach to discover influential nodes in social networks," *IEEE Transactions on Automation Science and Engineering(99)*, pp. 1–18, 2010.

[23] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data (ACM SIGMOD)*, pp. 75–86, 2014.

[24] J. Tong, L. Shi, L. Liu, J. Panneerselvam, and Z. Han, "A novel influence maximization algorithm for a competitive environment based on social media data analytics," *Big Data Mining and Analytics*, vol. 5, no. 2, pp. 130–139, 2022.

[25] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detec community structures in large-scale networks," *Physical Review*, vol. 76, no. 3, p. 036106, 2007.

[26] W. Wang, H. Yang, Y. Lu, Y. Zou, X. Zhang, S. Guo, and L. Lin, "Influence maximization in multi-relational social networks," in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 4193–4202, 2021.

[27] W. Wang and W. N. Street, "Modeling and maximizing influence diffusion in social networks for viral marketing," *Applied Network Science*, pp. 3–6, 2018.

[28] Y. Wang, G. Cong, G. Song, and K. Xie, "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (ACM SIGKDD)*, 2010.

[29] W. Yang, Y. Zhang, and D. Z. Du, "Influence maximization problem: Properties and algorithms," *J. Comb. Optim.*, vol. 40, p. 907–928, nov 2020.

[30] C. Zhou, P. Zhang, W. Zang, and L. Guo, "On the upper bounds of spread for greedy algorithms in social network influence maximization," *2015 IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 10, pp. 2770–2783, 2015.

Biography

Yang Li is currently a senior engineer at the State Information Center. He graduated with a doctor's degree. His main research interests are social networks, information security and data mining.

Zhiqiang Wang received the Ph.D. degree in information security from Xidian University. He is currently an Assistant Professor with the Department of Computer Science and Technology. His research interests include system security and network security.

Verifiable Encrypted Speech Retrieval Method Based on Blockchain and C-BiGRU

Fang-Peng Li[1], Qiu-Yu Zhang[1], Ying-Jie Hu[1], and Yi-Bo Huang[2]
(Corresponding author: Qiu-yu Zhang)

School of Computer and Communication, Lanzhou University of Technology[1]
No. 36 Peng Jia-ping Road, Lanzhou, Gansu 730050, China
College of Physics and Electronic Engineering, Northwest Normal University[2]
967 Anning E Rd, Anning District, Lanzhou 730070, Gansu, China
Email: zhangqy@lut.edu.cn, lllfape@163.com, modaoshiyan@163.com, huang_yibo@nwnu.edu.cn
(Received May 10, 2023; Revised and Accepted Sept. 23, 2023; First Online Apr. 25, 2024)

Abstract

To solve the problems of privacy leakage and lack of verification of the retrieval results in the existing encrypted speech retrieval while improving the performance of the retrieval and the security, a verifiable encrypted speech retrieval method based on blockchain and Convolutional Neural Networks-Bidirectional Gate Recurrent Unit (C-BiGRU) was proposed. Firstly, the speech is encrypted using the AES-128 algorithm and uploaded to the cloud servers. Secondly, the low-level features of speech are selected and fused into new features and input into the designed C-BiGRU model for training to extract the deep features with more robustness and generalization ability. Finally, the constructed deep hash codes are used as searchable encryption keywords and smart contracts to store encrypted file hashes and the corresponding encrypted indexes. It incorporates the blockchain to enable retrieval on a smart contract using a search token and verifies the integrity of the results using the SHA-256-based Hash Message Authentication Code (HMAC-SHA256) algorithm. Experimental results show that the proposed method can effectively prevent plaintext leakage, enhancing the accuracy and security of encrypted speech retrieval and protecting data privacy. The blockchain-based integrity verification ensures fairness and practicality while proving to be secure.

Keywords: Deep Hashing; Feature Fusion; Result Verification; Searchable Encryption; Verifiable Encrypted Speech Retrieval

1 Introduction

With the rapid development of mobile input devices, Cloud storage is becoming more and more popular to control the cost of storing massive speech and to facilitate daily work [14]. However, cloud servers are curious and semi-trustworthy. Encrypted speech protects privacy to some extent but loses its plaintext features and relevance, and the owner has no direct control and access to the stored speech [4], which prevents efficient retrieval and makes users suspicious of the retrieval results [12]. Therefore, it is challenging to quickly retrieve encrypted speech and verify that the results are complete while ensuring secure data storage conditions.

Currently, encrypted speech retrieval schemes use various easily distinguishable speech features, such as perceptual hashing, biohashing, deep hashing, and audio fingerprinting, with relatively stable retrieval efficiency and accuracy. However, the robustness of the features and the retrieval accuracy still need to be improved. The new features after feature fusion can effectively describe and distinguish different speech information. Perceptual hashing retrieval uses a single feature, which leads to insufficient semantic description, while the deep hashing retrieval also has insufficient generalization ability of the input features. Therefore, deep learning obtains deeper semantic features with new features after feature fusion, and efficiency of speech retrieval is improved by reducing dimensionality through feature hashing and avoiding exhaustive search. Data stored in the cloud is often accessed, leak, or modified by malicious servers without authorization. To protect data privacy, users usually encrypt speech before uploading. Therefore, it is urgent to efficiently retrieve encrypted speech and verify the integrity of the returned speech. Currently, the searchable encryption algorithm [10] allows data retrieval without compromising privacy. Nevertheless, the search results are often incorrect due to malicious acts of cloud servers, so users need to verify the correctness of the retrieval results. To avoid users falsifying the validation results due to financial interests, which leads to a lack of fairness in validation, data integrity can be ensured based on verifiable algorithms using the tamper-proof nature of blockchain [3]. In addition, smart contracts [24], as self-

executing contracts without the involvement of a centralized institution. Therefore, the combination of blockchain and smart contracts is more suitable for verifying results, ensuring the verification's reliability and fairness, and providing strong support for users to verify the originality of the data.

In order to prevent malicious cloud servers from violating privacy or providing false retrieval results and to improve the accuracy of retrieving encrypted speech and verify results, a verifiable encrypted speech retrieval method based on blockchain and C-BiGRU is proposed. The main innovations of this paper can be summarized as follows:

1) A deep learning-based feature fusion encrypted speech retrieval scheme is designed. The method extracts the MFCC and Fbank features of speech and, after Principal Components Analysis (PCA), reduces the dimensionality into MFCC-Fbank features and uses the C-BiGRU model to extract deep features that have stronger robustness and improve retrieval accuracy;

2) A new deep hash construction method is designed to randomly assign two equal-sized subsets to the dataset based on double k-means and obtain compact deep hash codes using different threshold settings, which avoids fuzzy centroid assignment, reduces retrieval errors and improves retrieval performance;

3) Verifiability of the integrity of the retrieval results is achieved. The HMAC-SHA256 algorithm is combined with the smart contract to complete the verification of the retrieval results. Due to the tamper-proof nature of the blockchain and the collision resistance of the HMAC-SHA256 algorithm, the user computation overhead is reduced, and the fairness of the verification is ensured.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 details the proposed system model and the concrete implementation scheme. Section 4 presents the experimental results and analysis and compares the performance with existing encrypted speech retrieval schemes. Section 5 summarizes the work in this paper.

2 Related Works

Speech is an important medium in daily life and is characterized by high frequency and a high degree of confidentiality. In recent years, domestic and international researchers have paid more and more attention to the technology for retrieving encrypted speech and proposed various methods for retrieving encrypted speech. These mainly include perceptual hash based retrieval methods, content based retrieval methods, searchable encryption based retrieval methods, and deep learning based retrieval methods. For example, Zalkow *et al.* [20] proposed a speech retrieval scheme based on CTC (Connectionist Temporal Classification), which computes a matching function based on SDTW (Soft-Dynamic Time Warping), which adds matching process based on SDTW, offers a fixed length sliding window and more flexible retrieval, but the feature robustness is not sufficient. Zhang *et al.* [22] proposed an audio fingerprinting method based on features downscaling and features fusion, extracting the original speech MFCC, LPCC (Linear Prediction Cepstrum Coefficient) features, based on information entropy feature dimensionality reduction method to construct audio fingerprints with good robustness, but the retrieval efficiency is slightly insufficient. Huang *et al.* [5] constructed a hash sequence and encrypted them by extracting speech features through 2D-Gabor transform and PCA dimension reduction, with high security and retrieval efficiency but the computational complexity is too large. Li *et al.* [6] proposed a low-dimensional audio fingerprint extraction method based on locally linear embedding and an efficient hierarchical retrieval method to construct a hash table using single-frame audio fingerprint hash, reducing retrieval complexity and narrowing the retrieval range. Yang *et al.* [18] made the decision function more discriminative, the number of parameters reduced, and the depth feature extraction better by reducing the size of convolutional layers and increasing the number. Cai *et al.* [1] used a convolutional neural network to extract features. A new loss function is designed to extract unique features with stronger generalization capability by combining with a Siamese network and hashing algorithm to extract unique features with stronger generalization ability, effectively improving retrieval accuracy. However, the same effect cannot be achieved in cryptographic conditions. Petcharat *et al.* [11] proposed an unsupervised learning hashing method based on deep learning audio retrieval, which performs unsupervised learning through neural networks and uses hash binary codes to improve retrieval efficiency and accuracy. Still, the efficiency of the retrieval algorithm is not enough. Zhang *et al.* [21] proposed a speech retrieval scheme based on multi-user searchable encryption with an LSTM (Long Short Term Memory) network to extract speech deep features as searchable encryption keywords to achieve efficient retrieval of encrypted speech and avoid privacy leakage during retrieval. Li *et al.* [7] proposed a multi-user searchable encrypted speech retrieval scheme that used the Diffie-Hellman algorithm to ensure data security for multi-user retrieval and improved retrieval accuracy through deep learning. However, the retrieval results lacked integrity verification.

To efficiently retrieve multimedia information and verify the integrity of retrieval results, Liu *et al.* [9] proposed a verifiable dynamic encryption and retrieval scheme, building inverted index and verifiable proofs and using RSA accumulators to generate proofs of search results. Tong *et al.* [13] proposed a verifiable keyword retrieval scheme with adaptive security, using a twin Bloom filter to store keywords to improve retrieval efficiency, com-

bined with Merkle Tree structure and the adapted multi-set accumulator to verify the data. Li *et al.* [8] proposed an efficient and verifiable fuzzy keyword retrieval scheme with LSH (Locality Sensitive Hashing) to improve the retrieval accuracy and used homomorphic MAC and random challenge technique to verify the returned results. Ge *et al.* [2] proposed a searchable cryptographic authentication scheme based on symmetric-key with cumulative authentication labels that can be dynamically updated to support the efficient verification of dynamic data.

Blockchain is decentralized, verifiable, and tamper-evident. Efficient retrieval and verification of results in multimedia information retrieval schemes through verifiable algorithms. Yu *et al.* [19] improved the Merkle Tree structure and introduced the Bloom filter to crop the data query to achieve fast query location. Zheng *et al.* [23] proposed a blockchain based verifiable medical data retrieval scheme that uses verifiable oblivious pseudo-random functions on the blockchain to generate proofs to ensure data transparency and privacy security. Yang *et al.* [17] proposed a blockchain based multi-keyword verifiable retrieval method that utilizes smart contracts to verify the integrity of results and achieve fair verifiability. Xu *et al.* [16] proposed a blockchain based multi-keyword verifiable searchable symmetric encryption scheme, which combines bitmap index with hash functions to achieve lightweight retrieval and improve retrieval and verification efficiency.

In summary, existing content based encrypted speech retrieval methods have received much attention regarding retrieval accuracy and integrity verification of the returned results. Therefore, this paper proposed a verifiable encrypted speech retrieval method based on blockchain and C-BiGRU based on this hot issue, which improves the accuracy of encrypted speech retrieval and verify the results in blockchain, protecting data privacy and verification fairness.

3 The Proposed Method

Figure 1 shows the system model of the verifiable encrypted speech retrieval method based on blockchain and C-BiGRU. This model consists of four entities, including the data owner (DO), the data user (DU), the blockchain (BC), and the cloud server (CS). The main tasks of the four entities are as follows:

DO: DO encrypts the speech and uploads it to CS storage, and sends the search key to DU when DU requests authorization for identity authentication is passed, and also uses the C-BiGRU model to extract features to build deep hash codes and create encrypted index tables, which are stored in the blockchain smart contract.

DU: DU first requests the DO to grant search permission, and when it needs to retrieve, extracts the keywords of the speech to be retrieved and generates a search token to send to the smart contract for matching and retrieval.

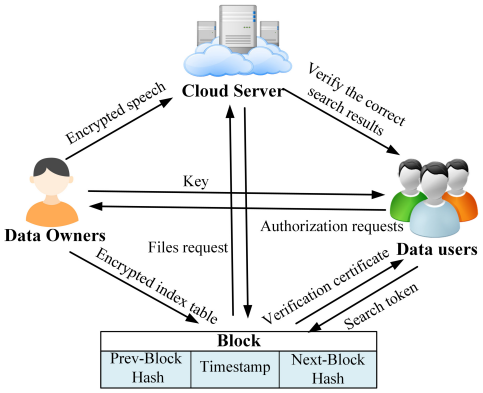


Figure 1: System model

BC: The smart contract in BC stores the encrypted index table for retrieval and requests the relevant speech files from CS based on the retrieval results. When verifying data integrity, the retrieved files are verified using the collision resistance of HMAC-SHA256 in the verify contract, and the verification proof is sent to DU.

CS: CS stores the encrypted speech uploaded by DO, and when DU performs the retrieval, sends the retrieval result to smart contract to calculate the hash value, and returns the correct retrieval result to DU.

3.1 Feature Fusion

MFCC is obtained by filtering the input signal from low to high frequencies through bandpass filter with discrete cosine transform, which has better robustness and recognition performance than other low-level features. Fbank extraction removes the discrete cosine transform in the last step of MFCC, which is relatively less computationally intensive and has higher feature correlation. Therefore, MFCC and Fbank features are extracted for fusion in this paper. The high-dimensional vector space of the new features after feature cascading is efficiently processed by the PCA method and the important feature information is preserved by dimensionality reduction. Figure 2 shows the processing flow of PCA features fusion. The specific steps are as follows:

Step 1: MFCC and Fbank features are extracted from the original speech and used for feature fusion.

Step 2: The extracted features are subjected to normalization operation. This paper uses the zero-mean normalization algorithm to map the original data to a distribution with a mean 0 and a standard deviation 1.

Step 3: PCA is performed on the high-dimensional features, which are transformed into a few significant

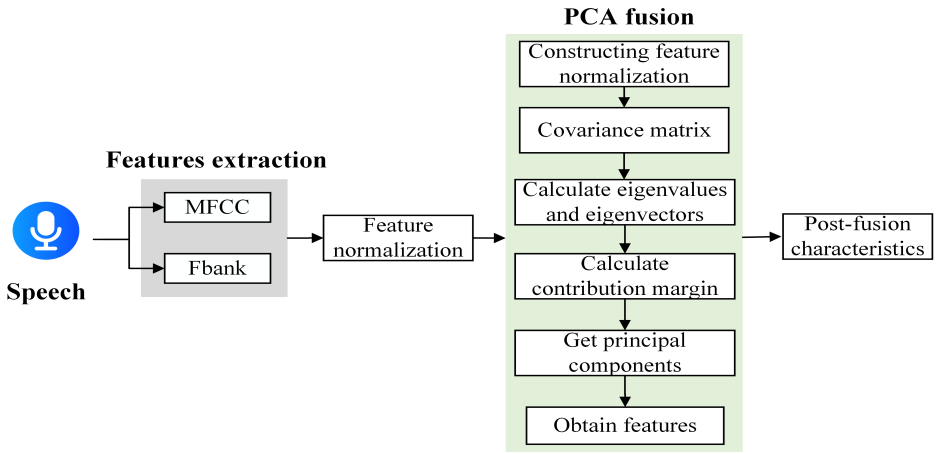


Figure 2: PCA features fusion processing flow

elements. Firstly, the standardized sample matrix $\boldsymbol{A_{m\times n}}$ is established, where p is the feature dimension, and n is the number of samples. The matrix as in Equation (1) is standardized to obtain the standardized matrix $\boldsymbol{X}$. The covariance matrix $\boldsymbol{R}$ of the standardized matrix $\boldsymbol{X}$ is calculated using Equation (2); then the eigenvalues λ and eigenvectors α of the covariance matrix $\boldsymbol{X}$ are calculated, and the eigenvalues are rearranged to obtain λ, and the contribution rate V_i of each principal component is calculated according to Equation (3); finally, the top k principal components are selected according to the cumulative contribution rate V_i, and the transformation matrix $\boldsymbol{E}$ is obtained using Equation (4), and the final principal component Y is calculated from Equation (5) and used as the last fusion feature.

$$x_{i,j} = \frac{a_{i,j} - \bar{a_i}}{s_i}, i = 1, 2, ..., n; j = 1, 2, ..., p. \tag{1}$$

$$R = \frac{X^T X}{n-1}$$

$$r_{i,j} = \frac{\sum_{j=1}^{p}\sum_{k=1}^{n} x_{j,k} \times x_{k,j}}{n-1}; i, j = 1, 2, ..., p; k = 1, 2, ..., n. \tag{2}$$

$$V_i = \frac{\lambda_i}{\sum_1^n \lambda_i}. \tag{3}$$

$$\begin{aligned} |R - \lambda E| &= \vec{0} \\ E &= [\overline{\alpha_1}, \overline{\alpha_2}, ..., \overline{\alpha_n},] \end{aligned}. \tag{4}$$

$$Y = E \times K. \tag{5}$$

3.2 Speech Secure Retrieval

Figure 3 shows the speech secure retrieval processing flow, which consists of three main modules: construction of encrypted speech library, deep hash, secure index table construction, and speech retrieval.

1) Construction of encrypted speech library. DO digitizes, frames, and matrix reorganizes the original speech signal into a 4×4 byte matrix, encrypts the bytes in the matrix using AES-128 encryption algorithm, obtains the encrypted speech set $C = \{c_1, c_2, ..., c_n\}$, and uploads to CS for storage.

2) DO extracts and fuses the MFCC and Fbank features in the original speech and the newly fused features are input to the C-BiGRU model to extract the deep features, and constructs the hash binary codes $H = \{h_1, h_2, ..., h_n\}$ as search keywords $W = \{w_1, w_2, ..., w_n\}$, and then use the generated key pair to encrypt the hash codes H, which corresponds to the encrypted speech one by one, and obtain the encrypted index table I, which is sent to the smart contract and CS for storage, respectively.

3) Speech retrieval. DU requests authorization from DO, and after the authentication is passed, DU gets the trusted authorization and search key. When DU retrieves, it performs features fusion, extracts depth features w, constructs hash code H_w, generates search token T_w, CS retrieves according to the T_w, sends the matching result to BC for verification, and sends the correct result to DU if the validation is passed.

3.3 Encrypted Speech Library and Secure Index Table Construction

To prevent malicious cloud servers from leaking or tampering with stored speech, this paper uses the AES-128 algorithm to encrypt the speech to be uploaded and constructs an encrypted speech library. Figure 4 shows the speech encryption processing flow.

The steps of speech encryption processing are as follows:

Step 1: Digitize the signal from the original speech, framing it according to the speech duration, sampling frequency, and speech signal value, and then the framed signal value is matrix reorganized into a 4×4 matrix of bytes.

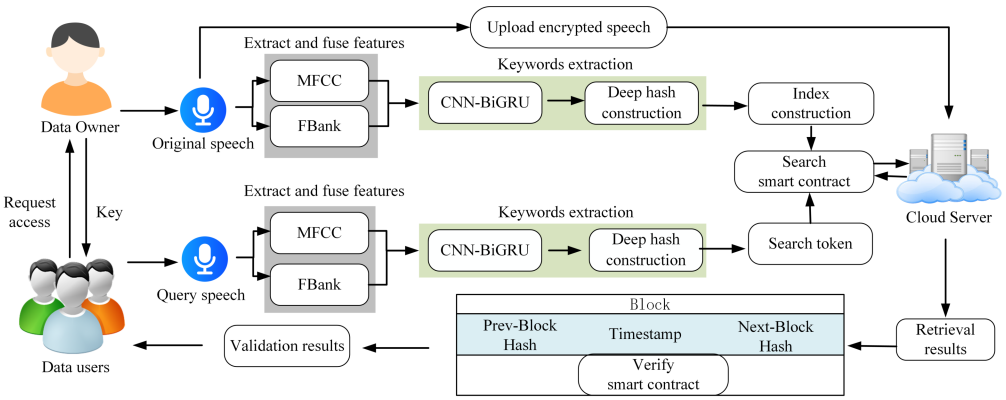


Figure 3: Speech security retrieval processing flow

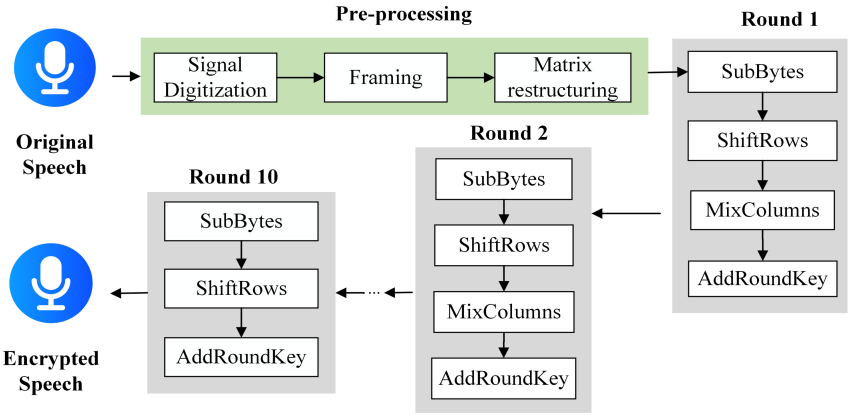


Figure 4: Speech security retrieval processing flow

Step 2: AES-128 algorithm encryption processing. Firstly, the byte matrix of 4×4 is subjected to the *SubBytes* transformation for obfuscation; then the *ShiftRows* transformation, so that the order of the bytes changes and the order of the bits in the bytes remain unchanged; then *MixColumns* transformation is performed, and bitwise XOR operation is applied between adjacent bytes; finally, the corresponding rounds of the *ExpandedKey* with matrix data are subjected to bitwise XOR operation.

Step 3: *AddRoundKey* transformation. The AES-128 algorithm consists of ten rounds of encryption. Expand the input key for ten rounds. At the last round, the *MixColumns* transformation was no longer performed. Finally, the encrypted speech is obtained and uploaded to CS for storage.

Using a secure index table I can significantly speed up data retrieval and is of reference value in verifying data integrity. The specific construction process is as follows:

Step 1: DO extracts the deep features by training the fused speech features with the C-BiGRU model and constructs the hash binary codes.

Step 2: DO takes the constructed hash binary codes as the speech keyword, symmetrically encrypts the hash codes by the generated public key pk, obtains the security index of speech, and uploads the encrypted speech file with the corresponding security index to be stored in the security index table I for subsequent DU retrieval.

3.4 C-BiGRU Model and Deep Hash Construction

Figure 5 shows the designed C-BiGRU model. The model mainly consists of convolutional layer, pooling layer, Bi-GRU, and fully connected layer.

As shown in Figure 5, after the training of the C-BiGRU model, the retrieval speed is slow due to the large data size and linear search. This paper uses a double k-means hash construction instead of traditional retrieval to boost the retrieval speed. By mapping the high-dimensional features to the low-dimensional space, then dividing them into two parts randomly, using different thresholds for comparison, generating hash codes separately, and then merging them, the problem of fuzzy assignment of k-means centroids or quantization errors is effectively solved, reducing the retrieval error and improving the retrieval accuracy. The specific process of constructing the deep hash binary code is as follows:

Step 1: The neurons in the hidden layer after convo-

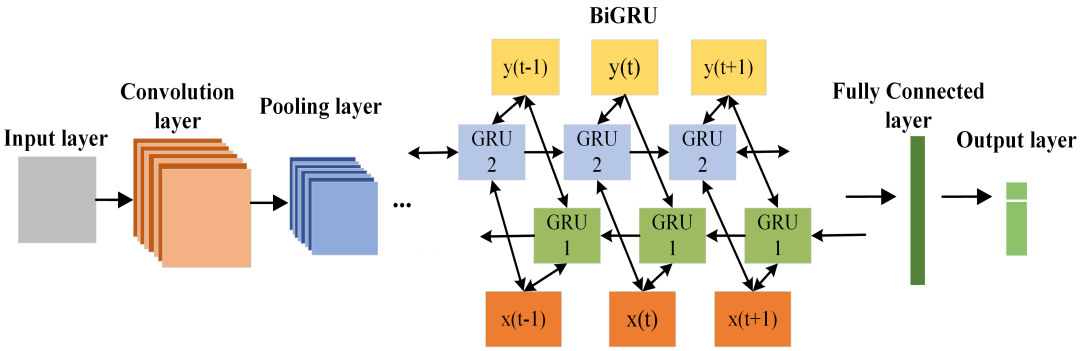


Figure 5: System model

lution, pooling, and BiGRU are activated by the **ReLU** function. The data from the hidden layer is fed into a fully connected layer containing 384 neurons, which is also activated by the **ReLU** function. This fully connected layer outputs a depth feature vector $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n\}$, where n is the number of nodes in the output fully connected layer.

Step 2: Hash construction of the feature vector $\boldsymbol{V}$ using dual k-means. Firstly, k centroids $P = \{p_1, p_2, ..., p_k\}$ are obtained according to Equation (6) for constructing and distributing deep hash binary codes, where each specific bit of the hash code is associated with a centroid. The training data is randomly divided into two groups, and a hash code is created by comparing each feature vector to a threshold.

$$v_{i,j} = \begin{cases} 1, ||x - p_j|| \leq d \\ 0, ||x - p_j|| \geq d \end{cases} \tag{6}$$

where $v_{i,j}$ is the j-th bit value of the i-th depth feature vector, and d is the evaluation threshold.

Step 3: The hash codes are constructed for the two training data sets according to Equation (7) and Equation (8), respectively, which may still be associated with a close centroid even if they are not associated with the corresponding centroid. Finally, the two generated hash codes are combined as the depth hash binary code of the speech data for subsequent retrieval use.

$$d = \frac{\sum_{j=1}^{k} ||x - P_j||}{k} \tag{7}$$

$$d = \begin{cases} \frac{||x_{\frac{i}{2}} - P_j||}{2}, i = 1, 3, 5, ..., 2n+1 \\ \frac{||x_{\frac{i}{2}} - P_j|| + ||x_{\frac{i+1}{2}} - P_j||}{2}, i = 2, 4, 6, ..., 2n \end{cases}. \tag{8}$$

where d_{2i+1} is the formula for calculating the arithmetic mean and d_{2i} is the formula for calculating the median.

3.5 Verifiable Searchable Encryption

The verifiable searchable encryption scheme consists of six probabilistic polynomial time algorithms: *Setup*, *Enc*, *TokenGen*, *Search*, *Verify*, *Dec*, among them.

1) $Setup(1^{\lambda}) \rightarrow (sk_1, sk_2)$: Given a security parameter λ, choose a pseudo-random function $\mathbf{F} : \{0,1\}^* \rightarrow \{0,1\}^a$, and choose a pseudo-random function $\mathbf{H} : \{0,1\}^* \rightarrow \{0,1\}^b$, which is used to compute to verify the data integrity. With the help of the security parameter λ, DO generates a symmetric key sk_1 and also randomly generates $\{0,1\}^{\ell} \rightarrow sk_2$. The sk_1 generated by this algorithm is used to encrypt the speech file stored in the cloud and the file hash, and sk_2 is used to encrypt each keyword with the index structure corresponding to the file.

2) $Enc(D, W, sk_1, sk_2) \rightarrow (C, I_B, B)$: Given a speech file $D = \{d_1, d_2, ..., d_n\}$ to be stored in the cloud servers, a keyword set $W = \{w_1, w_2, ..., w_n\}$ and keys sk_1, sk_2, the algorithm generates a encrypted speech library $C = \{c_1, c_2, ..., c_n\}$, a checklist B, an encrypted index I_B. For each speech file d_i, id_i is the file identifier of d_i, the extracted keyword w_i and id_i are used to construct the index table, DO encrypts speech by key sk_1, sk_2 encrypt the index structure to get the encrypted index I_B, and at the same time, the pseudo-random function H is used to calculate the encrypted speech hash value, and store encrypted speech and hash value into the encrypted speech library C and checklist B respectively. Finally, I_B and B are sent to the blockchain, and C and I_B are sent to CS for storage separately.

3) $TokenGen(w, sk_2) \rightarrow T_w$: Given the keyword w_i and the key sk_2 of the desired query speech file d_i as input, the search token T_w is output, and T_w is sent to the blockchain and CS, respectively.

4) $Search(C, I_B, B, T_w) \rightarrow (C_w, Judge)$: Given the encrypted speech library C, encrypted index I_B, checklist B, and search token T_w, the cloud servers, after receiving the search token, will query the corresponding file C_w and send it to the blockchain for verification based on the given keywords w and I_B. The blockchain obtains the hash value $H = \{hash_1, hash_2, ..., hash_n\}$ and computes $Judge = hash_1 \oplus hash_2 \oplus ... \oplus hash_n$ for data validation. The algorithm is shown as follows.

Algorithm 1 Search algorithm

Input: Encrypted speech library C, encrypted index I_B, checklist B, search token I_B

Output: Encrypted speech C_w, Verifying certificate $Judge$

DU:
$wList = [\,]$
for $w \in W$ **do**
  $b_w = F(H(w), sk_2)$; $t_w = H(\Sigma w \text{ —— } b_w)$; $wList = wList \cup [\text{tw}]$
end for
Server:
Fw = [ ]
for $w \in W$ **do**
  $b_w = F(H(w), sk_2)$; $t_w = H(\Sigma w \text{ —— } b_w)$; $wList = wList \cup [\text{tw}]$
end for
get encrypted specch $C_w = \{c_1, c_2, \ldots, c_n\}$
Blockchain:
get checkList $cL = \{hash_1, hash_2, \ldots, hash_n\}$
get $Judge = hash_1 \oplus hash_2 \oplus ... \oplus hash_n$

5) *Verify*(C_w, *Judge*)→(pf, r) : Given the retrieval result C_w and the Verifying Certificate *Judge*, the verification result and proof are output. To verify the integrity of the speech data, the blockchain will retrieve the result C_w by $H(C_w) \rightarrow H^{'}$, calculating the hash value and calculating $Judge^{'}$, then get the hash value $H^{'}$ corresponding to the file ID in the result from checklist B, compare the two of *Judge* and $Judge^{'}$, if they are equal, pf is true, the retrieval result r is returned. Otherwise it is false, and the retrieval result is returned ϕ. The verification process is performed on the blockchain, and the hash value stored on the blockchain cannot be modified, so the fairness of the verification is guaranteed. The algorithm is shown as follows.

Algorithm 2 Verify algorithm

Input: Retrieval results C_w, Verifying certificate *Judge*

Output: Proof pr, Verifying result r

Blockchain:
$H_w = \phi$
for $c_i \in C_w$ **do**
  $H_w = H_w \oplus \mathbf{H}(c_i)$
end for
if $H_w = Judge$ **then**
  pr = True, $r = C_w$
else
  pr = False, $r = \phi$
end if
send pr, r to **DU**

6) *Dec*(C_w, sk_1)→D_w : Given the symmetric key sk_1 and the verified correct retrieval file C_w, output the plaintext speech file D_w of this encrypted speech.

4 Performance Analysis

To evaluate the retrieval performance of the proposed method and to verify the universality, the open Chinese speech database THCHS-30 [15] of Tsinghua University and the speech in the TIMIT [25] English dataset were used to evaluate the proposed method. THCHS-30 dataset contains news clips containing 1,000 different contents, comprising 13,388 speech clips. TIMIT dataset consists of 630 speakers from 8 regions in the United States speaking ten sentences, with a total of 6,300 speech clips. Different speech content clips from the two datasets were subjected to content retention operations to obtain 15,000 speech clips each for model training.

The experimental hardware platform is Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz, 32GB RAM. The software environment is Windows 10, PyCharm 2021.1.3, MATLAB R2021b, Solidity language to write smart contracts, and using TestRPC to simulate an Ethereum environment.

4.1 Speech Encryption Performance Analysis

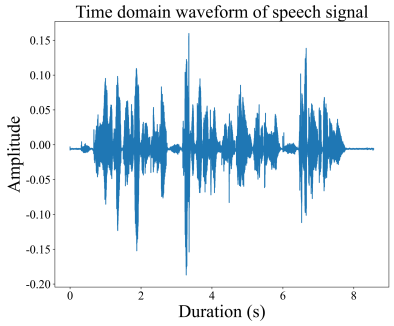
Speech encryption must first ensure the privacy of the speech and resist extraneous attacks. The encryption algorithm should also be guaranteed to be unbreakable for several hours. The security of the proposed encryption algorithm is analyzed regarding key space, histogram, and audio entropy.

Key space analysis. The key space is an important component of the encryption algorithm for encrypting speech, and the larger the key space of the algorithm, the higher the key flexibility and achievability. The key length of the AES-128 encryption algorithm is 128 bits, and the key space is 2^{128}, which is enough to resist any brute force attack to crack the algorithm. Therefore, the AES-128 algorithm is secure enough.

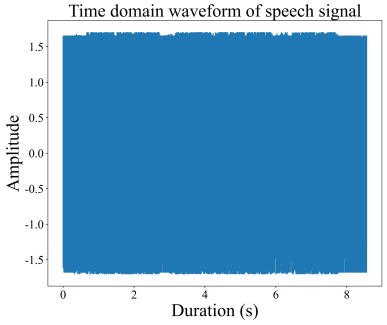
Histogram analysis. Figure 6(a) and Figure 6(b) show the original and encrypted speech signals of a randomly selected speech in the THCHS-30 dataset, respectively.

As can be seen in Figure 6, the waveform in Figure 6(b) produces rapid changes during the encryption process, in contrast to the waveform graph shown in Figure 6(a); meanwhile, the speech fragments after being encrypted cannot be partially recovered, which fully demonstrates that the proposed encryption algorithm has better security.

Information entropy analysis. Information entropy is the expectation of the amount of information that could be generated to gain the degree of uncertainty. To resist statistical attacks, a higher expectation is required. Table 1 shows the results of

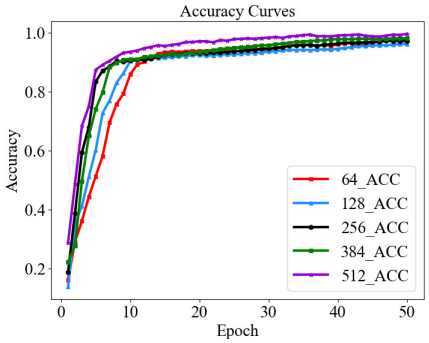


(a) Original speech

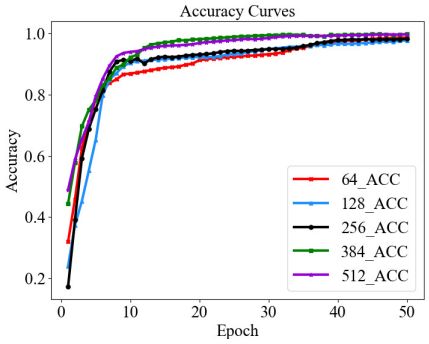


(b) Encrypted speech

Figure 6: The flow diagram of decryption algorithm



(a) THCHS-30 dataset



(b) TIMIT dataset

Figure 7: Test accuracy curves of C-BiGRU model

the information entropy comparison for some of the encrypted speech used in this paper.

As seen from Table 1, the speech encrypted by the proposed encryption methods has higher entropy values, better speech encryption performance and higher security than the encryption methods used in Ref. [2, 5, 13] and is suitable for speech encryption.

4.2 Retrieval Performance Analysis

To achieve the most superior model feature representation capability, this section employs hash sequences of different lengths to evaluate the C-BiGRU model's test accuracy experimentally. Figure 7(a) and Figure 7(b) show the test accuracy curves of the C-BiGRU model under the THCHS-30 and the TIMIT dataset, respectively.

As seen in Figure 7, the proposed model tests the highest accuracy when the hash code length is 512. Although the longer hash codes have more advantages in terms of feature expression, the retrieval efficiency only improves some of the time. Considering the subsequent searchable encryption and the calculation overhead during data integrity verification, as well as the basic overlap in test accuracy between 384-bit and 512-bit in the last ten iterations. The proposed method choose to construct a deep hash code of length 384.

Figure 8 compares test accuracy and loss curves of the C-BiGRU model under THCHS-30 and TIMIT datasets when the hash code length is 384 bits. As the number of iterations increases, there is no difference in the retrieval accuracy between the training and test sets in the two different datasets, which fully indicates that no overfitting and underfitting occurs in the training of the proposed C-BiGRU model, and the test accuracy and loss value reach the expected level.

Figure 9(a) and Figure 9(b) show the test accuracy curves of the C-BiGRU model before and after feature fusion under THCHS-30 and TIMIT datasets. Under different datasets, the test accuracy of the MFCC-Fbank is significantly higher than that of the features before fusion after the same number of training of the model, reaching the expected value, fully indicating the better retrieval performance of the MFCC-Fbank.

Table 2 shows the test accuracy and mAP values comparison of different features of different models. When the extracted features are MFCC and Fbank, respectively, the highest test accuracy reaches 99.06% and 98.21% when constructing a deep hash code of 384 bits in length; when the selected features are MFCC-Fbank, the highest test accuracy reaches 99.75%, which has better retrieval accuracy compared with other features. At the same time, the mAP values of the fused MFCC-Fbank features are also improved compared to the low-level features before fusion. The mAP values are calculated as shown in Equation (9). Test accuracy and mAP values of the C-BiGRU model also have a significant advantage over the other models.

Table 1: Information entropy and PSNR of the encryption algorithm

Methods	Encryption method	THCHS-30		TIMIT	
		Original	Encrypted	Original	Encrypted
Proposed	AES-128	4.9043	7.9786	2.1445	7.6568
Ref. [5]	4D Hyperchaotic	5.3306	7.1511	3.9566	7.2589
Ref. [13]	AES	3.8852	6.9655	3.0133	6.8844
Ref. [2]	RSA	5.967	6.5581	4.2106	6.3697

Table 2: Comparison of the test accuracy and mAP values for different features of different models

Network	Methods	Test Accuracy (%)	mAP(%)
CNN	Fbank	97.18	95.15
	MFCC	99.03	98.21
	MFCC-Fbank	99.14	98.85
BiGRU	Fbank	94.68	91.86
	MFCC	98.37	97.43
	MFCC-Fbank	98.55	98.33
C-BiGRU	Fbank	98.21	96.83
	MFCC	99.06	98.62
	MFCC-Fbank	**99.75**	**99.07**

$$mAP = \frac{1}{|Q_R|} \sum_{q \in Q_R} AP(q). \tag{9}$$

where —Q_R— denotes the number of queries, $AP(q)$ denotes the accuracy of the q-th query, N denotes the number of speech files, and $pr(N)$ denotes whether the N-th speech fragment is relevant to the queried content, and is 1 if it is relevant and 0 if it is not.

The recall rate (R), precision rate (P) and F1 score are also commonly used to evaluate the performance and goodness of the model, and the calculation formulae are shown in Eqs. (10)-(12). Table 3 shows the comparison results of the proposed model's P, R and F1 score after five different content preserving operations.

$$R = \frac{TP}{TP + FN} \times 100\%. \tag{10}$$

$$P = \frac{TP}{TP + FP} \times 100\%. \tag{11}$$

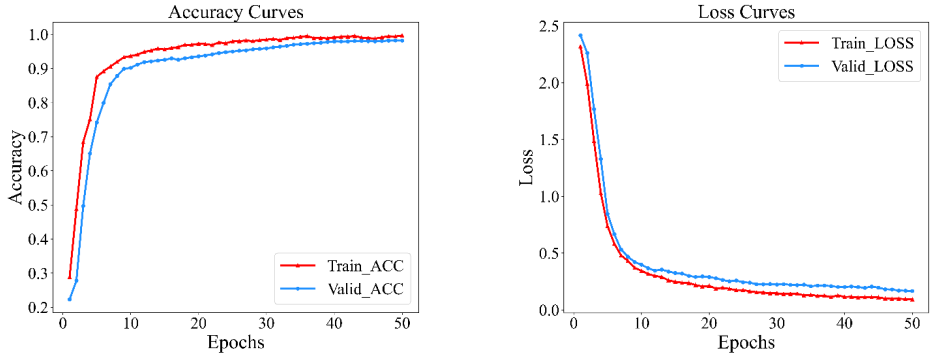
$$F1 = \frac{2P \times R}{P + R}. \tag{12}$$

where TP and FP are the retrieved query-related and unrelated speech, respectively, FN is the number of not retrieved query-related speech, the sum of TP and FN is the total number of query-related speech, and the sum of TP and FP is the total number of returned speech.

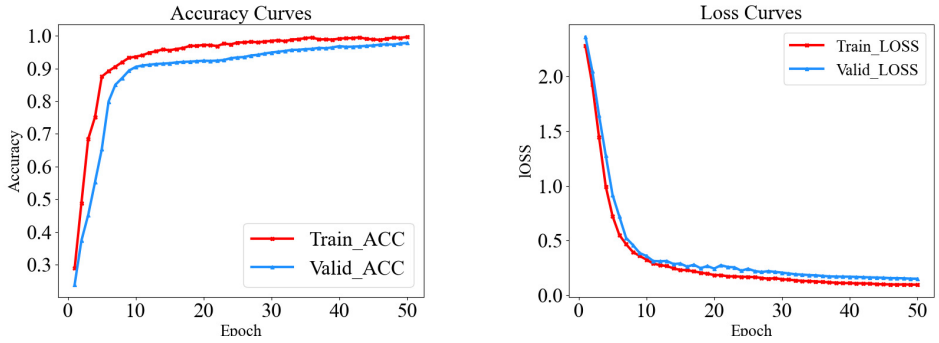
As shown in Table 3, the proposed model has obvious advantages in R and P under different data sets and five various content retention operations. The retrieval accuracy rate is close to 100% under multiple content retention operations, which is more robust. Although it is slightly deficient in individual content preserving operations, it is still an excellent retrieval algorithm suitable for most retrieval environments.

As can be seen from Figure 10, the retrieval performance of the proposed method is slightly less than that of Zhang's method (2021) [22]. Still, from the area enclosed by the comparative Huang's method (2021) [5], Li's method (2021) [6] and Petcharat 's method (2019) [11] method and the area surrounded by the coordinate axes, the retrieval performance of the proposed method is significantly better than the comparative Huang's method (2021), Li's method (2021) and Petcharat 's method (2019) method.

To analyze the retrieval efficiency of the proposed method for encrypted speech, 1,000, 3,000, 5,000, 7,000 and 10,000 speech fragments were randomly selected from the THCHS-30 and TIMIT datasets, respectively. Table 4 shows the comparison of the retrieval time under different datasets.

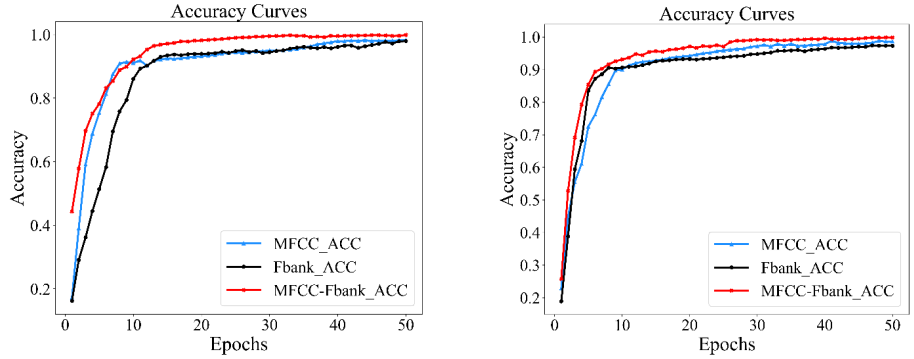


(a) THCHS-30 accuracy curve (b) THCHS-30 loss curve



(c) TIMIT accuracy curve (d) TIMIT loss curve

Figure 8: Comparison of accuracy and loss curves of C-BiGRU model in different datasets



(a) THCHS-30 dataset (b) TIMIT dataset

Figure 9: Comparison of accuracy curves of C-BiGRU model before and after feature fusion

4.3 Performance Comparison with Existing Speech Retrieval

To demonstrate the superiority of the retrieval performance of the proposed method, an objective analysis is performed from several evaluation metrics and the retrieval performance is compared with that of existing Ref. [6, 7, 11, 21, 22]. Table 5 shows the performance comparison results of the proposed method with existing speech retrieval methods.

Table 5 shows that the R, P, and F1 score of the proposed method is better than speech retrieval methods in Ref. [6, 7, 11], respectively, indicating that the proposed method has good applicability to different speech fragments. The speech retrieval performance is improved by constructing compact hash binary codes as keywords for encrypted speech retrieval. The proposed method outperforms Ref. [6, 7, 11] in terms of P, R, and F1 score, mainly because the features used in the speech retrieval in Ref. [6, 7, 11] are low-level features of speech. In contrast, the proposed method fuses the extracted features MFCC and Fbank for dimensionality reduction, extracts deep se-

Table 3: Comparison of the test accuracy and mAP values for different features of different models

Content preserving operations	THCHS-30			TIMIT		
	P/%	R/%	F1	P/%	R/%	F1
Re-sampling	99.7	99.5	0.996	99.9	99.6	0.9975
MP3 compression	100	99.9	0.9995	98.79	99.61	0.992
Amplitude increase	100	100	1	100	100	1
Amplitude decrease	100	100	1	100	100	1
Adding noise	98.3	97.1	0.977	99.1	98.5	0.988

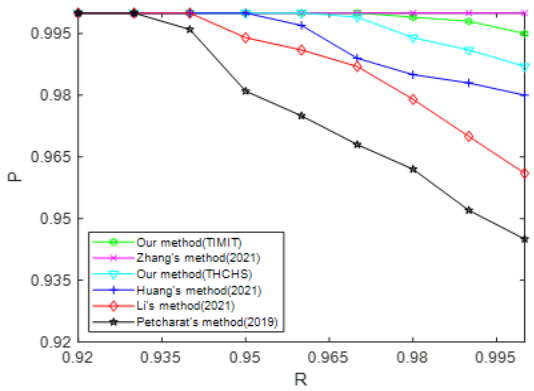


Figure 10: P-R curve comparison results

Table 4: Comparison of retrieval time for different number of speech segments

Number of speech	Retrieval time (ms)	
	THCHS-30	TIMIT
1,000	11.41	10.33
3,000	12.24	11.59
5,000	13.89	13.29
7,000	16.97	16.07
10,000	18.19	17.85

mantic features, constructs a hash binary code with richer semantic information, and improves the speech retrieval accuracy. The proposed method is lower than that proposed by Ref. [21, 22] in terms of P, R, and F1 score because the extracted speech features MFCC and LPCC are used to construct a combined feature matrix, the feature dimensionality reduction method with information entropy and energy is given to reduce the dimensionality of the feature matrix and retain most of the features. At the same time, the deep learning used in the proposed method to extract semantic features will lose the original feature information to some extent during the training process; Ref. [21] is relatively small in the number of speech, and the number of matches is reduced during speech retrieval. When extracting the features of MFCC, the retrieval performance of the proposed method could be better in comparison. The model is complex compared to the LSTM model used in Ref. [21] due to the large number of parameters used, and the speech features appeal to poor realism, thus, the retrieval performance is slightly deficient.

Regarding retrieval efficiency, the proposed method outperforms Ref. [6, 11, 22]. In constructing hash binary codes, the proposed method uses double k-means to divide the dataset into two random parts with the threshold values of arithmetic mean and median, which may still be associated with a close centroid even if they are not associated with the corresponding centroid, and finally the two parts of hash codes are combined as hash binary codes, which effectively reduces the retrieval error. The Ref. [22] did not change in the index structure and did not construct hash binary codes, and only retrieved by the normalized Hamming distance algorithm. Ref. [6, 11] retrieves data through the hierarchical retrieval structure of hash tables. Still, the number of hash tables is too large, the multi-probe method will have the problem of coarse quantization or quantization error, and the retrieval workload is large. The retrieval efficiency of the proposed method is slightly lower than that of Ref. [7,21], the same model used for deep hash construction, and the feature dimensionality is slightly slower in retrieval speed due to the higher dimensionality of the individual features extracted by the fusion compared to Ref. [21]. The dataset of Ref. [7] is a homemade foreign language dataset with a short length of speech segments, which has a shorter total duration and higher retrieval efficiency compared to the Chinese speech library used in the proposed method.

Regarding the verifiability of retrieval results, the proposed method achieves result verification compared to Ref. [6, 7, 11, 21, 22]. Combining the tamper-evident nature of the blockchain and the HMAC-SHA256 algorithm, the integrity of the retrieved results is verified while preventing malicious cloud servers from modifying the results and data users from evading verification, which is more fair and trustworthy and has high practicality.

Table 5: Comparison of the test accuracy and mAP values for different features of different models

Metrics	Ref. [22]	Ref. [6]	Ref. [11]	Ref. [21]	Ref. [7]	Proposed
Model	-	-	DNN	LSTM	LSTM	C-BiGRU
Speech Feature	MFCC-LPCC	Audio Fingerprint	Audio Fingerprint	MFCC	MFCC	MFCC-Fbank
R(%)	100	98.1	91.5	100	93	99.7
P(%)	100	97.2	98.92	100	94	99.5
F1 score	1	0.9765	0.9506	1	0.935	0.996
Number of speech	1,000	5,000	678	1,000	3,600	10000
Retrieval time (s)	0.5328	1.027	0.15	0.269	0.139	0.0182
Result Verification	**No**	**No**	**No**	**No**	**No**	**Yes**

4.4 HMAC-SHA256 Algorithm Security Analysis

DU verifies the integrity of retrieval results by the HMAC-SHA256 algorithm on the smart contract. The speech is combined with an encryption key to generate a fixed-length message digest as the output using the SHA-256 function.

The security of HMAC is determined partly by the chosen hash function and partly by the key, and the SHA256 algorithm, as the core algorithm, has several important properties:

1) Anti-sub-originality: given a speech file d_i, another input speech d cannot be constructed such that the hash value h_i obtained by d_i using the hash function corresponds to the hash value h obtained by speech d using the hash function.

2) Collision resistance: given any speech file in the dataset, the same hash value will not be generated.

3) Avalanche effect: The algorithm is highly sensitive; for each bit position of the original speech file changes, a large part of the output hash value will also change.

The key also plays a crucial role in the security of the HMAC algorithm. The key length of the HMAC algorithm is arbitrary. Even if the length of the key keeps increasing, the security of the HMAC algorithm does not keep improving, and a shorter key reduces security. Therefore, the output hash value length is the prime choice for the length of the key of the proposed method. The key is generated by a random number generator and updated frequently to ensure security. If malicious users cannot get the key, even if the SHA-256 algorithm is cracked, it still cannot cause an effective attack on the content. Therefore, HMAC is used for verification and is very secure.

4.5 Security Analysis of Searchable Encryption Algorithm

The searchable encryption scheme requires that the CS cannot identify any information related to the document and the keywords queried by the user after the search is completed. In this paper, the adversary in the searchable encryption scheme is adaptive, i.e., the adversary can select new keywords based on previous keywords and search results, constructing games $Ideal_{A,S}(\gamma)$ and $Real_A(\gamma)$, proving that the scheme is indistinguishably secure under selective keyword attacks (INDistinguishability under Chosen Keywords Attack, IND-CKA2).

Definition 1. *A verifiable searchable encryption scheme on the blockchain is IND-CKA2 secure under the attack of adaptive adversary* A. $\mathbf{L} = \{Setup, Search\}$ *is the leakage function,* A *is the probabilistic polynomial-time adversary,* S *is the polynomial-time simulator of* A. *Adversary* A *cannot computationally distinguish the output results of the game* $Ideal_{A,S}(\gamma)$ *and* $Real_A(\gamma)$*, and the security game of simulator* S *with adversary* A *is as follows:*

$Real_A(\gamma)$**:** *Challenger* CH *runs* $Setup(1^\lambda)$ *generates a key set* $K = \{sk_1, sk_2\}$*, and sends it to adversary* A. A *randomly chooses document* $D = \{d_1, d_2, ..., d_n\}$ *and sends it to* CH. *Challenger* CH *runs* Enc*(*D, W, sk_1, sk_2*)*$\rightarrow$*(*C, I_B, B*), and sends (*C, I_B*) to* A. A *chooses different keywords* w_i *for polynomial times of adaptive queries, and for each query* $Q = \{q_1, q_2, ..., q_n\}$*, receives the search token generated from the challenger run* TokenGen*(*w, sk_2*)*$\rightarrow T_w$*, and finally,* A *receives the token and returns the output of the game.*

$Ideal_{A,S}(\gamma)$**:** A *randomly chooses document* D*, known as* $\mathbf{L}$*(*D*), and simulator* S *generates and sends (*C, I_B*) to* A. A *chooses different keywords* w_i *and generates a polynomial number of adaptive queries* $Q = \{q_1, q_2, ..., q_n\}$*. According to each query* q_i*, simulator* S *generates search tokens* T_w *and* $S(\mathbf{L}(D)) \rightarrow D_i$*, and finally,* A *receives the tokens to return the output of this game. If, for* A*, there exists a valid simulator* S *satisfying Equation (13), it is shown that the proposed searchable encryption scheme is secure under IND-CKA2.*

$$|\Pr[Real_A(\gamma) = 1] - \Pr[Ideal_{A,S}(\gamma) = 1]| \leq negl(\gamma). \quad (13)$$

where negl *is the ignorable function and* γ *is the safety parameter.*

4.6 Smart Contracts Consumption Analysis

In Ethereum, when smart contracts need to be deployed or invoked, a certain consumption is generated, which is in wei as a unit. The consumption varies from command to command, increasing with the computational resources consumed increase. When conducting this experiment, 1 ether = 1,405 USD and 1 gasprice = 2×10^{-9} ether. According to different function methods, Table 6 shows the cost of smart contracts.

It can be from Table 6 that the cost of deploying search and verify contracts is 5.64562×10^5 wei = 1.5864 USD and 5.86431×10^5 wei = 1.6477 USD, respectively. **TestSearch** and **TestVerify** functions initialize the search function and **Verify** function parameters. The **AddTrapdoor** function only adds search tokens and does not consume much gas. DU calls the **Search** function is designed to compare the search token with the stored keyword index and consumes the most gas; **calHash** function calculates the retrieval result hash value by the HMAC-SHA256 algorithm; the primary gas consumption is to call HMAC-SHA256 algorithm; **Verify** function is invoked by DU to verify whether the incoming file hash value is consistent with the file hash value stored in the blockchain. If the CS returns the retrieval result is modified, DU will not pay the corresponding gas to CS. In the meantime, the blockchain is introduced to verify the retrieval results, and DU cannot save computational overhead and skip the verification or falsify the results. Therefore, the fairness and reliability of the verification are ensured.

The above experimental results show that the gas consumption of smart contracts is acceptable for all entities, and it is feasible to use smart contracts to search and verify the integrity of the returned data.

5 Conclusions

This paper proposes a verifiable encrypted speech retrieval method based on blockchain and C-BiGRU, which not only solves the problems of privacy leakage and doubtful correctness of retrieval results in the retrieval process of existing encrypted speech but also protects the privacy security of speech, improves the retrieval accuracy and efficiency, and realizes the verification of retrieval results in the blockchain environment. To ensure efficient speech retrieval, the method designs a C-BiGRU model, and the fused MFCC-Fbank features are fed into the C-BiGRU model for training to obtain deeper features with higher retrieval accuracy, which further improve the retrieval performance. A searchable encryption algorithm is used to ensure the privacy and confidentiality of the data returned by the user and improve the security of the speech data retrieval.

The future research in this paper will concentrate on multi-keyword verifiable encrypted speech retrieval, with the aim of enhancing retrieval efficiency while ensuring data integrity.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61862041). Natural Science Foundation of Gansu Province of China(No.21JR7RA120). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] Y. H. Cai, Y. Y. Li, and C. Y. Qiu, "Medical image retrieval based on convolutional neural network and supervised hashing," *IEEE access*, vol. 7, pp. 51877–51885, 2019.

[2] X. R. Ge, J. Yu, and H. L. Zhang, "Towards achieving keyword search over dynamic encrypted cloud data with symmetric-key based verification," *IEEE Transactions on Dependable and Secure computing*, vol. 18, no. 1, pp. 490–504, 2019.

[3] H. Q. Guo and X. J. Yu, "A survey on blockchain technology and its security," *Blockchain: research and applications*, vol. 3, no. 2, p. 100067, 2022.

[4] K. Hossain, M. Rahman, and S. Roy, "Iot data compression and optimization techniques in cloud storage: current prospects and future directions," *International Journal of Cloud Applications and Computing*, vol. 9, no. 2, pp. 43–59, 2019.

[5] Y. B. Huang, S. H. Wang, and Y. Wang, "A hyperchaotic encrypted speech perceptual hashing retrieval algorithm based on 2d-gabor transform," *International Journal of Network Security*, vol. 23, no. 5, pp. 924–935, 2021.

[6] T. H. Li, M. S. Jia, and X. Cao, "A hierarchical retrieval method based on hash table for audio fingerprinting," in *Intelligent Computing Theories and Application: 17th International Conference (ICIC)*, pp. 160–174, Shenzhen, China, Augest 2021.

[7] W. Li, Y. Z. Xiao, C. Tang, and X. J. Huang, "Multi-user searchable encryption voice in home iot system," *Internet of Things*, vol. 11, p. 100180, 2020.

[8] X. H. Li, Q. Y. Tong, and J. W. Zhao, "VRFMS: Verifiable ranked fuzzy multi-keyword search over encrypted data," *IEEE Transactions on Services Computing*, vol. 16, no. 1, pp. 698–710, 2023.

[9] Q. Liu, Y. Tian, and J. Wu, "Enabling verifiable and dynamic ranked search over outsourced data," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 69–82, 2019.

[10] A. Michalas, "The lord of the shares: Combining attribute-based encryption and searchable encryption for flexible data sharing," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC)*, p. 146–155, New York United States, April 2019.

[11] P. Petcharat, S. Kamonsantiroj, and L. Pipanmaekaporn, "Unsupervised learning hash for content-based audio retrieval using deep neural networks," in *2019*

Table 6: Cost of smart contracts

Operation	Gas ($\times 10^5$)	ETH ($\times 10^{-3}$)	USD
Deploy Search Contract	5.64562	1.129124	1.5864
Deploy Verify Contract	5.86341	1.172682	1.6477
Execute TestSearch function	0.09027	0.018054	0.0253
Execute AddTrapdoor function	0.48962	0.097924	0.1376
Execute Search function	1.78111	0.356222	0.5005
Execute TestVerify function	0.17833	0.035667	0.0501
Execute calHash function	0.07112	0.014224	0.0199
Execute Verify function	**1.49168**	**0.298336**	**0.4192**

11th International Conference on Knowledge and Smart Technology (KST), pp. 99–104, Phuket, Thailand, January 2019.

[12] P. Prajapati and P. Shah, "A review on secure data deduplication: Cloud storage security issue," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 3996–4007, 2022.

[13] Q. Y. Tong, Y. B. Miao, and J. Weng, "Verifiable fuzzy multi-keyword search over encrypted data with adaptive security," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 5386–5399, 2022.

[14] L. Turchet, G. Fazekas, and M. Lagrange, "The internet of audio things: State of the art, vision, and challenges," *IEEE internet of things journal*, vol. 7, no. 10, pp. 10233–10249, 2020.

[15] D. Wang and X. W. Zhang, "THCHS-30: A free chinese speech corpus," *arXiv e-prints*, p. arXiv:1502.03167, 2015.

[16] W. S. Xu, J. B. Zhang, and Y. L. Yuan, "Towards efficient verifiable multi-keyword search over encrypted data based on blockchain," *PeerJ Computer Science*, vol. 8, p. e930, 2022.

[17] Y. Yang, H. R. Lin, and X. M. Liu, "Blockchain-based verifiable multi-keyword ranked search on encrypted cloud with fair payment," *IEEE Access*, vol. 7, pp. 140818–140832, 2019.

[18] Z. L. Yang, J. Yue, and Z. B. Li, "Vegetable image retrieval with fine-tuning vgg model and image hash," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 280–285, 2018.

[19] J. A. Yu, Y. K. Hou, and S. Li, "A high-speed data retrieval model on blockchain," in *2022 11th International Conference of Information and Communication Technology (ICTech)*, pp. 101–105, Wuhan, China, February 2022.

[20] F. Zalkow and M. Müller, "Ctc-based learning of chroma features for score–audio music retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2957–2971, 2021.

[21] Q. Y. Zhang, M. R. Fu, and Y. B. Huang, "Encrypted speech retrieval scheme based on multiuser searchable encryption in cloud storage," *Security and Communication Networks*, vol. 2022, p. 9045259, 2022.

[22] Q. Y. Zhang, F. J. Xu, and J. Bai, "Audio fingerprint retrieval method based on feature dimension reduction and feature combination," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 2, pp. 522–539, 2021.

[23] X. K. Zheng, Y. Q. Zhao, and H. L. Li, "Blockchain-based verifiable privacy-preserving data classification protocol for medical data," *Computer Standards and Interfaces*, vol. 82, p. 103605, 2022.

[24] Z. B. Zheng, S. A. Xie, and H. N. Dai, "Vegetable image retrieval with fine-tuning vgg model and image hash," *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.

[25] V. Zue, S. Seneff, and J. Glass, "Speech database development at mit: Timit and beyond," *Speech communication*, vol. 9, no. 4, pp. 522–539, 1990.

Biography

Fang-Peng Li is currently a master student of the School of Computer and Communication, Lanzhou University of Technology, China. He received the BS degrees in software engineering from Lanzhou University of Technology, Gansu, China, in 2021. His research interests include network and information security, encrypted speech retrieval, searchable encryption.

Qiu-yu Zhang Researcher/Ph.D. supervisor, graduated from Gansu University of Technology in 1986, and then worked at school of computer and communication in Lanzhou University of Technology. He is vice dean of Gansu manufacturing information engineering research center, a CCF senior member, a member of IEEE and ACM. His research interests include network and information security, information hiding and steganalysis, multimedia communication technology.

Ying-jie Hu Received a master's degree from Lanzhou University in 2011 and currently serves as a teacher at the School of Computer and Communication at Lanzhou University of Technology. Her research interests include

multimedia information processing, information security, and speech recognition?network information security.

Yi-Bo Huang received the PhD degree from Lanzhou University of Technology in 2015, and now working as a Associate Professor in the college of physics and electronic engineering in Northwest Normal University. His research interests include multimedia information processing, information security and speech recognition.

An Improved Received Signal Strength Indication Location Algorithm Based on Gaussian Filter and Quasi-Newton Method

Xin Qiao[1], Jing Wang[1], Haiyang Shen[1], and Fei Chang[2]
(Corresponding author: Xin Qiao and Jing Wang)
School of Electronic Engineering, Chaohu University
Chaohu, Anhui 238000, China[1]
Huishang Futures Co. Ltd, Hefei, Anhui, 230061, China[2]
Email: fei612@sina.com
(Received May 16, 2023; Revised and Accepted Sept. 22, 2023; First Online Apr. 25, 2024)

Abstract

Aiming at the low accuracy and poor robustness of localization algorithm in wireless sensor networks, an improved Received Signal Strength Indication (RSSI) location algorithm based on Gaussian filter and quasi-Newton method without adding node hardware is proposed. First, in the ranging stage, the received RSSI values are processed by Gauss filtering, and the dynamic parameters of the model are obtained by periodically measuring the path loss factor to correct the RSSI ranging. Then, in the location stage, the errors between the unknown and anchor nodes are used to introduce a correction coefficient of the distance measurement error $\Delta\mu$. Finally, taking the estimated value as the initial value, the coordinates of the unknown nodes are iteratively optimized by the quasi-Newton optimization algorithm. The simulation results show that the location accuracy of the proposed algorithm is improved by 20% compared to the traditional RSSI algorithm and about 5%-10% to literature9 and literature 18. The improved algorithm has higher positioning accuracy without increasing the node hardware cost. It can solve the problems of large positioning errors and a small application range of wireless sensor networks.

Keywords: Correction Coefficient; Path Loss Factor; Quasi-newton Algorithm; Wireless Sensor Network

1 Introduction

Node location technology is one of the key supporting technologies in wireless sensor network (WSN), and a basis of network topology management, coverage control and routing algorithm design of nodes. Its location accuracy will directly affect the overall performance of networks. Thereby it is particularly important to improve the location accuracy of sensor nodes [22].

Many researchers have proposed various location algorithms without GPS modules. Practically, they can be divided into two types: ranging algorithm and non-ranging algorithm [19]. Since non-ranging location algorithms are coarse-grained, the current location systems cannot meet the requirements of high-precision location, and most are based on ranging location algorithms. Common ranging algorithms are Time difference of arrival (TDOA) [12], Time of arrival (TOA) [13], Angle of arrival(AOA) [15] and Received signal strength indication(RSSI) [21]. The first three methods need high-cost hardware. The node has wireless transmission function and the signal strength is easy to measure. RSSI algorithm is widely used due to its low price and high location accuracy. However, in actual environment, it is easy to be affected by reflection, multipath propagation, antenna gain and obstacles, with large measurement errors. As such, related measures are necessary for the improvement of location accuracy [4].

In recent years, there have been many related studies on RSSI location. Since the RSSI algorithm depends on the environment and leads to poor location accuracy, the algorithm can reduce the ranging error by increasing the correction factor and introducing weights, so as to correct the ranging value between the nodes to-be-located and the beacon nodes [8]. In this way, a probabilistic algorithm was proposed based on Bayes' theorem [7]. The algorithm used the Blue-tooth RSSI measurement data as the prior information to dynamically select the location beacons by setting distance thresholds. According to the Bayesian algorithm, the location distribution probability of the to-be-located points for multiple location beacons was estimated. The coordinates of the maximum probability points were used as the estimated location of unknown nodes. Based on the location technology of RSSI, a weighted centroid location method with power correction was introduced to improve the accuracy [2]. In the Hop Distance-Corrected Localization algorithm [1], the

Hop distance was corrected through unbiased estimation, and the collinearity was used to select suitable anchor nodes to estimate the position of unknown nodes. At the same time, an improved unconstrained optimization 3D DV-Hop (Distance Vector-Hop) location algorithm was used [26]. The minimum hop value was calculated by a dual communication radius strategy, and its weighted hop distance values were used as the hop distance values of the unknown nodes. In addition, a new distributed localization algorithm was established to solve highly nonlinear non-convex optimization problems in large-scale sensor networks [25]. This algorithm decomposed a global undirected graph composed of large-scale wireless sensor networks into a series of partially overlapping subgraphs, allowing the optimization problem to be solved iteratively in each subgraph independently. Furthermore, a clustering routing algorithm based on link quality was proposed to solve the problem of unstable or failed data transmission [6]. A link quality estimation model based on gradient boosting decision tree algorithm was constructed to determine the packet reception rate based on RSSI, LQI (Link Quality Indicator) and SNR (Signal to Noise Ratio). Clusters were then established based on the estimated PRR (Pulse Repetition Rate) to implement efficient data transmission. The non-line-of-sight recognition method based on weighted k-nearest neighbor classification [24] has high recognition accuracy and wide applicability. Aiming at the fact that the underground WLAN location fingerprinting personnel positioning system does not fully consider the singularity problem, a class- relationship K-Means algorithm was proposed [23]. This algorithm takes the ratio of intra-class dispersion and inter-class dispersion as the objective function, and realizes the optimal clustering without singularity through the clustering aggregation and separation process of the minimum ratio, thus completing the reasonable division of location area. Consequently, the positioning accuracy using this algorithm was improved effectivity.

The positioning algorithms have certain application value in improving the positioning accuracy, but most of the algorithms require high network topology, large communication overhead and large computation. This paper aims to effectively improve the positioning accuracy of nodes without increasing the computational and hardware costs.

In view of the low accuracy and high computation in current location algorithms, an improved RSSI location algorithm based on Gaussian filter and Quasi-Newton method is improved. At each location stage, different schemes are adopted to reduce errors. In the ranging stage, the RSSI values received by unknown nodes are filtered by Gauss filter, and the mean value is taken as the measurement value. Through the periodical and dynamical measurement of the RSSI values between the anchor nodes near the nodes to-be-located, the path loss factor at the anchor node is calculated in real time, and the distance is obtained according to the wireless signal transmission model. In the location stage, the distance errors are fully utilized by unknown nodes and anchor nodes, and the correction coefficient $\Delta\mu$ of the distance measurement error is introduced. The position coordinates of unknown nodes are iteratively estimated by quasi-Newton method.

The background, research status, existing problems and solutions of the improved method are presented in Section 1; the principle of RSSI localization algorithm is described in Section 2; the localization algorithm of wireless sensor networks based on Gaussian filter and quasi-Newton method is presented in Section 3, including the pretreatment of RSSI Gaussian filter algorithm, dynamic acquisition of path loss factor, establishment of signal transmission model in line with the actual environment, optimization of distance measurement error, and iterative refinement of unknown node coordinates using quasi-Newton method. In Section 4, the proposed algorithm is simulated and compared with the traditional localization algorithm, literature 9 and literature 18 to verify the positioning accuracy of this algorithm. Finally, Section 5 gives the conclusion and further research direction.

2 Principles of RSSI Location Algorithms

The common used signal propagation model in wireless sensor networks is the logarithmic-normal distribution model [5, 11, 16], described as:

$$P(d) = P(d_0) - 10k\lg(\frac{d}{d_0}) + \xi_n \tag{1}$$

where $P(d)$ is the RSSI value of the signal strength after the electromagnetic wave signal passes the distance d;$P(d_0)$ is the signal strength when the electromagnetic wave signal passes the distance d_0,$d_0 = 1m$;k is the Path loss factor, reflecting the influence of the surrounding environment on the ranging, usually as 1-5; ξ_n is a Gaussian random variable with a mean of zero and a variance of σ, and the standard deviation is generally 2-10.

The distance between two nodes can be derived from Formula (1) [27]:

$$d = 10^{\frac{P(d_0)+\xi_n-P(d)}{10k}} \tag{2}$$

When the unknown nodes receive information from three or more anchor nodes, the trilateral location method or least square method can be used to solve the position coordinates [3, 20].

As the path loss factor varies, the relationship between the RSSI and the path loss factor k is shown in Figure 1, and that between the RSSI and the signal transmission distance under theoretical and practical conditions is shown in Figure 2.

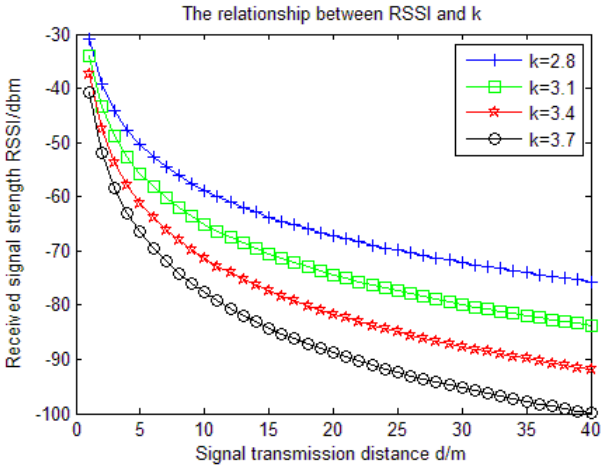


Figure 1: The relationship between RSSI and k

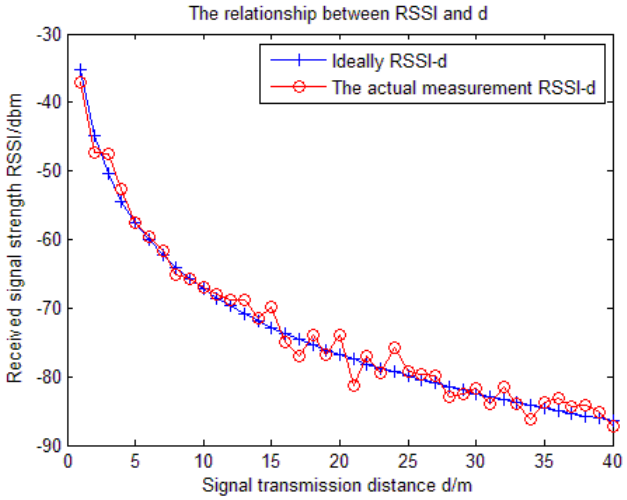


Figure 2: The relationship between RSSI and and d

From the logarithmic-normal distribution model, it can be seen that the received signal strength is related to the distance d between the anchor node and the unknown node, and also related to the path loss factor k. The path loss factor is a real-time dynamic parameter that varies in different environments. The traditional RSSI location algorithm uses the fixed path loss factor to calculate the distance, which will result in large measurement errors.

3 An Improved RSSI Location Algorithm Based on Gaussian Filter and Quasi-Newton Method (G-K-RSSI)

3.1 RSSI Gaussian Filter Preprocessing

During the target location process, Gauss filtering is performed on the RSSI values received by the unknown nodes, and then the arithmetic mean of these RMS values is calculated. Then processed RSSI values are used for location. Gauss filtering model is suitable for event sets that obey or approximately obey the lognormal distribution law [3, 18]. Formula (1) is the lognormal distribution model. Assuming that an unknown node receives n sampling value of RSSI signal $\{x_1, x_2, x_3, \cdots, x_n\}$, the sampling n value obeys the normal distribution (μ, σ^2), and the RSSI value approximately obeys the Gauss distribution . The probability density function is as follows:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3}$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4}$$

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x-\mu)^2 \tag{5}$$

where μ is the mean, dBm and σ is the standard deviation, dBm. The interval probability of the regular function $F(x)$ is $P(|x_i - \mu_i| < \sigma) = 2\Phi(1) - 1 = 0.683$. $\Phi(x)$ is the distribution function of the standard normal distribution. The RSSI value presents in this area are selected as valid values and average to obtain the optimized RSSI value:

$$\overline{RSSI} = \frac{1}{n}\sum_{i=1}^{n} x_i (x_i \in (\mu - \sigma, \mu + \sigma)) \tag{6}$$

The RSSI values obtained from the original sampling and Gauss filter model are compared. As shown in Figure 3, the RSSI values optimized by the Gauss filter are smoother, and the abrupt data and noise fluctuations are removed in the first stage.

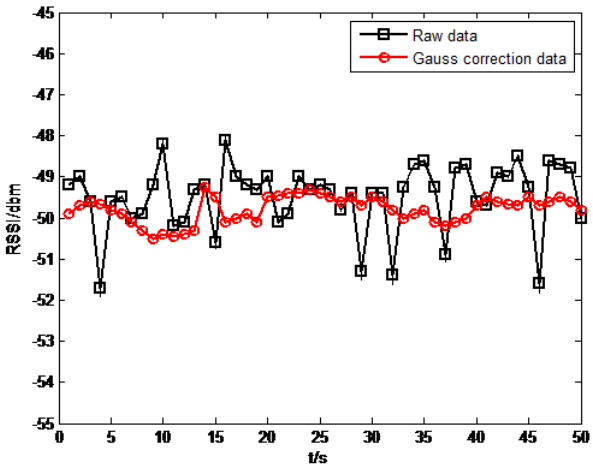


Figure 3: Comparison of Gaussian filter before and after correction

3.2 Dynamic Acquisition of Path Loss Factor k

Assuming that there are three nodes N_1, N_2, N_3, and the unknown node N_3 receives the RSSI value of the adjacent anchor nodes N_1, N_2, which can be obtained from Equation (1):

$$P(d_1) = P(d_0) - 10k\lg(\frac{d_1}{d_0}) + \xi_n \tag{7}$$

$$P(d_2) = P(d_0) - 10k\lg(\frac{d_2}{d_0}) + \xi_n \tag{8}$$

The subtraction of Equation (7) and Equation (8) is:

$$P(d_1) - P(d_2) = 10k\lg(\frac{d_1}{d_2}) \tag{9}$$

where $P(d_1), P(d_2)$ are the RSSI values between N_3 and N_1, N_2, and d_1, d_2 are the actual distances between N_3 and N_1, N_2. From Equation (9), k can only be obtained by calculating $P(d_1), P(d_2)$ and d_1, d_2, regardless of $P(d_0)$. According to the wireless signal transmission model, the actual distance between the unknown node and the nearby anchor node are used to calculate the path loss factor of the anchor node in this area in real time. Then the real-time dynamic loss factor is applied to the RSSI ranging to effectively eliminate the influence of environmental factors on the ranging accuracy and obtain accurate distance information.

3.3 Establishment of Radio Signal Transmission Model

After the RSSI values are preprocessed by Gaussian filtering, the RSSI values between the anchor nodes near the unknown nodes are measured periodically and dynamically. The path loss factor k is calculated and obtained in real time for anchor node location. The parameters are substituted into the radio transmission loss model to obtain a new signal transmission model that conforms to the actual environment.

3.4 Correction Coefficient of Distance Measurement Error $\Delta\mu$

After the signal transmission model that conforms to the actual environment is established, in order to further improve the location accuracy, based on the distance d_i, the error correction coefficient of distance measurement $\Delta\mu$ is used to participate in the calculation of location coordinates. When the anchor nodes in the communication range of the unknown nodes are k, the three anchor nodes closest to the unknown node are selected for location calculation. The coordinates of the three anchor nodes are set as follows:

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = (d_1 + \Delta\mu_1)^2 \\ (x_2 - x)^2 + (y_2 - y)^2 = (d_1 + \Delta\mu_2)^2 \\ (x_3 - x)^2 + (y_3 - y)^2 = (d_1 + \Delta\mu_3)^2 \end{cases} \tag{10}$$

Since the distance between the unknown node and the anchor node cannot avoid some errors with the actual distance, it can be concluded that $Ax + \Delta\mu = b, A = -2\begin{bmatrix} x_1 - x_3 & y_1 - y_3 \\ x_2 - x_3 & y_2 - y_3 \end{bmatrix}, b = \begin{bmatrix} d_1 - d_3 - x_1 + x_3 - y_1 + y_3 \\ d_2 - d_3 - x_2 + x_3 - y_2 + y_3 \end{bmatrix}, x = \begin{bmatrix} x \\ y \end{bmatrix}$, and $\Delta\mu$ is the dimension random error vector of $k-1$. From the matrix equation, $\Delta\mu = b - Ax$ can be obtained. The smaller the matrix equation $\Delta\mu$, the more accurate the location. The least square method can be used to obtain the estimation formula of the coordinates of the nodes $\tilde{x} = (A_g^T A)^{-1} A_g^T b$.

3.5 Unconstrained Quasi-newton Iterative Refinement of Unknown Node Coordinates

Quasi-Newton method is an efficient method to solve optimization problems due to its fast convergence speed and high location accuracy. For location calculation, the least square method is used to estimate the coordinates of unknown nodes, but its accuracy is low. Taking the estimated value as the initial value, the quasi-newton optimization algorithm is used to iteratively optimize the coordinates of unknown nodes. Thereby, the localization problem can be transformed into a non-linear least squares optimization problem, also known as the unconstrained minimal squares sum function problem [10, 17]:

$$\left.\begin{aligned} &F(x,y) = \sum_{i=1}^{n}((x_i - x)^2 + (y_i - y)^2 - d_i^2)^2 \\ &minF(x,y) \end{aligned}\right\} \tag{11}$$

The steps to solve the unknown node coordinates through the quasi-Newton method are as follows:

Step 1: The estimated value of the unknown coordinate X is obtained by the improved least square method.X is taken as the initial value of the quasi-Newton algorithm $X^{(0)}, H_0 \in R^{n\times n}, 0 \le \epsilon \le 1, i = 0$;

Step 2: If $\| g_i \| \le \| \bigtriangledown F(X^j) \| \le \epsilon$, the calculation is stopped; otherwise, continued;

Step 3: A linear search is performed along the direction d_j to obtain find a_j, and then let $x_j + 1 = x_j + a_j d_j$;

Step 4: $H_{j+1} = H_j$ is corrected to establish the quasi-Newtonian condition;

Step 5: Let $j = j + 1$, and repeat Step 2.

3.6 Algorithm Implementation Process

Three improvements are made based on RSSI location algorithm. The specific implementation process is as follows:

1) Improvement of the ranging accuracy: the anchor nodes in wireless sensor networks periodically send data with their own ID (Identity Document) and coordinate information; after receiving the information, the location nodes record their RSSI values and corresponding coordinate information; then the RSSI values are optimized by Gauss filtering and the arithmetic mean involves in the subsequent location calculation.

2) Improvement of the radio transmission model: according to Equation (7)-Equation(9), the path loss factor is calculated and obtained in real time for the location of anchor node. Then a new signal transmission model that conforms to the actual environment is obtained.

3) Improvement in the location estimation stage: the correction parameter of the distance error is used to optimize for locating coordinate calculation. Then the quasi-Newton method is used to iteratively optimize the coordinates of the estimated unknown nodes.

The improvement of the location algorithm in this paper is as shown in Figure 4:

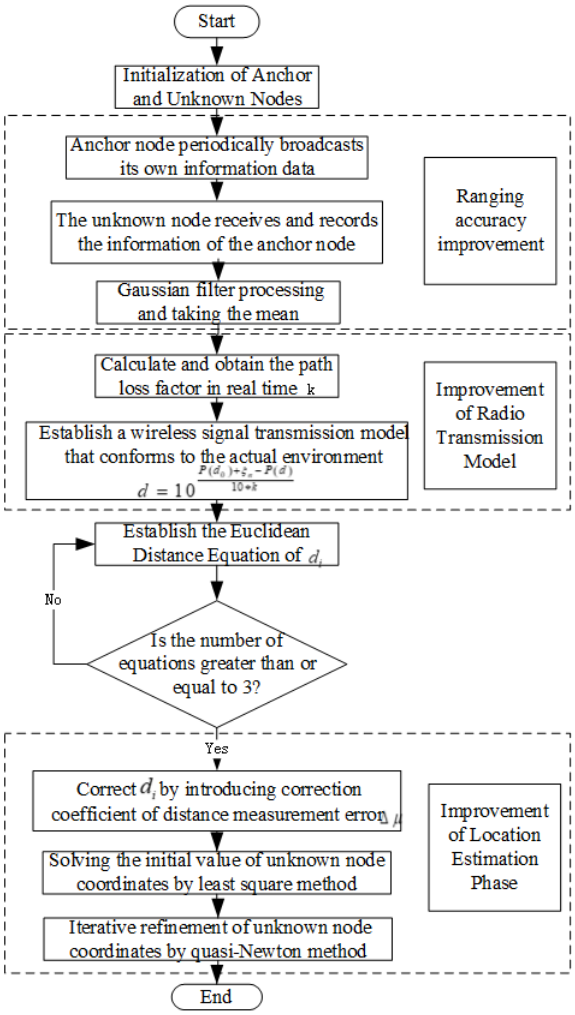


Figure 4: The flow chart of localization algorithm implementation in this paper

4 Simulation and Experiment Result Analysis

4.1 Performance Evaluation Index and Numerical Simulation Environment

To verify the location performance and validity of the proposed algorithm, as well as to compare it with the algorithm in reference [9] and [18], and the RSSI algorithm, experimental verification based on MATLAB simulation was conducted.

1) The location error E_i of node i in the network is defined as [9, 14]:

$$E_i = \frac{\| p_i - z_i \|}{R} \tag{12}$$

2) The normalized average location error of node E_{avg} is defined as:

$$E_{avg} = \frac{\sqrt{\| p_{fi} - z_{fi} \|^2}}{NR} \tag{13}$$

where $i = 1, 2, 3, \cdots, N$, and N is the number of network nodes ; R is the network communication radius; p_i is the final estimated coordinate; z_i is the real coordinate, p_{fi} is the final estimated vector, z_{fi} and is the real position vector.

The simulated environment is set as the wireless sensor nodes that randomly distributed in a square area of

$100 \times 100m^2$. The network parameters are set according to the communication radius of nodes, the total nodes and the anchor nodes. In the actual communication environment, there are some errors in RSSI ranging due to the interference of various conditions in the wireless channel, so it is necessary to simulation the random error. The standard deviation of Gauss random variable σ can be expressed as $\sigma = R\beta$. R is communication radius of the node and β is the ratio of RSSI ranging error to the node R. β can be adjusted to simulate the ranging error, and total 100 times are simulated to take the mean value. The simulation variables and parameters in this paper are shown in Table 1.

4.2 Performance Analysis and Comparison

Figure 5 is the simulation environment. Figure 6 is the deviation diagram of the improved algorithm and the original RSSI algorithm to locate unknown nodes. As show in Figure 6, the improved algorithm can greatly improve the location accuracy with better stability than the original RSSI algorithm.

Figure 7 shows the anchor nodes under ten different schemes of multiple tests to verify the effectiveness of the proposed algorithm. Compared with the traditional RSSI algorithm, the proposed algorithm can significantly improve the accuracy and stability of the location.

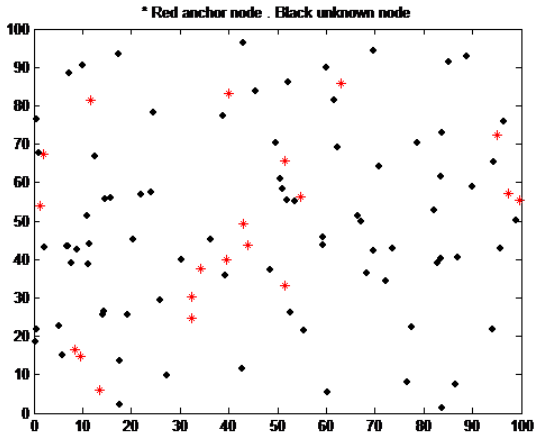


Figure 5: Node random distribution diagram

Figure 8 shows the relation diagram of the location error and anchor nodes. In the simulation environment, the total nodes are set to 100; the communication radius of nodes is 20 meters, and the variance of the ranging error σ^2 is 20. As the anchor nodes increase, the location accuracy is high and then tends to be stable. Compared with the traditional RSSI algorithm, the localization accuracy of our method is improved by about 20%. Compared with the literature [9] and the literature [18], the localization accuracy is improved by about 5%-10%.

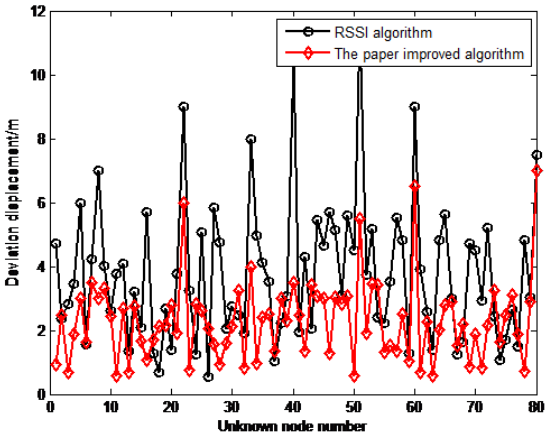


Figure 6: Unknown node location error

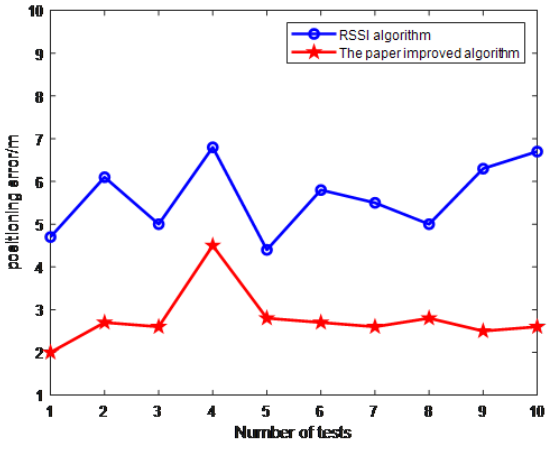


Figure 7: Comparison of location errors between two algorithms

Table 1: Simulation variables and parameters

variable	Description(parameter setting)
BorderLength	Regional scope(100 × 100)
NodeAmount	Total number of nodes(100)
SimulationTimes	SimulationTimes(100)
AnchorAmount	Anchor nodes(10,20,30,40)(100)
UNAmount	Unknown nodes
NodeAmount	Total number of nodes(100)
R	Communication radius(15,20,25,30,35,40)
k	Dynamic path loss factor
$\Delta\mu$	Correction coefficient of distance measurement error
σ	The Standard deviation of Gauss random variable

Figure 9 shows the relation diagram between the location error and communication radius of the anchor nodes. The total nodes are set to 100; the anchor nodes are 30, and the variance of the ranging error σ^2 is 15. It can be seen from Figure 9 that when the communication radius are 15m, 20m, 25m, 30m, 35m and 40m respectively, the location errors of the three algorithms decrease with the communication radius. When the communication radius is 20 meters, the normalized mean error of the RSSI algorithm is 0.302; the algorithm in the literature [9] is about 0.276, the algorithm in the literature [18] is about 0.238, and the improved algorithm is only 0.201. This indicates that as the communication radius increases, the location where the anchor nodes involve in increase, which is beneficial to improve the location accuracy.

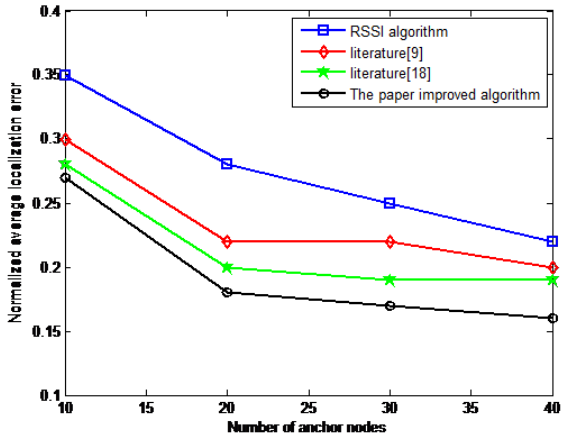


Figure 8: The relation diagram of location error and anchor nodes

Table 2 simulates the average location error of G-K-RSSI algorithm and RSSI algorithm when the communication radius is 20. It can be seen that the average location error of G-K-RSSI algorithm is smaller than that of RSSI algorithm, which does not need additional hardware and hardware overhead. Thereby it has obvious advantages in location.

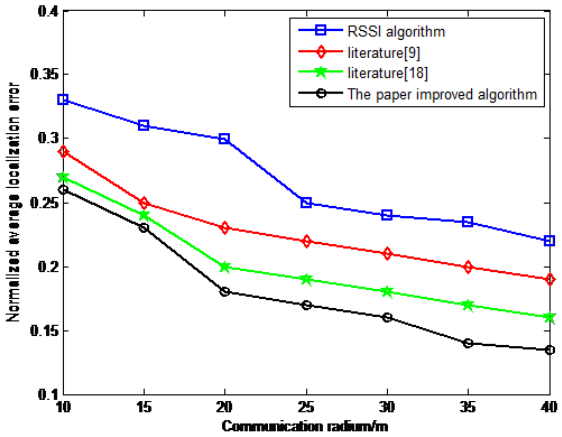


Figure 9: The relation between the location error and the communication radius of anchor nodes

Table 2: Comparison of two algorithms

anchor nodes	RSSI		G-K-RSSI algorithm	
	Error/m	NRMSE	Error/m	NRMSE
10	6.2156	0.04012	4.6682	0.01272
20	5.5126	0.02869	3.7961	0.00358
30	5.0189	0.02143	3.1862	0.00218
40	4.6225	0.01156	2.5796	0.00146

5 Conclusions

From our paper, an improved RSSI node location algorithm is proposed, and the error suppression measures are taken for each localization stage location without increasing the node hardware. First, Gaussian filter is used to preprocess the RSSI, and a new signal transmission model is obtained to improve the ranging accuracy by calculating the dynamic path loss factor in real time. Then, the error between the anchor nodes and unknown node is fully utilized, and the correction coefficient of distance measurement error $\Delta\mu$ is used to calculate the location coordinates. Finally, the position coordinates of unknown nodes are estimated iteratively by the quasi-newton method. The experimental results show that the location accuracy of the proposed algorithm is higher than that of RSSI, literature 9 and literature 18 under the same conditions. At the same time, the G-K-RSSI algorithm is applicable to many fields, such as environmental monitoring, disaster monitoring, post-disaster rescue, etc. The proposed algorithm is only verified in the simulation environment, and the environmental factors of actual applications, such as multipath effect and signal fading are not considered, which will have a great impact on the radio transmission model. It is necessary to further optimize the algorithm model in different environments, which is also our further research direction.

Acknowledgments

The work was supported by Projects of Natural Science Foundational in Higher Education Institutions of Anhui Province (2023AH052099, 2023AH052105,2023AH052104); Key Research Projects of Chaohu University (XLZ-202207); Chaohu University Quality Engineering Project (ch21yykc02); Provincial Quality Engineering Project of University in Anhui Province (2019jyxm0395); Anhui Teaching Demonstration Course (2020sfk35).

References

[1] K. Chen, "Hop distance-corrected localization algorithm in wireless sensor networks," *Modular Machine Tool & Automatic Manufacturing Technique*, vol. 12, pp. 23–26, 2021.

[2] T. F. Wang C. S. Li, Q. L. Kong and W. H. Fang, "Simulation and accuracy analysis of the improved rssi centroid location algorithm," *Journal of Navigation and Location*, vol. 10, no. 1, pp. 48–52, 2022.

[3] M. P. Deisenroth, R. D. Turner, M. F. Huber, and U. D. Hanebeck nad C. E. Rasmussen, "Robust filtering and smoothing with gaussian processes," *IEEE Transactions on Automatic Control*, vol. 7, no. 1, pp. 1865–1871, 2012.

[4] E. J. Ding, X. Qiao, and F. Chang, "Improved weighted centroid localization algorithm based on rssi differential correction," *International Journal on Smart Sensing and Intelligent Systems*, vol. 7, no. 3, pp. 1156–1173, 2014.

[5] Y. J. Guo, J. Yang, and L. Gan, "Positioning algorithm based on improved pdr and rssi fusion," *Chinese Journal of Sensors and Actuatiors*, vol. 33, no. 7, pp. 1027–1032, 2020.

[6] Q. S. Hu and D. W. Luo, "Clustering routing based on link quality estimation for disaster monitoring sensor network," *Huazhong University of Science & Technology (Natural Science Edition)*, vol. 48, no. 6, pp. 26–32, 2020.

[7] L. Ben, C. Z. Ma, and J. C. Jin, "Bayesian probabilistic localization algorithm based on rssi measurement," *Journal of HeFei University of Technology (Natural Science)*, vol. 10, no. 44, pp. 1413–1419, 2021.

[8] W. Lei and J. L. Jing, "Dv-hop location algorithm based on ranging modification and improved whale optimization," *Instrument Technique and Sensor*, vol. 2, pp. 116–121, 2022.

[9] B. Liu, C. Z. Ma, J. C. Jin, S. Z. Jin, and X.X. Li, "Bayesian probabilistic localization algorithm based on rssi measurement," *Journal of Hefei University of Technology (Natural Science)*, vol. 44, no. 10, pp. 1413–1419, 2021.

[10] L. Li, Z. H. Zhang, E. J. Ding, and L. Zhang, "Precision location algorithm in coal mine tunnel based on rssi," *Journal of China University of Mining & Technology*, vol. 46, no. 1, pp. 183–191, 2017.

[11] J. Long, L. L. Pei, S. Zhang, and Q. S. Hu, "Improved weighted centroid localization correction algorithm based on virtual beacon nodes," *Microelectronics and Computer*, vol. 34, no. 3, pp. 74–78, 2017.

[12] D. J. Luo, M. Li, and D. J. Zhang, "An improved tdoa lighting location approach with considering l-m algorithm and acoustics," *Journal of Shanghai Jiao Tong University*, vol. 56, no. 13, pp. 353–360, 2022.

[13] Y. Y. Ma, H. Z. Bian, and Q. Z. Liu, "Toa estimation of wi-fi based on indoor localization signal by waveform edge detection," *Journal of Zhengzhou University*, vol. 5, no. 3, pp. 1–6, 2022.

[14] Y. F. Ni and X. H. Shi, "Indoor staff kalman filter location algorithm based on rssi," *Journal of Xi'an University of Science and Technology*, vol. 40, no. 1, pp. 167–172, 2020.

[15] D. Niculescu and B. Nath, "Ad hoc location system (aps) using aoa," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, pp. 1734–1743, March 2003.

[16] X. Qiao and F. Chang, "Underground location algorithm based on random forest and environmental factor compensation," *International Journal of Coal Science & Technology*, vol. 8, no. 5, pp. 1108–1117, 2021.

[17] X. Qiao, F. Chang, E. J. Ding, and T. Wang, "Modifying average hopping distances based iterative algorithm for quasi-newton in wsn," *Chinese Journal of*

Sensors and Actuatiors, vol. 27, no. 6, pp. 797–801, 2014.

[18] X. Qiao, F. Chang, and J. Ling, "Improvement of localization algorithm for wireless sensor networks based on dv-hop," *International Journal of Online & Biomedical Engineering*, vol. 15, no. 6, pp. 53–65, 2019.

[19] X. Qiao, H. S. Yang, and Z. C. Wang, "Iterative lm algorithm in wsn–utilizing modifying average hopping distances," *International Journal of Online Engineering (iJOE)*, vol. 13, no. 10, pp. 4–20, 2017.

[20] A. Rana and D. Sharma, "Mobile ad-hoc clustering using inclusive particle swarm optimization algorithm," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 1–8, 2018.

[21] K. Q. Ren and C. M. Pan, "Collaborative localization algorithm based on dynamic correction of rssi model parameters," *J. Huazhong Univ. Sci. Technol.(Nat. Sci. Ed.)*, vol. 48, pp. 97–102, 2020.

[22] R. Singh and M. S. Manu, "An energy efficient grid based static node deployment strategy for wireless sensor networks," *International Journal of Electronics and Information Engineering*, vol. 7, no. 1, pp. 32–41, 2017.

[23] M. Z. Song, J. S. Qian, and Q. S. Hu, "Research on improved region division method in underground wlan location fingerprints positioning," *Industry and Mine Automation*, vol. 46, no. 3, pp. 64–68, 2020.

[24] Z. H. Wei, Y. L. Xie, S. Z. Wang, and X. Y. Ye, "Research on non-line-of-sight recognition method based on weighted k-nearest neighbor classification," *Journal of Electronics & Information Technology*, vol. 43, pp. 1–10, 2021.

[25] S. S. Xu, F. Zhou, and Y. J. Li, "New distributed location algorithm for sensor nodes," *Journal of Xidian University*, vol. 49, no. 2, pp. 1–9, 2021.

[26] J. Zhang and Y. Li, "An improved unconstrained optimization 3d-dv-hop localization," *Computer Engineering & Science*, vol. 44, no. 1, pp. 75–83, 2022.

[27] Y. H. Zhang, H. H. Chen, Y. Guo, and S. Q. Nei, "Indoor positioning algorithm based on weighted mixed dgmm filtering rssi," *International Journal of Electronics and Information Engineering*, vol. 10, no. 3, pp. 25–30, 2022.

Biography

Xin-Qiao biography. was born in 1988, She is an associate professor, She has been published a number of high-level papers and her research interests are: wireless sensor network positioning technology, signal processing etc. She is with School of Electronic Engineering, Chaohu University, Chaohu, Anhui, China.

Jing-Wang biography. She is an associate professor, Her research interests include Intelligent Control Theory and application, She is with School of Electronic Engineering, Chaohu University, Chaohu, Anhui, China.

Hai-Yang Shen biography. He is a Lecturer, He research interests include image processing and IoT applications, He is with School of Electronic Engineering, Chaohu University, Chaohu, Anhui, China.

Fei-Chang biography. He is with Huishang Futures co. LTD, Hefei, 230061,Anhui, China.

A Trust and Risk Adaptive Access Control Model for Internet of Vehicles

Pengshou Xie, Xiaoye Li, Tao Feng, Minghu Zhang, Pengyun Zhang, and Pengfei Li
(Corresponding author: Xiaoye Li)

School of Computer and Communications & Lanzhou University of Technology
No. 36 Peng Jia-ping Road, Lanzhou, Gansu 730050, China
Email: 976339400@qq.com

(Received May 19, 2023; Revised and Accepted Sept. 22, 2023; First Online Apr. 25, 2024)

Abstract

To solve the problems of unreasonable role and permission assignment and risky access behavior in role-based access control for the Internet of Vehicles, this paper proposes a trust and risk adaptive access control model based on RBAC research. In trust control, the final trust value is obtained by a weighted summation of direct, recommended, and historical trust. In risk control, the risk value of access behavior is predicted by the BP neural network, and the risk-adaptive access control is realized by reducing the amount. The experimental results show that the trust threshold will effectively influence the role assignment, the risk impacts the final trust, the expected risk has high accuracy, and the comparison verifies the usability of this improved model.

Keywords: Access Control; BP Neural Network; Internet of Vehicles; Risk Adaptive

1 Introduction

Internet of Vehicles (IOV), as an emerging industry with the deep integration of a new generation of network communication technology and automobiles, electronics, road traffic and transportation, the cybersecurity issues in its development have also received the attention of the delegates and members of the National People's Congress [12]. At the same time, the demand for data security has also brought industry opportunities. The industry generally believes that with the rapid development of intelligent networked vehicles worldwide, the connected vehicle security industry will usher in an explosive period [6]. With this, a large amount of vehicle information will be generated, while increasing the interaction between vehicle information, facing a more complex security communication environment, and the in-vehicle network architecture is vulnerable to information security challenges. In order to provide a better experience and security to users, the trust and risk adaptive access control model for the Internet of Vehicles is further investigated.

Currently, Internet of Vehicles services focus on vehicle access, device management, and business functions that meet the needs of data collection, and remote control, and provide data and information access capabilities for Internet of Vehicles users. Scholars have conducted research on the information collection and risk assessment of the IOV, and have used access control techniques to implement differentiated management of its functions by making operations such as cutting off sessions or reducing access levels through trust relationships [9]. In recent years, there has been a good research prospect for the research of security access control technology for Internet of Vehicles, which provides a safe and reliable access environment for users through trust assessment of access rights of access users and dynamic access control of data changes [20, 22].

This paper addresses the above issues and intends to establish further research on access control technology in the vehicle networking environment. By integrating trust mechanisms and risk control in role-based access control, the research results will have a theoretical contribution to vehicle networking access control and will have a propulsive effect on enhancing vehicle networking security, popularizing vehicle networking applications, and improving people's quality of life.

2 Related Theories

2.1 IOV Network Framework

IOV is conceptually derived from the Internet of Things (IoT), which takes the moving vehicle as the information-sensing object and realizes the network link between the car and the cloud platform, the car and the car, the car and the road, the car and the people, and the car and the scenario inside the car with the help of new generation information and communication technology, and integrates a large number of cameras, radars, speed detectors, navigators and other kinds of sensors to realize the transformation of the car from a simple manual driving

vehicle to an intelligent car with It has become a networked information node and data hub that deeply collects, processes, transmits and uses a large amount of personal information, car operation data and environmental data. In the process of generating and flowing a large amount of car data, the safety of the car data itself becomes the key to the safe operation of smart cars, and the important data among them extends to the safety issues involving personal information subjects, enterprises, and the state [3, 5].

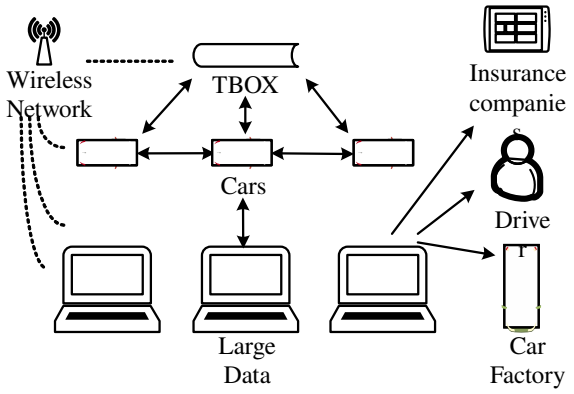


Figure 1: IOV network architecture

2.2 Information Entropy

Entropy is used as a measure of uncertainty, and information entropy is called information quantity in information theory. Information entropy can be used as the basic unit of information to describe the degree of dispersion of a random variable [7]. A higher value of information entropy entropy indicates a greater degree of dispersion of that kind of information. The information entropy of a discrete random variable is defined as (1):

$$H(X) = -\sum_{x \in X} p(x) \log_b p(x) \tag{1}$$

where the random variables take values with probability of occurrence and the base takes different values indicating that there are different ways of measuring information entropy.

The literature [21] conducted a privacy security metric based on information entropy, which describes the privacy risk environment based on Markov theory, considering the impact of multiple factors on the risk. The literature [2] measured the stability of the system based on information entropy for accessing the system with legitimate and non-legitimate user metrics.

This paper intends to use the information entropy theory to measure the degree of confusion in the selection of work targets and connected vehicle information as a basis for determining the risk level of connected vehicle user access behavior.

2.3 BP Neural Network

BP neural network is a multi-layer feed-forward neural network where the signal is propagated forward while the error is propagated backward. The BP neural network starts with feed-forward propagation, which determines the input values of the neural network. The input values are passed through the input layer to the implicit layer, where they are processed by trained implicit units, which may have more than one layer, and then passed to the output layer, which finally produces one or more output values, and the feed-forward propagation is completed. If there is an error between the output prediction and the desired result, and the error is higher than required, the feedback propagation process is entered. Feedback propagation is the reverse propagation of the resultant error from the output layer to the implied layer and then to the input layer, in which the feed-forward and feedback processes are repeated until the output error meets the error requirements through the continuous correction of the implied layer units and the final prediction is output [11]. The Levenberg-Marquardt algorithm (L-M) provides a numerical solution for number nonlinear minimization (local minimum). This algorithm can modify the parameters at execution time to achieve the advantages of combining the Gauss-Newton algorithm and gradient descent method and improve the shortcomings of both, this algorithm can increase the training speed and obtain higher accuracy when training neural networks [10].

3 Improved Access Control Model

Role-based Access Control (RBAC) differs from other access control models by adding roles between subjects and information resources to achieve the separation of subjects and information resources, and the set of subjects will not be directly mapped to the set of information resources [4, 8, 16, 17].

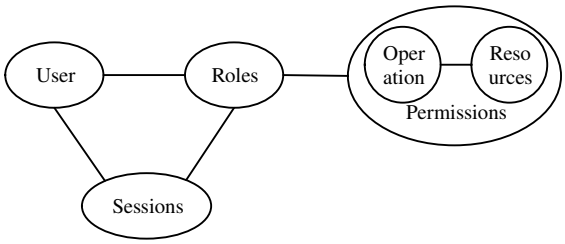


Figure 2: RBAC model

In the role assignment process, it is only necessary to match the roles of the subjects, and it is not necessary to match the access rights, because the corresponding roles correspond to the corresponding access rights of the information resources.

The literature [18]proposed a multi-factor trust assessment method based on the sub-analysis method to calculate trust values from direct trust, indirect trust, and RSU

trust, and did not consider the influence of the historical access process on trust. The literature [19] proposes a quantitative assessment method for the security risk of vehicle networking based on combination weighting, and uses a fuzzy evaluation method to establish the index evaluation set and calculate the risk interval of each index, but ignores the perfection of security risk index. In this paper, based on Figure 2, we combine trust-to-control and risk-control-related technologies, make several improvements to the model, and propose a trust and risk adaptive access control model for vehicle networking as shown in Figure 3.

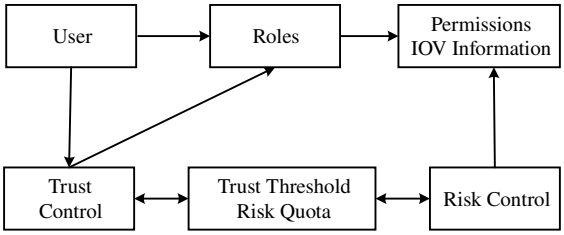


Figure 3: Improved trust and risk adaptive access control model

1) Introduction of trust control and quantification of trust

Most scholars have studied trust-based access control for the access rights phase after role assignment [13, 15]. In this model, trust control is added in the middle of the user-role assignment, and the role assignment can be performed only when the user's trust value is higher than the trust threshold required for the role he/she applied for, so as to achieve trust control in role assignment.

Relative to the calculation of trust value most of them are single, only direct trust and recommendation trust are taken into account. This paper finds that historical access risk has some influence on user trust, so the historical trust value influenced by historical risk is added to the trust value calculation, and then the direct trust, recommendation trust, and historical trust are weighted and summed to reflect the influence of risk on trust, which makes the quantification of trust more specific.

2) Introduce risk control and quantify the risk

Users have the corresponding object access rights after obtaining the role, and problems can easily occur during this period. Such as some unlawful users' ultra-authorized access and beyond the authorized read-on-write behavior can lead to a series of potential problems such as data leakage. This paper adds risk control in the middle of role and object permissions, which can identify and predict the risk of access behavior, and then develop access policies to strengthen the access control effect.

Through the influence of trust on the risk of access behavior and assuming the difference between normal and abnormal user behavior, the information entropy theory was invoked to quantify the entropy value of the user when choosing the work target and accessing the vehicle network information, and finally the user trust value, the entropy value of the user choosing the target and the entropy value of accessing the vehicle network information were identified as the influencing factors for the occurrence of risk during access. In this way, a risk prediction model based on the BP neural network was constructed to predict the risk of access behavior.

3) Adaptive access control through quota reduction

According to the special nature of big data in the realistic scenario of vehicle networking, especially in some important moments when users need to access certain more information, the special permission control cannot just stay in the control of allowing or denying absoluteness. To address this issue, by setting up risk bands as shown in Figure 4, the risk band value is generally set by the system administrator, and the risk band value is used as a hierarchical division, by calculating the risk and risk band value, different risk amounts are consumed under different risks, and then a risk-adaptive access control policy is developed to control access requests dynamically based on satisfying the minimum privilege.

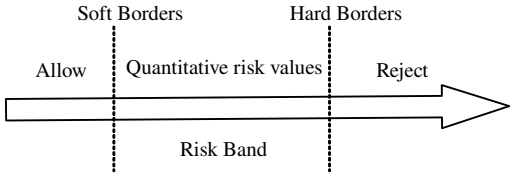


Figure 4: Diagram of risk zone

By studying a large number of trust-based access control models and risk-based access control models it is found that there are subjective and objective factors in the quantification method of trust. For example, the risk of data leakage due to the failure of vehicle systems or devices is an objective cause, while privacy leakage events due to misuse or careless management of vehicle managers are subjective. In a word, trust and risk are independent but also influence each other. This paper considers the reliability of access behavior while also considering the risk brought by access behavior, and this model is more comprehensive and scientific compared to the independent trust or risk for vehicle network access control.

4 Quantifying Trust and Risk in IOV Nodes

Trust refers to the degree of trustworthiness of the user's ability to ensure that the risk of access behavior is within controllable limits based on his direct access experience to information resources within a certain time frame and in an open network environment, also known as direct trust. In the same time frame and network environment, the degree of trustworthiness of the ability to ensure that the risk of access behavior is within the controllable range based on the recommendation information of other recommenders for both information resources can be accessed honestly, also known as indirect trust. In addition, in this paper, after studying the relationship between trust and risk, historical trust is considered in the calculation of trust value based on the fact that the access behavior of vehicle network users causes historical risk.

4.1 Trust Quantification

In the context of the era of big data gradually tends to be openly shared, there will be a variety of users accessing vehicle network data information, such as car owners, car dealerships, drivers, 4S stores, DMV, traffic police, car manufacturers, insurance companies, etc. The categories of users are more detailed and their access rights are different, and there are inevitably risks when assigning roles to users. To avoid this risk, this paper adds trust control before assigning roles to users and blocks the assignment of roles if the trust value is lower than the trust threshold.

In this paper, the final trust value is calculated using the weighted sum of direct trust value, recommended trust value, and historical trust.

1) Calculation of direct trust value DT

The direct trust value is the most basic trust. When a vehicle network access user needs to log in to the system, he/she has to provide authentication information to the administrator, and the system will calculate the direct trust value of the user through Formula (2) based on the current environment the user is in and the identity information he/she provides.

$$DT(U)=\frac{\sum D(i,u)\cdot\alpha_1+D(p,u)\cdot\alpha_2+D(e,u)\cdot\alpha_3}{\alpha_1+\alpha_2+\alpha_3} \quad (2)$$

Where $D(i,u)$is the trust correlation judged by the system administrator based on the identity information of the accessing user,$D(p,u)$is the trust correlation judged by the system administrator based on the authority applied by the user,$D(e,u)$is the trust correlation judged by the system administrator based on the network environment in which the user is located,α_1 is the weight of the user identity information in the direct trust value;α_2 is the weight of the authority applied by the user in the direct trust value;α_3 is the weight of the network environment information in which the user is located in the direct trust degree.

2) Calculation of recommended trust value RT

Recommendation trust value in judging the trust value of a user will have trust transfer recommendation, with the help of the dynamic reputation tree model, other individuals with indirect trust relationship with the subject can be constructed very clearly, while the weight between different levels can be specified according to the difference of trust difference levels between the subject and the recommended subject, and the general principle is that the closer to the subject the greater the weight of the recommended subject. It is generally judged by the recommendation information and satisfaction degree of other vehicle network users to the user, which can be described as

$$RT(U)=\sum_{k=1}^{\mathrm{n}}\left(\lambda\left(U_{\mathrm{i}}\right)\cdot DT\left(U_{\mathrm{i}},U_{\mathrm{j}}\right)\right)\cdot\frac{1}{\sum_{k=1}^{n}\lambda\left(U_i\right)} \quad (3)$$

where n is the number of indirect referrers,$\lambda\left(U_i\right)$ is the weight factor of the referrer, and $\lambda\left(U_i\right)$ is defined as according to the different levels of the referrer

$$\lambda\left(U_{\mathrm{i}}\right)=\prod_{\mathrm{m}=0}^{1}DT\left(U_{\mathrm{m}},U_{\mathrm{n}}\right) \quad (4)$$

The definition $DT(U_m,U_n)$denotes the direct trust value of U_m on its successor nodes on the trust path from U_m to U_k in the trust tree, and l denotes the level at which the recommendation weights are placed.

3) Calculation of historical trust value HT

The historical trust value contains the trust value obtained by the vehicle network users in accessing the information resources and the risk value generated during the access by the vehicle network users. The historical trust value is positively correlated with the trust value obtained by the user in accessing the information resource.

$$HT_1(U)=\frac{\sum_{i=1}^{n}H(u,j,t)\cdot vj}{n} \quad (5)$$

where $H(u,j,t)$ is the trust value of the vehicle network user u at time t when the accessed information j is accessed,$vj$is the defined coefficient for this access, and vj takes a negative value if it is an illegal access and a positive value vice versa.

Besides, the user's historical trust value is inversely correlated with the risk of its access history. When the risk of historical access records is higher, the trust value is lower; when the risk of historical access records is lower, the trust value is higher. The

time of access behavior is used as the weight, and the closer the time is, the greater the weight and the further the time is, the smaller the weight. The initial determination of the weight distribution is shown in Table 1, and the real operation can be adjusted flexibly according to the situation.

Table 1: Weight assignment

Time Period	Weights w_i
Within one month	$w_1 = 0.4$
One month to three months	$w_2 = 0.3$
Three months to six months	$w_3 = 0.2$
Six months to one year	$w_4 = 0.1$

The relationship between historical trust value and risk value is shown in Equation (6)

$$HT_2(U) = \mathrm{a}\sum_{\mathrm{i}=1}^{4} \mathrm{w_i}\overline{R_{\mathrm{t}}} \tag{6}$$

R_t denotes the average of the risk values of access behaviors occurring in the corresponding period, and a is the reconciliation factor. Then, the historical trust value is calculated as (7)

$$HT(U) = HT_1(U) + HT_2(U) \tag{7}$$

4) Calculation of the final trust value FT

The final trust value is obtained based on the weighted sum of direct trust value, recommended trust value, and historical trust value obtained from Equation (2), Equation (3), and Equation (7), and the weights of these three trust values are set as W_{DT},W_{HT},$W_{RT}$and the relationship is as in Equation (8)

$$W_{DT} + W_{HT} + W_{RT} = 1 \tag{8}$$

The final trust value is calculated as

$$FT(U) = W_{DT}DT(U) + W_{RT}RT(U) + W_{HT}HT(U) \tag{9}$$

4.2 Trust Quantification

The access behavior of a vehicle network user after being assigned a certain role can be divided into selecting a work target and accessing vehicle network information, as shown in Figure 5. The information of one access behavior of the user is noted as a triad $\langle u, o, M_u \rangle$, where $u \in U$, $o \in O$, $M_u \subseteq M$.

After the user has been given the appropriate role, the user has to choose a work objective for this visit. For example, the car dealership detects whether the vehicle owner is paying the loan on time, predicts the vehicle owner's repayment status, etc. After that, the car dealership user selects the car network information related to

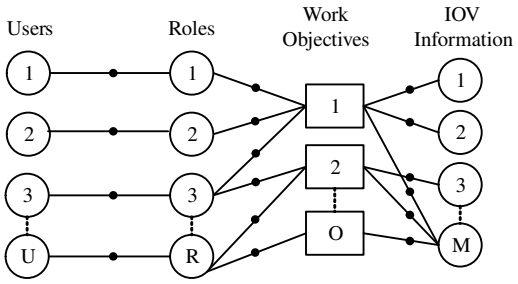


Figure 5: User access behavior

that goal to help him/her confirm the vehicle situation. For example, check the vehicle movement status and specific location, etc. Of course, if the prediction proves to be wrong, the user can select a working target again and then view the corresponding vehicle network information. The process can be repeated until the job is completed.

4.2.1 Calculation of Entropy Value

It is assumed that normal vehicle network users make access always select work targets related to actual work and access only vehicle information related to work targets, and do not regularly select work targets unrelated to actual work or frequently access vehicle information unrelated to work targets. Anomalous users will frequently select job targets that are not related to the actual job, or frequently access vehicle information that is not related to the job target.

To describe the model formally, this paper introduces the notation shown in Table 2.

For the stage of user selection of the work target, the entropy value of the user at the time of selecting the work target is calculated based on the history of the user's visits.The probability that user u chooses job target o is

$$p_u(o) = \frac{f\left(SO_u, o\right)}{\sum_{t \in SO_u} f\left(SO_u, o\right)} \tag{10}$$

Definition 1 Work goal entropy

The job goal entropy describes the degree of confusion in the job goal chosen by the user, formally described as

$$H^o(u, o) = -\sum_{o \in SO_u} p_u(o) \log\left(p_u(o)\right) \tag{11}$$

The probability that the user selects the car network information r for a given work objective o in this stage of selecting car network data information is

$$p_{u,o}(r) = \frac{f\left(SM_u^o, r\right)}{\sum_{r \in SM_u^o} f\left(SM_u^o, r\right)} \tag{12}$$

Definition 2 Vehicle Network Information Entropy

Vehicle information entropy describes the level of confusion in the user's selection of vehicle information for a particular job objective. Formally, it is described as

Table 2: Description of symbols

Symbols	Meaning
U	Collection of users
O	Collection of work objectives
M	A collection of IOV information records
$f(M,e)$	The number of occurrences of element e in the set M
SO_u	A collection of work objectives selected by the user u
SM_u^o	The user u selects the set of car network information recorded under the given work objective o

$$H^R(u,r) = -\sum_{t \in SM_u^0} P_{(u,o)}(r) \log\left(p_{o,t}(r)\right) \tag{13}$$

This paper introduces risk into the access control model to greatly reduce the losses incurred by malicious access, quantifying risk by introducing the concept of entropy benchmark. Risk is always present, and in risk quantification risk greater than the benchmark will have an impact on their access behavior, which can also be understood as the benchmark is the risk threshold. Suppose that users are divided into two categories according to the magnitude of access behavior risk: normal users and abnormal users. The entropy values of the two types of users' access behavior obey two normal distributions with different means, and a threshold can be determined as a benchmark based on the weighted average of their means.

By analyzing users' work target selection and vehicle information selection, it is assumed that normal users always select targets and information corresponding to their roles when selecting work targets and accessing vehicle information, and rarely access irrelevant targets and information. In contrast, abnormal users will select work targets and vehicle information more often to access targets and information that are not related to their roles. The normal user selects the work target more clearly and accesses the vehicle information more singularly. In contrast, abnormal users choose multiple work targets or access multiple information about vehicles, so abnormal users have more confusion in choosing work targets and vehicle information. This paper measures this level of confusion by information entropy and uses it as a basis for risk quantification.

4.2.2 Build BP Neural Network

In this paper, the inputs to the neural network are three factors, user trust value, work goal entropy, and vehicle network information entropy, which are the key factors affecting the risk size of user access behavior, so the output value is the risk value of access behavior. The three elements are preprocessed and normalized in the range of [0,1]. The number of nodes in the input layer of the BP neural network is 3. There is no certain formula to quantify the number of hidden layers, generally, a single hidden layer is needed when learning a continuous function, and only two hidden layers are needed when learning a discontinuous function. In this paper, the number of hidden layers of the neural network is set to 1. This results in a three-layer neural network with input, hidden, and output layers. The L-M algorithm is based on the Gauss-Newton method with the addition of a variable factor, and the derivation of the Gauss-Newton method is shown in the following equation.

$$f(x+h) \approx L(\mathrm{h}) = f(x) + J(x)h$$

$$F(x+h) \approx L(\mathrm{h}) = \frac{1}{2}L(h)^T L(h)$$

$$= \frac{1}{2}f^T f + h^T J^T f + \frac{1}{2}h^T J^T J h$$

$$= F(x) + h^T J f + \frac{1}{2}h^T J^T J h$$

Thus the minimum transformation of finding $F(x+h)$ is given by

$$hgn = \arg\min\{L(h)\}$$

$$L'(\mathrm{h}) = J^T f + J^T J h$$

$$L''(h) = J^T J$$

make $L'(\mathrm{h}) = 0 \Rightarrow \left(J^T J\right)\ \mathrm{hgn}\ = -J^T \mathrm{f}$
J is a full rank matrix

$$\Rightarrow hgn^T F'(x) = hgn^T \left(J^T f\right)$$

$$= -hgn^T \left(J^T J\right) hgn < 0$$

That is hgn as the direction of descent of F.

The L-M algorithm is similar to the Gauss-Newton algorithm and is described as follows

$$\left(J^T J + \mu I\right) \mathrm{h_{lm}} = -\mathrm{g} \text{ with } \mathrm{g} = \mathrm{J}^T f \text{ and } \mu \geq 0$$

1) Degenerate to the Gauss-Newton algorithm when $\mu = 0$;

2) When μ is large,$\mathrm{h}_{lm} = -\left(F'(\mathrm{x})\right)/\mu$ degenerates to the gradient descent method with a smaller step size.

The focus of the L-M algorithm is how to determine the μ value, introduce an evaluation quantity

$$e = \frac{F(\mathrm{x}) - F\left(x + h_{lm}\right)}{L(0) - L\left(h_{lm}\right)}$$

This quantity describes the degree of approximation of the decline in F using the decline in L.

With the above algorithm it is possible to train the training sample data and decide whether the current learning is finished according to whether the error reaches the corresponding accuracy requirement.

5 Trust and Risk Control Strategies

When a vehicle network user requests access, the trust control center will first calculate the final trust value of the current user and determine whether the final trust value of the user is greater than the trust threshold, and if it is greater than the corresponding role is assigned, otherwise the role assignment and access are denied.

In the process of risk control, the quantification of risk values is achieved by establishing a risk prediction model, which in turn requires the development of access control policies to achieve adaptive access control to meet the access requirements in line with the big data of Internet of vehicles. In this paper, adaptive access control is achieved by establishing a risk amount to consume the risk amount.

The trust and risk control strategy is shown in Figure 6. The user of Telematics will request the corresponding access privileges after obtaining the role. The risk value of the access behavior is calculated according to the user's trust influence, work goal entropy, and Telematics information entropy, and the risk amount is changed in the way of " slowly increasing and decreasing" to make the risk assessment more rationalized. According to the risk value calculation and different levels of risk quota reduction, different privileges are given to the current user. If there is any remaining risk level, access will be assigned accordingly; if the risk level is exhausted, access will be denied.

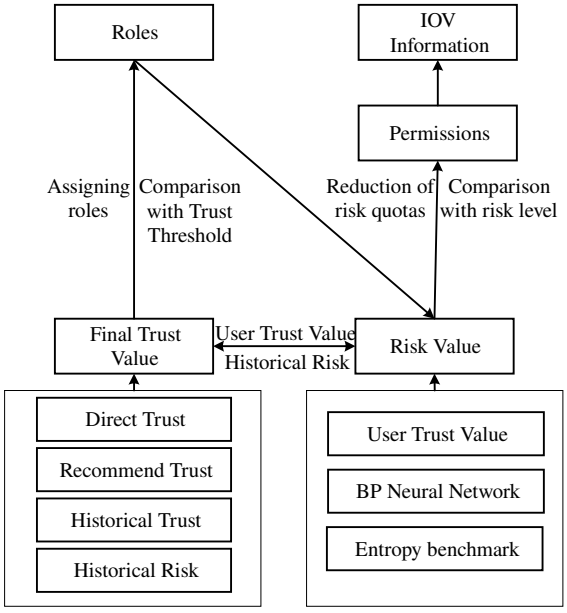


Figure 6: Trust and risk control strategies

6 Experiments and Analysis of their Results

6.1 Experimental Environment and Description

Software: conducted by using veins as an open source framework as well as OMNET++ as a network simulator and SUMO as a traffic simulator, SUMO is able to simulate vehicle motion states more realistically and is widely used in vehicle networking research [1, 14].

Hardware: Intel(R) Core i5-6200U CPU @2.30 GHz processor, 8 GB RAM, Microsoft Windows 10 Professional operating system.

The maps in this experiment simulation process use OpenStreetMap to import with realism, use SUMO's own netconvert tool to convert osm files to road files (net.xml), use radomTrips.py and duarouter tool to generate traffic flow files (rou.xml), and then generate vehicle movement OMNET++ queries and schedules the vehicle movement status through TraCI.

After a large number of simulations, this paper gets 600 groups of visitors' historical access records, including 10% of abnormal visitors, and makes statistics by calculating the trust value and risk value of each user's access. This experiment sets the trust value range between 0 and 1, the initial trust value is 0.5, and the trust threshold is 0.3, in the case the initial trust value is greater than the trust threshold, the Internet of Vehicles information access users are given the appropriate role, while having the corresponding rights to generate access will also then generate risk.

6.2 Analysis of Experimental Results

In the study of role-based access control models for vehicle networking, many scholars are developing their own research through the perspective of trust assessment. This paper is to introduce the characteristics of trust and risk interactions and to further study the connected vehicle access control under their mutual influence.

1) Trust analysis in the interaction process

The final comprehensive trust value of this experiment consists of direct trust value, recommended trust value, and historical trust value, which in turn includes the impact of historical access behavior trust and historical risk.

In this experiment, 20 sets of sample data are randomly selected for simulation, and the sample data

are shown in Table 3, which describes the risk value and trust value of the samples.

In the calculation process, it can be found that the risk is generated immediately after the IOV user generates access, and after some access operations it is found that the trust value is changing in the process of changing risk.

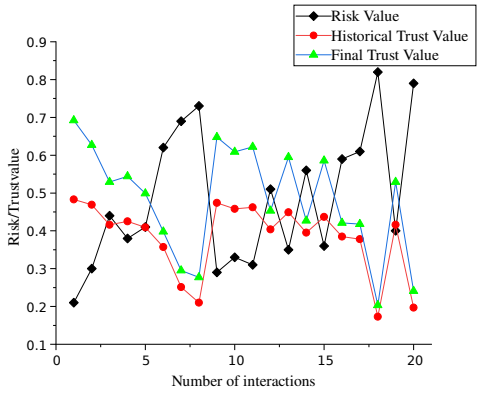


Figure 7: Relationship between trust and risk variables

From the analysis of Figure 7, it can be found that when the access risk value increases continuously during the access process, the corresponding historical trust value decreases and the final trust value decreases, indicating that the greater the risk the lower the trust and the higher the trust the lower the risk, confirming that risk and trust affect each other and the impact of risk in the access process cannot be ignored.

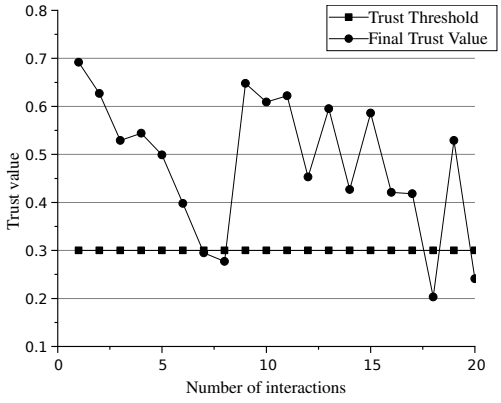


Figure 8: Comparison of trust values for access interactions

The comparative analysis of the experimental data in Figure 8 shows that the initial trust value set in the early stage is greater than the trust threshold and the corresponding roles can be assigned to the users accessing the Internet of Vehicles information, and the role is created while the access interaction also starts, thus generating the risk and trust value. After calculating the final trust value, it is found that most of the access users with a final trust value greater than the trust threshold are allowed to access, and very few access users with a final trust value less than the trust threshold will be denied access.

2) Interaction process risk analysis

The 100 sets of access records generated during the simulation were selected during this experiment, including 10 % of abnormal access records, and the rest were normal accesses. The risk values of normal and abnormal accesses vary with the number of interactions.

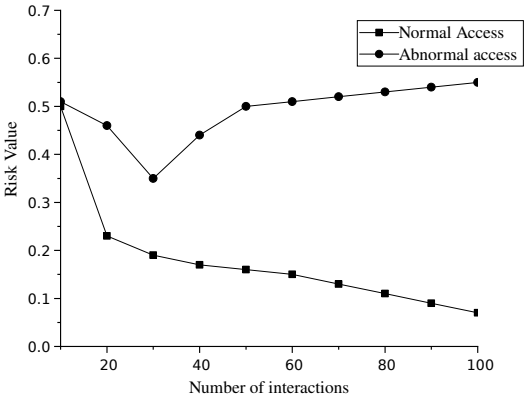


Figure 9: Trend of normal/abnormal access risk value

As can be seen from the analysis of Figure 9, the risk value decreases continuously and relatively quickly with the increase in the number of interactions in normal access. Starting abnormal access operations in order to access more system resources (disguise), the risk value of access shows a decreasing trend, but its access behavior with the increase in the number of access interactions, will launch an attack on the system, and the risk value is rapidly increasing trend.

3) Risk prediction and error analysis

In this paper, 300 sets of data as comprehensive and representative as possible were selected as training samples and 45 sets as test samples. In the model, it was determined that a three-layer BP neural network was used, and the trinlm training function was used after several training sessions, with an accuracy requirement of le-4.

As can be seen from the analysis in Figure 10, the actual output risk value and the expected output risk value almost overlap during the experiment, indicating that the error between the actual risk value and the expected risk value is extremely small in accordance with the expected requirements, and the BP neural network has high accuracy in calculating the risk value under the trinlm training function.

Table 3: Results of sample trust value calculation

Access Serial Number	Initial Trust Value	Time Weighting	Risk Value	History Trust value	Final Trust value
1	0.5	0.1	0.21	0.483	0.692
2	0.5	0.1	0.30	0.469	0.627
3	0.5	0.3	0.44	0.416	0.529
4	0.5	0.2	0.38	0.425	0.544
5	0.5	0.2	0.41	0.409	0.499
6	0.5	0.4	0.62	0.357	0.398
7	0.5	0.4	0.69	0.251	0.295
8	0.5	0.4	0.73	0.210	0.277
9	0.5	0.1	0.29	0.474	0.648
10	0.5	0.2	0.33	0.458	0.609
11	0.5	0.1	0.31	0.462	0.622
12	0.5	0.3	0.51	0.404	0.453
13	0.5	0.1	0.35	0.449	0.595
14	0.5	0.3	0.56	0.395	0.427
15	0.5	0.1	0.36	0.437	0.586
16	0.5	0.3	0.59	0.385	0.421
17	0.5	0.3	0.61	0.378	0.418
18	0.5	0.4	0.82	0.173	0.203
19	0.5	0.2	0.40	0.416	0.529
20	0.5	0.4	0.79	0.197	0.241

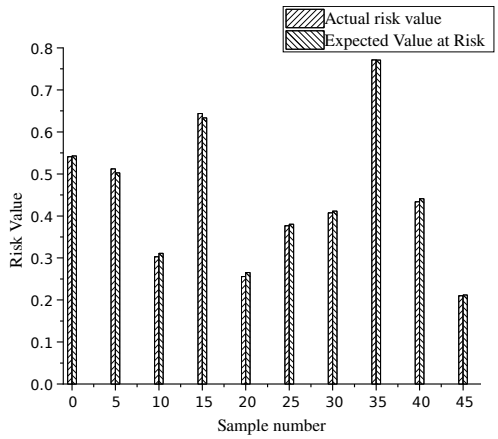


Figure 10: Actual risk value versus expected risk value

7 Conclusions

In response to the shortcomings of role-based access control research for the Internet of Vehicles, this paper proposes a trust and risk adaptive access control model for the Internet of Vehicles. This paper focuses on the relationship between trust and risk, and trust control of roles when they are assigned reduces the risk of illegal access by unknown identity users to a certain extent, and quantifying the risk when the access behavior occurs can achieve effective control of malicious access. The experimental results show that trust and risk interact with each other, the setting of trust value can effectively control role assignment, there is a significant change of risk value during normal and abnormal interaction, and the trinlm training function has a high accuracy in the calculation of risk value. In future research, the setting of initial trust value and trust threshold can be more objective, and the consideration of risk factors can be more comprehensive, which makes the research more practical application value.

Acknowledgments

This research is supported by the National Natural Science Foundations of China under Grants No.61862040 and No.62162039.The authors gratefully acknowledge the anonymous reviewers for their helpful comments and suggestions.

References

[1] I. A. ALjabry and G. A. Al-Suhail, "A survey on network simulators for vehicular ad-hoc networks (vanets)," *Int. J. Comput. Appl*, vol. 174, no. 11, pp. 1–9, 2021.

[2] Y. Cui, K. Wang, J. Hu, W. Zhao, L. Feng, and J. Cui, "Compromised accounts detection based on information entropy," *International Journal of Network Security*, vol. 23, no. 3, pp. 401–411, 2021.

[3] L. Elmoiz Alatabani, E. Sayed Ali, R. A. Mokhtar, R. A. Saeed, H. Alhumyani, and M. Kamrul H.,

"Deep and reinforcement learning technologies on internet of vehicle (iov) applications: Current issues and future trends," *Journal of Advanced Transportation*, vol. 2022, 2022.

[4] M. B. Gunjal and V. R. Sonawane, "Multi authority access control mechanism for role based access control for data security in the cloud environment," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 2s, pp. 250–264, 2023.

[5] A. Hbaieb, S. Ayed, and L. Chaari, "A survey of trust management in the internet of vehicles," *Computer Networks*, vol. 203, p. 108558, 2022.

[6] Kexun He and Baotian L., "Automotive v2x communication security key technology and test method research," in *2022 7th International Conference on Cyber Security and Information Engineering (ICCSIE)*, pp. 40–43. IEEE, 2022.

[7] R. Jiang, S. Han, Y. Yu, and W. Ding, "An access control model for medical big data based on clustering and risk," *Information Sciences*, vol. 621, pp. 691–707, 2023.

[8] A. Kesarwani and P. M. Khilar, "Development of trust based access control models using fuzzy logic in cloud computing," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 1958–1967, 2022.

[9] A. J. Khan and S. Mehfuz, "Fuzzy user access trust model for cloud access control," *COMPUTER SYSTEMS SCIENCE AND ENGINEERING*, vol. 44, no. 1, pp. 113–128, 2023.

[10] P. Liao, "Bp neural network computer network information security risks and solutions," in *2021 International Conference on Big Data Analytics for Cyber-Physical System in Smart City: Volume 1*, pp. 1305–1309. Springer, 2022.

[11] Y. Liu, M. Xiao, Y. Zhou, D. Zhang, J. Zhang, H. Gacanin, and J. Pan, "An access control mechanism based on risk prediction for the iov," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–5. IEEE, 2020.

[12] C. Ma, H. Zhao, and T. Wang, "Research on cyber security risk of telematics box in intelligent connected vehicle," in *MATEC Web of Conferences*, vol. 355, p. 03030. EDP Sciences, 2022.

[13] H. Ouechtati, N. B. Azzouna, and L.B. Said, "A fuzzy logic based trust-abac model for the internet of things," in *Advanced Information Networking and Applications: Proceedings of the 33rd International Conference on Advanced Information Networking and Applications (AINA-2019) 33*, pp. 1157–1168. Springer, 2020.

[14] K. Raja Kumar, N. Karyemsetty, and B. Samatha, "Performance analysis of vehicular network scenarios using sumo and ns2 simulators," in *Data Engineering and Communication Technology: Proceedings of ICDECT 2020*, pp. 337–344. Springer, 2021.

[15] M. R. Salji, N. I. Udzir, M. I. H. Ninggal, N. F. M. Sani, and H. Ibrahim, "Trust-based access control model with quantification method for protecting sensitive attributes," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, 2022.

[16] W. Sun, X. Yuan, and H. Su, "Role-engineering optimization with user-oriented cardinality constraints in role-based access control," *International Journal of Network Security*, vol. 23, no. 5, pp. 845–855, 2021.

[17] K. Vijayalakshmi and V. Jayalakshmi, "A study on current research and challenges in attribute-based access control model," *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2021*, pp. 17–31, 2022.

[18] P. S. Xie, X. Tong, H. Wang, Y. Zhao, T. Feng, and Y. Yan, "A trust assessment mechanism of the iov based on multi-factor analytic hierarchy process," *International Journal of Network Security*, vol. 24, no. 3, pp. 482–492, 2022.

[19] P. S. Xie, L. Wang, S. Wang, Y. Zhao, T. Feng, and Y. Yan, "A quantitative assessment method for security risk of iov based on combination weighting," *International Journal of Network Security*, vol. 24, no. 2, pp. 296–304, 2022.

[20] P. S. Xie, X. Q. Wang, X. J Pan, Y. F. Wang, T. Feng, and Y. Yan, "Blockchain-based trust evaluation mechanism for internet of vehicles nodes," *Int J Netw Sec*, vol. 23, no. 6, pp. 1065–1073, 2021.

[21] M. Yang, L. Jia, T. Gao, T. Zhang, and W. Xie, "Research on privacy security steady stateevaluation model of mobile application based on information entropy and markov theory," *International Journal on Network Security*, vol. 23, no. 5, pp. 807–816, 2021.

[22] B. Yu, X. Tai, and Z. Ma, "The study on attribute and trust-based rbac model in cloud computing," *Computer Engineering and Applications*, vol. 56, no. 9, pp. 84–92, 2020.

Biography

Pengshou Xie was born in Jan. 1972. He is a professor and a supervisor of master student at Lanzhou University of Technology. His major research field is Privacy Protection, Security on Internet of Vehicles, Security on Industrial Internet. E-mail: xiepsh_lut@163.com.

Xiaoye Li was born in Oct. 1995. She is a master student at Lanzhou University of Technology. Her major research field is network and information security. E-mail: 976339400 @qq.com.

Tao Feng was born in Dec. 1970. He is a professor and a supervisor of Doctoral student at Lanzhou University of Technology. His major research field is modern cryptography theory, network and information security technology. E-mail: fengt@lut.cn.

Minghu Zhang was born in Dec.1986. He is an associate professor and a supervisor of Master student at Lanzhou

University of Technology. His major research field is computer vision, Internet of things, network and information security technology. E-mail: zhangmh@lut.edu.cn.

Pengyun Zhang was born in Dec. 1999. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail:2324327226 @qq.com.

Pengfei Li was born in May. 1999. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 3037897010 @qq.com.

The Detection and Prevention of Network Illegal Intrusion Vulnerability under Legal Supervision

Xia Li

(Corresponding author: Xia Li)

Henan Institute of Economics and Trade
Zhengzhou 450000, China
Email: lix_xial@outlook.com

(Received Jan. 26, 2023; Revised and Accepted Dec. 13, 2023; First Online Apr. 25, 2024)

Abstract

There are still many shortcomings in detecting network illegal intrusion vulnerabilities under legal supervision. Firstly, this paper briefly analyzed the legal regulation of vulnerability detection. Then, for the purpose of vulnerability detection, the Glove+ELMo was used to realize the stacked embedding of word vectors. A convolutional neural network+bidirectional gated recurrent unit (CNN+BiGRU) model was designed to acquire local and global features, and the probability distribution was obtained through the softmax layer to realize vulnerability detection. Experiments were carried out on synthetic and actual vulnerability datasets. The results showed that compared with word2vec, Glove, and ELMo, the word vector obtained by Glove+ELMo obtained a better vulnerability detection effect. Compared with the single model TextCNN, BiGRU, and CNN+bidirectional long short-term memory (BiLSTM), the accuracy, F1, and Matthews correlation coefficient of CNN+BiGRU was 94.81%, 94.57%, and 90.21% respectively, which were better than Devign and other existing methods. The results prove the reliability of the CNN+BiGRU model. Some suggestions were proposed for the prevention of vulnerabilities under legal supervision.

Keywords: Illegal Intrusion; Legal Supervision; Network Security; Vulnerability; Vulnerability Detection; Vulnerability Prevention

1 Introduction

Network vulnerability refers to the weaknesses and defects that threats in the network system may exploit. The vulnerability itself will not cause damage, but once it is not found, it may be used and abused, resulting in the illegal invasion of the network [6]. With the development of technology, the research on vulnerability mining and detection is also deepening [19]. Cheng *et al.* [3] designed a method that integrates environment information and binary code information and used graph embedding and deep neural networks to realize the detection of vulnerabilities. They found through experiments that the accuracy of this method was more than 80% on the real dataset.

Khanh *et al.* [4] designed a software vulnerability detection approach based on a long short-term memory (LSTM) model to learn the semantic and syntactic features of the code automatically. They found that the method performed well in the experiments. Chao *et al.* [15] designed an Android application vulnerability detection technique combining dynamic and static analysis. Through experiments, it was found that this method effectively improved the accuracy.

Alenezi *et al.* [2] designed an automatic vulnerability detection method based on character n-gram embedding technology, evaluated the effectiveness of this method in detecting four kinds of vulnerability, and obtained excellent performance. However, in this process, there are also some differences in the views of vulnerability mining and detection.

Driven by economic interests, vulnerability detection has become a tradable industry. It is also common to maliciously disclose vulnerabilities to obtain illegal benefits. At present, there are still many uncertainties in the legal regulation of the detection and disclosure of vulnerabilities.

Under legal supervision, this paper studied the detection and prevention of network illegal intrusion vulnerabilities, designed a convolutional neural network+bidirectional gated recurrent unit (CNN+BiGRU) vulnerability detection method based on Glove+ELMo word embedding, and proved its effectiveness through experimental analysis. This paper aims to provide a new and effective approach for vulnerability detection in enterprises and manufacturers to improve network security and achieve safer enterprise and individual networks.

2 Legal Regulation of Network Illegal Intrusion Vulnerability Detection

Due to some reasons in system design, development and testing, the existence of vulnerabilities is common and long-term, and they have become an essential problem in network security [11]. Through the vulnerability, the attacker can realize the illegal invasion of the network without authorization [14].

In recent years, with the economy and technology advancing constantly, coupled with the growing prevalence of vulnerabilities, the mining and detection of vulnerabilities has gradually formed an industrialized market. Due to the differences in policies, laws, and regulations of various countries, the vulnerability industry is slowly becoming specialized and fragmented, and the demand for its industrial regulation is becoming more and more urgent.

However, when network vulnerabilities are still exploding, the deficiencies at the system level are becoming more and more obvious. There is a lack of perfect legal norms for vulnerability reporting, and there are many difficulties in combating the underground vulnerability trading market.

In 2016, the Cyber Security Law clearly involved the legality of network security testing for the first time, and some regulations for vulnerability detection were proposed, but the operability was not strong. Moreover, the relevant provisions in the Criminal Law can only be used to judge the illegal degree of vulnerability detection behavior.

Articles 285, 286, and 287 all have relevant provisions for crimes carried out by using computers and the Internet. However, there are many places that need to be further clarified in judicial interpretation and practice, and there are still many deficiencies in the conviction and sentencing of malicious disclosure or sale of vulnerabilities. At present, the non-malicious vulnerability mining or disclosure is not clear.

White hat hackers, who are the main force in combating illegal network invasions, may potentially face charges of illegal access to computer data. The current law for security research exceptions still lacks relevant provisions.

For malicious and illegal vulnerability detection, in addition to further strengthening industry norms and multi-party governance, it is necessary to pay attention to legal supervision. It is necessary to standardize and legalize the means and tools of vulnerability detection from the legal level, guide and encourage disordered vulnerability detection to the legal, and establish a standardized and orderly vulnerability trading market to achieve healthy development.

3 Vulnerability Detection Algorithm Based on Source Code

3.1 Word Vector Model

This research primarily focuses on the vulnerability detection algorithm based on source code [12] because by using the source code, the system can be found in the development phase of the system in time. The source code contains many statements unrelated to the vulnerability. After preprocessing, the code slice can be obtained, and the next step of work can be carried out. Codes look like natural language. To convert them into symbols that algorithms can recognize, it needs to be represented by word vectors. At present, the commonly used word embedding models are shown below.

1) word2vec model
word2vec can be divided into skip-gram and continuous bag of words (CBOW) models [13]. The principle of CBOW is to use contextual words to predict context, which is more suitable for small datasets. In this paper, the CBOW model in word2vec is chosen for research. Figure 1 illustrates the structure of the CBOW model.

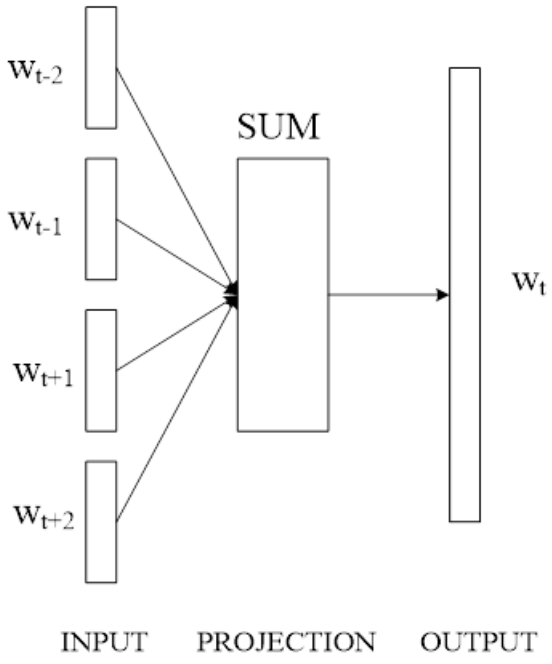


Figure 1: CBOW structure

CBOW takes the words in the context window of a word as input, averages them to obtain a vector to represent the context information, inputs it into a neural network for prediction, and updates the result according to the error between the prediction result and the actual target vector. After repeated iterations, the high-quality word vector representation is obtained.

2) Glove model
Glove is a method that uses a word-word co-occurrence matrix to train word vectors [8]. Let X_{ij} denote the number of times that $word_j$ appears in the context of $word_i$ in the corpus, then the co-occurrence probability of two words can be written as:

$$p_{ij} = \frac{x_{ij}}{x_i}$$

Then, the co-occurrence probability is used to reflect the correlation between different words:

$$ratio(w_i, w_j, w_k) = \frac{p_{ik}}{p_{jk}}$$

The loss function for the model is written as:

$$\begin{aligned} J &= f(X_{ij})[w_i^T w_j + b_i + b_j - \log(X_{ij})]^2 \\ f(x) &= \begin{cases} (\frac{x}{x_{\max}})^\alpha, & x < x_{\max} \\ 1, & x \geq x + \max \end{cases} \end{aligned}$$

where b_i and b_j are bias terms.

3) ELMo model
Compared with word2vec and Glove, ELMo [18] can obtain dynamic word vectors, which realizes model training based on double-layer bidirectional LSTM (BiLSTM). For sequences $t_1, t_2, \cdots, t_N$, word vectors $\overrightarrow{h}_{N,1}^{LM}$, $\overleftarrow{h}_{N,1}^{LM}$, $\overrightarrow{h}_{N,2}^{LM}$, and $\overleftarrow{h}_{N,2}^{LM}$, can be obtained after double-layer BiLSTM training. The final word vector can be obtained after weighted fusion:

$$ELMo_i^{task} = \beta^{task} \sum_{j=0}^{L} s_j^{task} h_{i,j}^{LM}$$

where β^{task} is the coefficient related to downstream tasks and s_j^{task} is the coefficient of softmax.

3.2 CNN+BiGRU-based Vulnerability Detection Model

3.2.1 Overall Structure of the Model

The current vulnerability detection approaches based on source code often only consider part of the features, i.e., the extraction of code features is not sufficient. Therefore, in the design of a vulnerability detection model in this paper, Glove and ELMo stacked embedding are used, and the dynamic and static word vectors are used as input. Moreover, a text convolutional neural network (TextCNN) [7] is used to obtain the local feature of the code, and BiGRU [17] is used to obtain the global feature of the code. The two outputs are combined, and the final vulnerability detection results are obtained by processing the output with the fully connected and softmax layers.

Figure 2 illustrates the overall structure of the model.

According to Figure 2, assume that the word vector obtained by Glove is w_g and the word vector obtained by ELMo is w_e. After stacking, the input of the CNN+BiGRU model is obtained:

$$Input = [w_g, w_e].$$

3.3 Feature Extraction Part

Taking the word vectors obtained by stacking in the the preceding stage as input, the convolutional layer of TextCNN is calculated:

$$F = f(I \odot W + b),$$

where I is the input of the convolution layer, W is the convolution kernel, $\odot$ is the convolution operation, b is the bias, f is the activation function, i.e., the ReLU function:

$$ReLU(x) = \max(0, x).$$

Max pooling is used in the pooling layer:

$$Z_{\max} = \max(Z_i),$$

where Z_i is the i-th feature map. BiGRU is used to achieve global feature extraction. At time t, the output of BiGRU is composed of forward GRU and backward GRU:

$$\begin{aligned} \overrightarrow{h_t} &= GRU(x_t, \overrightarrow{h_{t-1}}, \\ \overleftarrow{h_t} &= GRU(x_t, \overleftarrow{h_{t-1}}, \\ h_t &= |\overrightarrow{h_t}, \overleftarrow{h_t}|. \end{aligned}$$

where x_t is the input of the hidden layer, $\overrightarrow{h_{t-1}}$ and $\overleftarrow{h_{t-1}}$ are the forward and backward hidden layer states at time $(t-1)$.

3.4 Fully Connected and Softmax Layers

Firstly, the fusion feature is obtained by combining the local feature obtained by TextCNN and the global feature obtained by BiGRU in a Concat way:

$$F = F_P \oplus F_G$$

where $\oplus$ is the Concat operation.

Fused feature F is fed to the fully connected layer and fused with dropout at the fully connected layer to alleviate overfitting:

$$Y = Dropout(w \cdot F + b),$$

where Y is the output of the fully connected layer, w and b are the weight and bias of this layer.

Finally, the softmax layer is used to obtain the probability distribution:

$$\hat{Y} = softmax(w \cdot Y + b),$$

where w and b are the weight and bias of this layer.

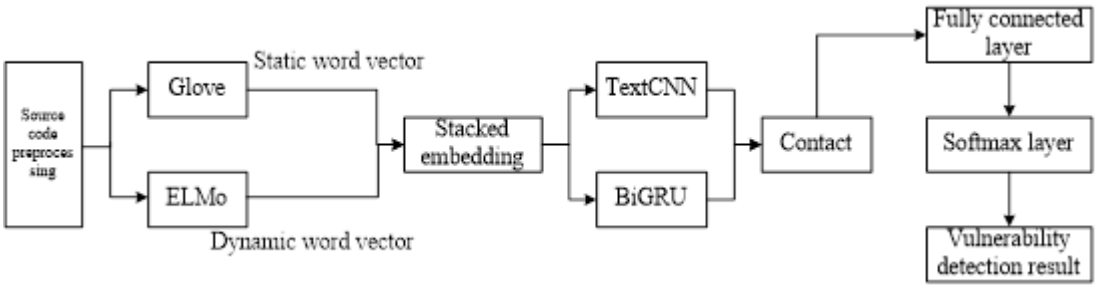


Figure 2: The structure of the CNN+BiGRU-based vulnerability detection model

4 Results

4.1 Experimental Setup

The experiment was carried out in a Windows 10 environment, with CPUNVIDIA RTX3060 i7-11800H and 32 G memory. Python3.6 development language was used, and the model was implemented based on the PyTorch framework. In the experiment, the Adam function was used for training, and rectified linear unit (ReLU) and Tanh activation functions were used for CNN and BiGRU, respectively. The remaining parameters are set as shown in Table 1.

Table 1: CNN+BiGRU parameter settings

Parameter	Value
Batch size	128
Number of iterations	30
Learning rate	0.0001
Word vector dimension	300
Dropout	0.5

Six different C/C^{++} source code datasets were selected from SARD (https://samate.nist.gov/SARD/) and the NVD vulnerability database (https://nvd.nist.gov/). Four were synthetic datasets, and the other two were from the real world (Table 2).

The source code was preprocessed, the length of the code slice was 300, and the training set and the test set were divided according to 4:1. The assessment of vulnerability detection results is based on the following indicators:

1) Accuracy (ACC): it evaluates the overall performance of an algorithm,

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

2) F1 value: it is a comprehensive measure of precision and recall rate,

$$F_1 = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

3) Matthews correlation coefficient (MCC): it evaluates the correlation between the detection ability of the algorithm and the true label:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN)}} \times \frac{1}{\sqrt{(TN + FP) \times (TN + FN)}}$$

where TP is the quantity of samples that are correctly detected as positive, TN is the quantity of samples that are correctly detected as negative, FP is the quantity of samples that were incorrectly detected as positive, and FN is the quantity of samples that were incorrectly detected as negative.

4.2 Analysis of Results

Firstly, the impact of different word vector models on vulnerability detection was analyzed, and the results were averaged (Table 3).

From Table 3, it can be found that in the comparison of single models, word2vec performed the worst in vulnerability detection, with an ACC of only 90.87%, an F1 value of 91.84%, and an MCC value of 88.12%. Glove, another static word vector model, demonstrated significantly better performance with an ACC of 92.37%, an F1 value of 93.12%, and an MCC of 89.03%. The results obtained by the ELMo model were better than the word2vec model but slightly worse than the Glove model, with an ACC of 91.62%, an F1 value of 92.16%, and a MCC of 88.94%. From this perspective, when it comes to vulnerability detection, if a single model is to be used, Glove should be chosen. Then, it can be found that the ACC of the Glove+ELMo model was 94.81%, which was 2.44% higher than the Glove model, the F1 value was 94.57%, which was 1.45% higher than the Glove model, and the MCC was 1.18%. These results verified the reliability of using dynamic and static word vector stacked embedding.

Then, The CNN+BiGRU model's effectiveness was confirmed through a comparison with the following structures: TextCNN, BiGRU, and CNN+BiLSTM. Table 4 presents the results obtained.

From Table 4, in the case of using single feature extraction, both TextCNN and BiGRU models performed

Table 2: Experimental datasets

Dataset	Not vulnerability	Vulnerability
CWE-362	375	189
CWE-476	1,227	396
CWE-754	3,684	1,359
CWE-758	995	367
Big-Vul [5]	168,752	10,547
FFMPeg+Qemu [20]	12,294	10,067

Table 3: Result analysis of different word vector models

Word vector model	ACC/%	F1/%	MCC/%
word2vec	90.87	91.84	88.12
Glove	92.37	93.12	89.03
ELMo	91.62	92.16	88.94
Glove+ELMo	94.81	94.57	90.21

Table 4: Comparison of CNN+BiGRU with other structures

Model	ACC/%	F1 value/%	MCC/%
TextCNN	87.21	88.76	83.56
BiGRU	89.34	91.25	84.33
CNN+BiLSTM	92.31	93.07	88.64
CNN+BiGRU	94.81	94.57	90.21

poorly in vulnerability detection, with both ACC below 90%. The performance of the BiGRU model was slightly better than the TextCNN model but still significantly inferior to the CNN+BiGRU model, which verified the advantages of using two models for feature extraction. Then, the ACC, F1 value, and MCC of the CNN+BiGRU model were 94.81%, 94.57%, and 90.21%, which was 2.5%, 1.5%, and 1.57% higher than the CNN+BiLSTM model. The results verified the reliability of using BiGRU to extract global features.

Finally, the proposed method was compared with the following approaches: Devign [20], IVDetect [9], FUNDED [16], and VulDeePecker [10]. The results obtained are presented in Table 5.

From Table 5, it can be found that compared with Devign and other methods, the CNN+BiGRU model showed the optimal values, with ACC, F1 value, and MCC above 90%. Among the compared methods, Devign performed poorly, and VulDeePecker performed slightly better but was still inferior to the CNN+BiGRU model. It further proved the reliability of the CNN+BiGRU model in vulnerability detection.

Table 5: Comparison between CNN +BiGRU and existing vulnerability detection methods

Model	ACC/%	F1 value/%	MCC/%
Devign	85.64	84.21	72.36
IVDetect	87.87	85.92	75.34
FUNDED	88.25	87.37	76.72
VulDeePecker	92.31	91.64	88.37
CNN+BiGRU	94.81	94.57	90.21

5 Discussion

The illegal invasion of computer viruses, i.e., propagation and replication, is based on the existence of vulnerabilities. Vulnerabilities provide attackers with an illegal invasion way. With the development of technology, the incidence and number of vulnerabilities are also increasing [1]. The management and prevention of vulnerabilities need the cooperation of technology and law. Using cutting-edge technology, coupled with strict adherence to legal regulations, can realize better legal security.

Given the current shortcomings in legal supervision, the following suggestions are put forward to prevent vulnerabilities.

1) The government should further improve the relevant provisions of the Network Security Law and promote the legalization and standardization of the behavior of vulnerability mining, detection, and disclosure. At the same time, it is also necessary to standardize the legality of the behavior of security researchers through the legal system, determine the exception system of network security research, reduce the legal risk of practitioners, and provide exemptions for the mining behavior of white hats under the supervision of the government. It should also provide a legal basis for the vulnerability mining and detection behavior of civil subjects.

2) The government should improve the vulnerability database. At present, many countries have established vulnerability databases, such as NVD, etc. A sound vulnerability database is more conducive to

managing and repairing vulnerabilities. In addition, it is also necessary to further establish and improve the vulnerability early warning mechanism to ensure security from the discovery of vulnerabilities to the repair process.

3) The authorization of vulnerability detection should be classified. According to the degree of damage caused by system destruction, the boundary of white hat mining behavior should be further clarified. For example, the mining of secret critical infrastructure should be prohibited, and the mining of other large facilities should be carried out under the premise of obtaining permission.

4) The cross-border flow of high-risk vulnerability data should be strictly restricted. Vulnerabilities should be managed as national strategic resources according to the National Security Law.

6 Conclusions

This paper studied vulnerability detection and prevention under legal supervision, improved the word vector embedding method, and designed a CNN+BiGRU detection model. Through experimental analysis, it was found that the word vector stacked embedding method obtained better vulnerability detection performance. Compared with other structures and existing methods, the CNN+BiGRU method obtained better results in various indicators, demonstrating its reliability in vulnerability detection and practical applicability.

References

[1] Y. B. Abushark, A. I. Khan, F. Alsolami, A. Almalawi, M. M. Alam, A. Agrawal, R. Kumar, R. A. Khan, "Cyber security analysis and evaluation for intrusion detection systems," *Computers, Materials, & Continua*, vol. 2022, no. 7, pp. 1765-1783, 2022.

[2] M. Alenezi, M. Zagane, Y. Javed, "Efficient deep features learning for vulnerability detection using character n-gram embedding," *Jordanian Journal of Computers and Information Technology*, vol. 7, no. 1, pp. 25-39, 2021.

[3] Y. Cheng, B. Cui, C. Chen, T. Baker, T. Qi, "Static vulnerability mining of IoT devices based on control flow graph construction and graph embedding network," *Computer Communications*, vol. 197, pp. 267-275, 2022.

[4] H. K. Dam, T. Tran, T. Pham, S. W. Ng, J. Grundy, A. Ghose, "Automatic feature learning for predicting vulnerable software components," *IEEE Transactions on Software Engineering*, vol. 47, no. 1, pp. 67-85, 2021.

[5] J. Fan, Y. Li, S. Wang, T. N. Nguyen, "A C/C++ code vulnerability dataset with code changes and CVE summaries," in *IEEE/ACM 17th International Conference on Mining Software Repositories (MSR'20)*, Seoul, Korea, Republic of, pp. 508-512, 2020.

[6] M. Hariharan, C. Sathish Kumar, A. Tanwar, K. Sundaresan, P. Ganesan, S. Ravi, R. Karthik, "Proximal instance aggregator networks for explainable security vulnerability detection," *Future Generations Computer Systems*, vol. 134, pp. 303-318, 2022.

[7] C. Hwang, H. Kim, H. Lee, T. Lee, "Effective DGA-domain detection and classification with TextCNN and additional features," *Electronics*, vol. 9, no. 7, pp. 1-18, 2020.

[8] D. Li, C. He, M. Chen, "Text sentiment analysis based on glove model and united network," *Journal of Physics Conference Series*, vol. 1748, no. 3, pp. 1-6, 2021.

[9] Y. Li, S. Wang, T. N. Nguyen, "Vulnerability detection with fine-grained interpretations," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, vol. 2021, pp. 292-303, 2021.

[10] Z. Li, D. Zou, S. Xu, X. Ou, H. Jin, S. Wang, Z. Deng, Y. Zhong, "VulDeePecker: A deep learning-based system for vulnerability detection," in *Network and Distributed System Security (NDSS) Symposium*, vol. 2018, pp. 1-15, 2018.

[11] G. Lin, S. Wen, Q. L. Han, J. Zhang, Y. Xiang, "Software vulnerability detection using deep neural networks: A survey," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1825-1848, 2020.

[12] T. Marjanov, I. Pashchenko, F. Massacci, "Machine learning for source code vulnerability detection: what works and what isn't there yet," *IEEE Security & Privacy*, vol. 20, no. 5, pp. 60-76, 2022.

[13] S. Nakamura, M. Kimura, "A calculation cost reduction method for a log-likelihood maximization in word2vec," in *25th International Conference on Automation and Computing (ICAC'19)*, Lancaster, UK, pp. 1-6, 2019.

[14] A. A. Rasheed, "Vulnerability detection towards protecting intrusion by social network analysis approach," in *5th International Conference on Trends in Electronics and Informatics (ICOEI'21)*, Tirunelveli, India, pp. 1219-1224, 2021.

[15] C. Wang, Q. Li, X. Wang, T. Ren, J. Dong, G. Guo, E. Shi, "An android application vulnerability mining method based on static and dynamic analysis," in *IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC'20)*, Chongqing, China, pp. 599-603, 2020.

[16] H. Wang, G. Ye, Z. Tang, S. H. Tan, S. Huang, D. Fang, Y. Feng, L. Bian, Z. Wang, "Combining graph-based learning with automated data collection for code vulnerability detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1943-1958, 2021.

[17] L. Yang, Y. Liao, R. Duan, T. Kang, J. Xue, "A bidirectional recursive gated dual attention unit based RUL prediction approach," *Engineering Applications of Artificial Intelligence*, vol. 120, pp. 1-22, 2023.
[18] M. Yang, J. Xu, K. Luo, Y. Zhang, "Sentiment analysis of Chinese text based on Elmo-RNN model," *Journal of Physics Conference Series*, vol. 1748, no. 2, pp. 1-6, 2021.
[19] J. Zhang, "Detection of network protection security vulnerability intrusion based on data mining," *International Journal of Network Security*, vol. 21, no. 6, pp. 979-984, 2019.
[20] Y. Zhou, S. Liu, J. Siow, X. Du, Y. Liu, "Devign: Effective vulnerability identification by learning comprehensive program semantics via graph neural networks," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems-December (NIPS'19)*, pp. 10197–10207, 2019.

Biography

Xia Li is an associate professor of Henan Institute of Economics and Trade, China. She received the Master's degree in civil and commercial law from Zhongnan University of Economics and Law, China, in 2008. Her main research interests include economic law, civil and commercial law.

An Improvement of Three-Factor Remote User Authentication Protocol Using ECC

Min-Shiang Hwang[1,2], Cheng-Ying Lin[3], and Chia-Chun Wu[4]
(Corresponding author: Chia-Chun Wu)

Department of Computer Science & Information Engineering, Asia University, Taichung, Taiwan[1]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taichung, Taiwan, R.O.C.
Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan[2]
The Ph.D. Program in Artificial Intelligence, Asia University, Taichung, Taiwan[3]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taichung, Taiwan, R.O.C.
Department of Industrial Engineering and Management, National Quemoy University, Taiwan (ROC)[4]
Kinmen 892, Taiwan, R.O.C.
Email: ccwu0918@nqu.edu.tw
(Received Apr. 28, 2023; Revised and Accepted Dec. 21, 2023; First Online Apr. 28, 2024)

Abstract

With the development of the Internet, many patients' private medical data are maliciously stolen by attackers. Therefore, Liu *et al.* proposed an improved ECC-based three-factor user authentication scheme. The main advantage of their scheme is that it provides the same or better security than other schemes using more secondary keys. However, this article will show that Liu *et al.*'s scheme is vulnerable to denial of service attacks. In addition, their protocol uses public key cryptography, which is inefficient. Finally, the method is not immune to counterfeit smart card attacks. In this paper, we also improve their protocol from the perspective of practicality and security.

Keywords: Password; Telemedicine Information System; Three-Factor Authentication; User Authentication

1 Introduction

With the rapid development of the Internet, the application of the Internet in various fields of our lives has made positive progress, such as wearable medical devices, industry, smart homes, etc. [6]. The Internet has become an essential aid in many fields. It improves work efficiency and actively drives us to change how we live, moving towards more advanced and innovative ways. However, the dependence on the network is getting higher and higher, and network security has become the focus of network development [10].

The user authentication protocols were designed to verify an authorized service in a server via an insecure channel. The user and server could authenticate each other and then use the server's services by employing a user authentication protocol [7, 8, 22].

Many user authentication protocols have been proposed [1–5, 9, 12, 14–16, 19, 21, 23–26]. A suitable user authentication protocol must meet security requirements, be simple, and be practical [17, 18, 20].

In 2018, Li *et al.* showed that Jiang *et al.* protocol [11] had some common weaknesses: clock synchronization and the lack of a password change stage [13]. Therefore, in 2018, Li *et al.* proposed an improved of Jiang *et al.* scheme. They claimed their scheme provides good security protection against relevant security attacks. However, in 2022, Liu *et al.* [18] showed that Li *et al.*'s scheme could not be against no user anonymity, no password change phase, inapplicable to IoT environments, failure to defend known session-specific temporary information attack, and no clock synchronization mechanism. They thus proposed an enhancement to Li *et al.*'s user authentication scheme. This article will show that Liu *et al.*'s user authentication scheme could not withstand the fake smart card attack and the denial of service attack.

The rest of the paper is organized as follows. First, the review and weaknesses of Liu *et al.*'s protocol are described in Sections 2 and 3. Then, in Section 4, we propose an improved remote user authentication protocol that can resist all possible attacks mentioned in Section 3. Finally, Section 5 concludes the paper.

2 Review of Liu *et al.* Three-Factor Remote User Authentication Protocol Using ECC

In 2022, Liu *et al.* proposed a secure three-factor remote user authentication protocol using elliptic curve cryptography [18]]. We will review their protocol in this section. In the Liu *et al.* protocol, there are three primary entities: the server S, gateway node GWN, and the user U_i. Furthermore, there are three phases: Registration, login and authentication, and password change phases.

2.1 Registration Phase

The procedures of the registration phase of Liu *et al.*'s protocol are listed as follows:

Step R1: The User U_i selects his/her identity (ID_i), password (PW_i), and a random number c_i.

Step R2: The User U_i extracts his/her biological feature (b_i) as the second-factor authentication.

Step R3: The User U_i computes a strength parameter, r_1 and r_2, an anonymous RID_i, and a strength password RPW_i as follows:

$$\begin{aligned} r_1 &= b_i P; \\ r_2 &= c_i r_1; \\ RPW_i &= h(PW_i||r_2); \\ RID_i &= h(ID_i||r_2). \end{aligned}$$

Here, P is a base point on the elliptic curve; c_i is a random number; and the symbol $||$ denotes a concatenation operation.

Next, The user U_i sends $\{RID_i, RPW_i, r_1\}$ to the gateway node GWN with a secure channel.

Step R4: The gateway note GWN generates a random number di and calculates $r_3, r_4, \alpha, \delta, A_i$, and B_i as follows:

$$\begin{aligned} r_3 &= d_i P; \\ \alpha &= h(r_1); \\ r_4 &= d_i r_1; \\ A_i &= h(RID_i||RPW_i||r_4); \\ B_i &= h(RPW_i||r_4) \oplus h(RID_i||K_{GWN}); \\ \delta &= r_1 \oplus r_4; \\ X &= xP_{par}. \end{aligned}$$

Here, K_{GWN} denotes a GWN secret key, and x denotes a secret key of GWN. P_{par} denotes a parameter chosen by GWN. Next, The Gateway Node stores RID_i in GWN database and stores $\{A_i, B_i, r_3, \alpha, \delta, X\}$ in SC. Next, WGN sends the smart card SC to the user U_i by a secure channel.

Step R5: The user U_i stores r_1, c_i into the smart card. The smart card SC includes $\{\alpha, \delta, A_i, B_i, r_3, X, c_i, r_1\}$.

2.2 Login and Authentication Phases

The procedures of the login and authentication phases of Liu *et al.*'s protocol are listed as follows:

Step LA1: U_i inserts the SC and extracts biological feature b_i', gets r_3 and δ from SC, and calculates

$$\begin{aligned} r_4' &= b_i' r_3 \\ &= b_i' d_i P; \\ r_1' &= b_i' P; \\ R_4\text{"} &= \delta \oplus r_1'. \end{aligned}$$

Next, SC checks whether $r_4' = R_4$" holds.

Step LA2: The user inputs ID_i and PW_i, and the smart card computes $A_i' = h(h(ID_i||cib_i'P)||h(PW_i||c_i b_i' P)||r_4')$ and checks whether $A_i' = A_i$ holds. Next, SC produces two random numbers, m_i and n_i, and computes

$$\begin{aligned} D_1 &= h(h(RPW_i||c_i||)||r_4') \oplus b_i; \\ D_2 &= m_i P_{par}; \\ D_3 &= m_i X; \\ D_4 &= RID_i \oplus D_3; \\ D_5 &= D_1 \oplus n_i \\ D_6 &= SID_i \oplus h(RID_i||n_i); \\ D_7 &= h(D_1||SID_i||D_3||n_i). \end{aligned}$$

SC sends $\{D_2, D_4, D_5, D_6, D_7\}$ to the Gateway Node.

Step LA3: GWN calculates D_3' and RID_i' as follows:

$$\begin{aligned} D_3' &= xD_2 = xmiP_{par}; \\ RID_i' &= D_4 \oplus D_3'; \\ D_1' &= h(RID_i'||K_{GWN}); \\ n_i' &= D_5 \oplus D_1'; \\ SID_i' &= D_6 \oplus h(RID_i'||n_i'); \\ D_7' &= h(D_1'||SID_i'||D_3'||n_i'). \end{aligned}$$

GWN checks whether $RID_i' = RID_i$ and $D_7' = D_7$ hold.

Step LA4: GWN generates a random number yi and calculates K'_{GWN-S} computes

$$\begin{aligned} D_{11} &= h(RID_i'||SID_i'||K'_{GWN-S}||n_i'||y_i); \\ D_{10} &= y_i \oplus n_i'; \\ D_9 &= y_i \oplus h(RID_i'||K'_{GWN-S}); \\ D_8 &= RID_i' \oplus K'_{GWN-S}. \end{aligned}$$

Next, GWN transmits $\{D_8, D_9, D_{10}, D_{11}\}$ to the server S.

Step LA5: Server S computes

$$\begin{aligned} RID_i\text{"} &= D_8 \oplus K_{GWN-S}; \\ Y_i' &= D_9 \oplus h(RID_i\text{"}||K_{GWN-S}); \\ n_i\text{"} &= y_i' \oplus D_{10}; \\ D_{11}' &= h(RID_i''||SID_i||K_{GWN-S}||n_i\text{"}||y_i). \end{aligned}$$

S verifies whether $D'_{11} = D_{11}$ holds. If it is true, S produces g_i (a random number) and computes

$$\begin{aligned} SK_i &= h(RID_i"||SID_i||n_i"||y'_i||g_i); \\ D_{12} &= g'_i \oplus K_{GWN-S}; \\ D_{13} &= h(K_{GWN-S}||SK_i||g_i). \end{aligned}$$

Next, S sends $\{D_{12}, D_{13}\}$ to GWN.

Step LA6: GWN computes

$$\begin{aligned} g'_i &= K'_{GWN-S} \oplus D_{12}; \\ SK_{GWN} &= h(RID'_i||SID'_i||n'_i||y_i||g'_i); \\ D'_{13} &= h(K'_{GWN-S}||SK_{GWN}||g'_i). \end{aligned}$$

GWN verifies whether $D'_{13} = D_{13}$ holds. If it holds, S calculates

$$\begin{aligned} D_{14} &= D'_1 \oplus y_i; \\ D_{15} &= n'_i \oplus g'_i; \\ D_{16} &= h(RID'_i||SK_{GWN}||y_i||g'_i). \end{aligned}$$

The Gateway Node transmits these parameters $\{D_{14}, D_{15}, D_{16}\}$ to the user U_i.

Step LA7: The user U_i computes

$$\begin{aligned} y_i" &= D_{14} \oplus D_1; \\ g_i" &= n_i \oplus D_{15}; \\ SK_i &= h(RID_i||SID_i||n_i||y_i"||g_i"); \\ D'_{16} &= h(RID_i||SK_i||y_i"||g_i"). \end{aligned}$$

U_i verifies whether $D'_{16} = D_{16}$ holds.

2.3 Password Change Phase

The procedures of the password change phase of Liu *et al.*'s protocol is listed as follows:

Step PC1: U_i selects a new PW_i^{new}. Next, the smart card computes

$$\begin{aligned} A_i^{new} &= h(h(ID_i||r_2)||h(PW_i^{new}||r_2)||r'_4); \\ B_i^{new} &= h(h(ID_i||r_2)||K_{GWN}) \oplus h(h(PW_i^{new}||r_2)||r'_4). \end{aligned}$$

Next, U_i updates A_i^{new} and B_i^{new} in smart card with A_i and B_i respectively.

3 Weakness of Liu *et al.* Three-Factor Remote User Authentication Protocol

This section shows that Liu *et al.*'s user authentication protocol using ECC [18] could not withstand a fake smart card attack and a difficult-to-remember random number.

3.1 The Fake Smart Card Attack

Step R6: The hacker intercepts the smart card sent from GWN and sends a fake smart card SC (FSC) to the user U_i. The fake smart card has installed malicious software in order to steal the user's U_i biological feature information.

Step LA1': U_i inserts the fake SC and extracts his/her biological feature bi. Once FSC obtains the biological feature information, it sends the biological feature information to the hacker and displays a message, "the smart card failed and should replace it", with U_i.

Since the hacker has obtained U_i biological feature information, the hacker can forge the identity of the user U_i and register with GWN and a server.

3.2 Denial of Service Attack (DoS)

In this attack, an attacker will make the GWN and server to cost a large of resources (computation). If the adversary intercepted the user's login request parameters $\{D_2, D_4, D_5, D_6, D_7\}$ in Step LA2, the attacker will attack the GWN and server as follows:

Step LA2': The adversary re-sends the intercepted message $\{D_2, D_4, D_5, D_6, D_7\}$ to the gateway node as a new login request.

Step LA3: Upon receiving $\{D_2, D_4, D_5, D_6, D_7\}$ from the adversary, the gateway node calculates D'_3, RID'_i, n'_i, SID'_i, and D'_7. Next, GWN checks whether $RID'_i = RID_i$ and $D'_7 = D_7$ hold. Since the login request parameters messages $\{D_2, D_4, D_5, D_6, D_7\}$ is a legal and valid message from the user U_i, GWN will pass the authentication in this step and continue to perform the following steps.

Step LA4: GWN generates a random number yi and K'_{GWN-S} computes D_8, D_9, D_{10}, and D_{11}. Next, GWN sends $\{D_8, D_9, D_{10}, D_{11}\}$ to the server S.

Step LA5: The server S computes $RID_i"$, Y'_i, $N_i"$, and D'_{11}. S checks whether $D'_{11} = D_{11}$ holds. Since the messages $\{D_8, D_9, D_{10}, D_{11}\}$ is a legal and valid message from the legal GWN, the server S will pass the authentication in this step. Therefore, S generates a random number g_i and calculates D_{12}, SK_i, and D_{13}. Next, S sends $\{D_{12}, D_{13}\}$ to GWN.

Step LA6: GWN computes g'_i, SK_{GWN}, and D'_{13}. GWN checks whether $D'_{13} = D_{13}$ holds. Since the messages $\{D_{12}, D_{13}\}$ is a legal and valid message from the legal server S, GWN will pass the authentication in this step. Therefore, GWN calculates D_{14}, D_{15}, and D_{16}. Next, GWN sends $\{D_{14}, D_{15}, D_{16}\}$ to the user U_i (hacker).

Although the hackers are unable to obtain the session key SK_i, the GWN and the server S will cost a large of computation and denial of other servers for other legal users.

4 The Proposed User Authentication Protocol

In the proposed protocol, there are also three phases: Registration, login and authentication, and password change. The password change phase of the proposed protocol is the same as that of Liu *et al.*'s user authentication scheme.

4.1 Registration Phase

The procedures of the registration phase are listed as follows:

Steps R1, R2, and R3: These steps are the same as that of Liu *et al.*'s user authentication protocol.

Step R4: The gateway note GWN generates di (a random number) and computes A_i, B_i, r_3, r_4, α, and δ as follows:

$$\begin{aligned} \alpha &= h(r_1); \\ r_3 &= d_i P; \\ r_4 &= d_i r_1; \\ \delta &= r_4 \oplus r_1; \\ X &= xP_{par}; \\ A_i &= h(RID_i||RPW_i||r_4); \\ B_i &= h(RPW_i||r_4) \oplus h(RID_i||K_{GWN}); \\ S &= Sig\{\alpha, \delta, A_i, B_i, r_3, X\}. \end{aligned}$$

Here, Sig denotes a signature which is produced by the GWN. Next, The Gateway Node stores RID_i in GWN database and $\{A_i, B_i, r_3, \alpha, \delta, X, S\}$ in SC. Next, WGN sends the SC to U_i by a secure channel.

Step R5: U_i verifies the signature S whether $\{\alpha, \delta, A_i, B_i, r_3, X\}$ was produced by GWN. If pass the verification, the user U_i stores r_1 and c_i in to the smart card. The smart card SC includes $\{\alpha, \delta, A_i, B_i, r_3, X, S, c_i, r_1\}$.

4.2 Login and Authentication Phases

The procedures of the login and authentication phases of the proposed protocol are listed as follows:

Step LA1: The step is the same as that of Liu *et al.*'s user authentication protocol.

Step LA2: Excepting D_3, the step is the same as that of Liu *et al.*'s protocol.

$$D_3 = m_i X \oplus T_i.$$

SC sends $\{D_2, D_4, D_5, D_6, D_7, T_i\}$ to the Gateway Node.

Step LA3: GWN checks the time stamp T_i. If T_i is the current time, GWN continues calculates D_3' and RID_i' as follows:

$$\begin{aligned} D_3' &= xD_2 \oplus T_i \\ &= xm_i P_{par} \oplus T_i; \\ RID_i' &= D_4 \oplus D_3'. \end{aligned}$$

GWN checks whether $RID_i' = RID_i$. If holds, GWN continues perform the following steps, otherwise step the login request. The other procedures are the same as that of the step of Liu *et al.*'s protocol.

Steps LA4 $\sim$ LA7: These steps are the same as that of Liu *et al.*'s user authentication protocol.

5 Cryptanalysis and Performance Analysis

The proposed three-factor remote user authentication protocol could withstand the fake smart card attack and denial of service attack.

Upon receiving $\{D_2, D_4, D_5, D_6, D_7, T_i\}$ from the adversary in Step LA2, the gateway node checks the time stamp T_i. Since the Ti is not current time, GWN thus stops the following procedures. Therefore, the proposed scheme can withstand the denial of service attack.

Table 1 compares the performance in routine procedures (without attacking situations). T_{hash} denotes a hash operation computation time; T_{mul} denotes a multiplication operation computation time; $T_{\oplus}$ denotes an exclusion OR operation computation time; T_{sig} denotes a signature operation computation time; and T_{ver} denotes a verification signature operation computation time. Although, the proposed scheme more one T_{sig} and T_{ver}, these cost is for withstanding the fake smart card attack.

Table 2 compares the performance of Liu *et al.* and the proposed schemes in a denial of service attack. In Step LA2, the gateway node checks the time stamp T_i. Since the T_i is not current time, GWN thus stops the following procedures. Therefore, the proposed don't need to perform Steps LA3 $\sim$ LA7.

Table 3 compares the performance of Liu *et al.* and the proposed schemes in one routine procedure and one denial of service attack. The total operation time of the Liu *et al.* scheme is $24T_{hash} + 6T_{mul} + 23T_{\oplus}$. However, the total operation time of the proposed scheme is only $4T_{hash} + 5T_{mul} + 7T_{\oplus}$.

6 Conclusion

In summary, we have shown the weakness of Liu *et al.*'s three-factor user authentication scheme based on ECC. Their protocol could not withstand the fake smart card

Table 1: Comparison of the performance in routine procedures (without attacking situations)

Schemes	Liu *et al.* Scheme [18]	The Proposed Scheme
Step R4	$4T_{hash} + 3T_{mul} + 2T_{\oplus}$	$4T_{hash} + 3T_{mul} + 2T_{\oplus} + 1T_{sig}$
Step R5	-	$1T_{ver}$
Step LA2	$1T_{mul}$	$1T_{mul} + 1T_{\oplus}$
Step LA3	$3T_{hash} + 1T_{mul} + 3T_{\oplus}$	$3T_{hash} + 1T_{mul} + 4T_{\oplus}$

Table 2: Comparison with the performance of Liu *et al.* and the proposed schemes in a denial of service attack

Schemes	Liu *et al.* Scheme [18]	The Proposed Scheme
Step LA3	$3T_{hash} + 1T_{mul} + 3T_{\oplus}$	0
Step LA4	$2T_{hash} + 3T_{\oplus}$	0
Step LA5	$4T_{hash} + 4T_{\oplus}$	0
Step LA6	$3T_{hash} + 3T_{\oplus}$	0
Step LA7	$2T_{hash} + 2T_{\oplus}$	0

attack and denial of service. We also propose an improved three-factor remote user authentication protocol that to withstand these weaknesses as that in Liu *et al.*'s protocol.

Acknowledgments

This research was funded by The National Science and Technology Council (Taiwan), grant number MOST 109-2221-E-468-011-MY3, NSTC 110-2221-E-507-004, and NSTC 112-2221-E-507-005-MY3.

References

[1] T. Y. Chang, W. P. Yang, M. S. Hwang, "Simple authenticated key agreement and protected password change protocol," *Computers & Mathematics with Applications*, vol. 49, pp. 703-714, 2005.

[2] T. Y. Chen, C. C. Lee, M. S. Hwang, J. K. Jan, "Towards secure and efficient user authentication protocol using smart card for multi-server environments," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1008-1032, 2013.

[3] S. F. Chiou, E. F. Cahyadi, C. Y. Yang, M. S. Hwang, , "An improved Chang-Lee's smart card-based authentication scheme," *Journal of Physics: Conference Series*, ICSP 2019, Mar. 29-31, 2019.

[4] S. F. Chiou, H. T. Pan, E. F. Cahyadi, M. S. Hwang, "Cryptanalysis of the mutual authentication and key agreement protocol with smart cards for wireless communications," *International Journal of Network Security*, vol. 21, no. 1, pp. 100-104, 2019.

[5] R. H. Dong, B. B. Ren, Q. Y. Zhang, H. Yuan, "A lightweight user authentication scheme based on fuzzy extraction technology for wireless sensor networks," *International Journal of Network Security*, vol. 23, no. 1, pp. 157-171, 2021.

[6] M. L. Dow, S. R. Dugan, "Hypothesis: A wearable device may help COVID-19 patients improve lung function," *Medical Hypotheses*, vol. 146, 2021.

[7] G. Hou, Z. Wang, "A robust and efficient remote authentication protocol from elliptic curve cryptosystem," *International Journal of Network Security*, vol. 19, no. 6, pp. 904-911, 2017.

[8] M. S. Hwang, L. H. Li, "A new remote user authentication protocol using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28-30, 2000.

[9] M. S. Hwang, J. W. Lo, S. C. Lin, "An efficient user identification scheme based on ID-based cryptosystem," *Computer Standards & Interfaces*, vol. 26, no. 6, pp. 565-569, 2004.

[10] X. Jia, D. He, N. Kumar, K. K. Raymond Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Networks*, vol. 25, no. 8, pp. 4737-4750, 2019.

[11] Q. Jiang, S. Zeadally, J. Ma, D. He, "Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376-3392, 2017.

[12] C. C. Lee, C. H. Liu, M. S. Hwang, "Guessing attacks on strong-password authentication protocol," *International Journal of Network Security*, vol. 15, no. 1, pp. 64-67, Jan. 2013.

[13] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, K. K. Raymond Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, vol. 103, pp. 194–204, 2018.

[14] C. H. Ling, C. C. Lee, C. C. Yang, M. S. Hwang, "A secure and efficient one-time password authentication scheme for WSN," *International Journal of Network Security*, vol. 19, no. 2, pp. 177-181, Mar. 2017.

[15] L. Liu, L. Hong, Z. Cao, "Analysis of one secure key agreement and key protection for mobile device user authentication," *International Journal of Network Security*, vol. 24, no. 2, pp. 238-242, 2022.

Table 3: Comparison with the performance of Liu *et al.* and the proposed schemes in one routine procedure and one denial of service attack

Schemes	Liu *et al.* Scheme [18]	The Proposed Scheme
Step R4	$4T_{hash} + 3T_{mul} + 2T_{\oplus}$	$4T_{hash} + 3T_{mul} + 2T_{\oplus}$
Step LA2	$1T_{mul}$	$1T_{mul} + 1T_{\oplus}$
Step LA3	$6T_{hash} + 2T_{mul} + 6T_{\oplus}$	$3T_{hash} + 1T_{mul} + 4T_{\oplus}$
Step LA3	$3T_{hash} + 1T_{mul} + 3T_{\oplus}$	0
Step LA4	$2T_{hash} + 3T_{\oplus}$	0
Step LA5	$4T_{hash} + 4T_{\oplus}$	0
Step LA6	$3T_{hash} + 3T_{\oplus}$	0
Step LA7	$2T_{hash} + 2T_{\oplus}$	0
Total	$24T_{hash} + 6T_{mul} + 23T_{\oplus}$	$4T_{hash} + 5T_{mul} + 7T_{\oplus}$

[16] J. Liu, X. He, H. Tang, D. Wang, B. Meng, "A novel privacy-preserving user authentication protocol for big data environment," *International Journal of Network Security*, vol. 23, no. 3, pp. 436-448, 2021.

[17] W. R. Liu, X. He, and Z. Y. Ji, "Security analysis and enhancements of a user authentication scheme," *International Journal of Network Security*, vol. 23, pp. 895-903, 2021.

[18] W. R. Liu, B. Li, Z. Y. Ji, "An improved three-factor remote user authentication protocol using elliptic curve cryptography," *International Journal of Network Security*, vol. 24, no. 3, pp. 521-532, 2022.

[19] J. Mo, Z. Hu, "Comments on a remote user authentication scheme for multi-server 5G networks," *International Journal of Network Security*, vol. 23, no. 5, pp. 878-882, 2021.

[20] J. Saadatmandan, A. Rahimi, "Digital certificate of public key for user authentication and session key establishment for secure network communications," *International Journal of Network Security*, vol. 23, no. 3, pp. 480-489, 2021.

[21] J. J. Shen, C. W. Lin, M. S. Hwang, "A modified remote user authentication protocol using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 414-416, 2003.

[22] C. S. Tsai, C. C. Lee, M. S. Hwang, "Password authentication protocols: Current status and key issues," *International Journal of Network Security*, vol. 3, pp. 101-115, 2006.

[23] C. H. Wei, M. S. Hwang, A. Y. H. Chin, "A mutual authentication protocol for RFID," *IEEE IT Professional*, vol. 13, no. 2, pp. 20-24, 2011.

[24] C. C. Yang, T. Y. Chang, M. S. Hwang, "The security of the improvement on the methods for protecting password transmission," *Informatica*, vol. 14, pp. 551-558, 2003.

[25] H. W. Yang, H. T. Pan, Y. H. Chen, M. S. Hwang, "A taxonomy of user authentication schemes for multi-server environments," *International Journal of Network Security*, vol. 22, no. 3, pp. 365-372, 2020.

[26] Q. Zhang, J. Zhang, L. Liu, J. Wang, P. Liu, "On security of privacy-preserving remote user authentication with k-times untraceability," *International Journal of Network Security*, vol. 23, no. 3, pp. 449-454, 2021.

Biography

Min-Shiang Hwang received M.S. in industrial engineering from National Tsing Hua University, Taiwan in 1988, and a Ph.D. degree in computer and information science from National Chiao Tung University, Taiwan, in 1995. He was a professor and Chairman of the Department of Management Information Systems, NCHU, during 2003-2009. He was also a visiting professor at the University of California (U.C.), Riverside, and U.C. Davis (USA) during 2009-2010. He was a distinguished professor of the Department of Management Information Systems, NCHU, during 2007-2011. He obtained 1997, 1998, 1999, 2000, and 2001 Excellent Research Awards from the National Science Council (Taiwan). Dr. Hwang was a dean of the College of Computer Science, Asia University (A.U.), Taichung, Taiwan. He is currently a chair professor with the Department of Computer Science and Information Engineering, A.U. His current research interests include information security, electronic commerce, database, data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 300+ articles on the above research fields in international journals.

Cheng-Ying Lin received the M.A. in Hochschule für Musik und Theater "Felix Mendelssohn Bartholdy" Leipzig, Germany, in 2015, and B.A. in National Hsinchu University of Education,Taiwan, in 2011. He is currently pursuing a Ph.D. degree in the Ph.D. Program in Artificial Intelligence, at Asia University, Taiwan. He was the owner and trombonist of the Brass Men ensemble. He was also served as a teacher and Wind Orchestra Conductor in many schools. His current research interests include

Artificial intelligence, Computer music.

Chia-Chun Wu received a Ph.D. degree from the Department of Computer Science and Engineering, National Chung-Hsing University, Taichung, Taiwan, in 2011. He is currently an associate professor at the Department of Industrial Engineering and Management, National Quemoy University, Kinmen County, Taiwan. His current research interests include artificial intelligence, internet of things (IoT), database security, secret image sharing, mobile applications development, and digital image techniques.

Guide for Authors
International Journal of Network Security

IJNS will be committed to the timely publication of very high-quality, peer-reviewed, original papers that advance the state-of-the art and applications of network security. Topics will include, but not be limited to, the following: Biometric Security, Communications and Networks Security, Cryptography, Database Security, Electronic Commerce Security, Multimedia Security, System Security, etc.

1. Submission Procedure

Authors are strongly encouraged to submit their papers electronically by using online manuscript submission at http://ijns.jalaxy.com.tw/.

2. General

Articles must be written in good English. Submission of an article implies that the work described has not been published previously, that it is not under consideration for publication elsewhere. It will not be published elsewhere in the same form, in English or in any other language, without the written consent of the Publisher.

2.1 Length Limitation:

All papers should be concisely written and be no longer than 30 double-spaced pages (12-point font, approximately 26 lines/page) including figures.

2.2 Title page

The title page should contain the article title, author(s) names and affiliations, address, an abstract not exceeding 100 words, and a list of three to five keywords.

2.3 Corresponding author

Clearly indicate who is willing to handle correspondence at all stages of refereeing and publication. Ensure that telephone and fax numbers (with country and area code) are provided in addition to the e-mail address and the complete postal address.

2.4 References

References should be listed alphabetically, in the same way as follows:
For a paper in a journal: M. S. Hwang, C. C. Chang, and K. F. Hwang, ``An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445--446, 2002.

For a book: Dorothy E. R. Denning, *Cryptography and Data Security*. Massachusetts: Addison-Wesley, 1982.

For a paper in a proceeding: M. S. Hwang, C. C. Lee, and Y. L. Tang, ``Two simple batch verifying multiple digital signatures," in *The Third International Conference on Information and Communication Security (ICICS2001)*, pp. 13--16, Xian, China, 2001.

In text, references should be indicated by [number].

Subscription Information

Individual subscriptions to IJNS are available at the annual rate of US$ 200.00 or NT 7,000 (Taiwan). The rate is US$1000.00 or NT 30,000 (Taiwan) for institutional subscriptions. Price includes surface postage, packing and handling charges worldwide. Please make your payment payable to "Jalaxy Technique Co., LTD." For detailed information, please refer to http://ijns.jalaxy.com.tw or Email to ijns.publishing@gmail.com.