IJNS
International Journal of Network Security

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

Vol. 25, No. 2 (March 2023)

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: 25, No: 2 (March 1, 2023)

International Journal of Network Security

Fast Face Presentation Attack Detection in Thermal Infrared Images Based on Morphological Filtering

Tai-Hung Lai, Ching-Yu Peng, and Chao-Lung Chou
(Corresponding author: Chao-Lung Chou)

Department of Computer Science and Information Engineering, Chung Cheng Institute of Technology
National Defense University
No. 75, Shiyuan Rd., Daxi Dist., Taoyuan City 335, Taiwan, ROC
Email: chaolung.chou@gmail.com

(Received Aug. 15, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)
The Special Issue on Trusted ICT Technologies on the Smart Society and Secure Multimedia Applications
Special Editor: Prof. Chin-Feng Lee (Chaoyang University of Technology)

Abstract

With the large-scale deployment and use of biometrics technology, the security threats of a biometric system are also increasing. The presentation attack (PA) is typical; an imposter spoofs legitimate users' biometrics and interferes with the system. Face recognition systems are most vulnerable to various presentation attacks, such as photo attacks, video attacks, and 3D mask attacks. Malicious attackers can easily download the victim's facial images from the Internet or social media to carry out PA simply using a printed photo on the facial recognition system. A novel approach to face presentation attack detection (PAD) is proposed because thermal infrared (TIR) images can easily capture a living person's body temperature and effectively segment facial contours using morphological filtering. The obtained face shape is binarized and performs the template matching with a predefined ellipse bitmask. The results of bit masking and the circularity of the input-filtered image are both regarded as feature vectors. Furthermore, they use the support vector machine (SVM) machine learning method for training to discriminate whether it is a spoof face. Finally, we built the "Thermal Infrared Face Attack Database" (TIFADB) database, including three photo attacks: displayed mobile phone, displayed laptop, and printer photo, for performance evaluation. The experimental results show that the proposed method is fast and efficient in detecting photo attacks, and the average accuracy can reach 95.67%.

Keywords: Biometrics; Bitmask; Morphological Filtering; Presentation Attack Detection; Thermal Infrared

1 Introduction

Biometric technology is based on various human physical and behavioral features such as the face, fingerprints, palm prints, iris, palm veins, voice, signature, and gait to identify users. Biometric authentication has the advantages of convenience, reliability, security, privacy, and efficiency; it has been widely used in numerous applications, such as access control, identity management, mobile payment, and law enforcement. However, biometrics systems are vulnerable to various presentation attacks (PA) to obstruct the biometrics system's operation [1]. The countermeasure for PA is called presentation attack detection (PAD), which can effectively distinguish whether the captured traits at the sensor are real or not. Face PAD methods have become increasingly popular among researchers in recent years because of widespread applications of face recognition. Most existing face PAD methods relied on visible light imagery based on handcrafted features, such as texture descriptors [9, 12] and motion analysis [2, 20]. However, these systems cause performance degradation due to different lighting conditions.

The essence of thermal infrared (TIR) images is that images can be obtained without being affected by different illumination conditions, even in a completely dark environment. Any object whose temperature is higher than absolute zero (-273°C) will emit infrared radiation. The spectrum's wavelength from 0.4 to 0.7 μm is the light that the eye can perceive is called visible light. The wavelength outside the visible spectrum is called infrared (IR) that cannot be seen by the human eye. The wavelength of TIR is mainly in the range of 0.75 μm∼15 μm. The TIR can be further classified into near IR (NIR, wavelength: 0.75∼1.4 μm), short-wave IR (SWIR, wavelength: 1.4∼3 μm), Middle-wave IR (MWIR, wavelength: 3∼8 μm), and

Long-wave IR (LWIR, wavelength: 8∼15 μm) [6]. Figure 1 shows the spectrum of wavelength [17]. The principle of thermal infrared imaging is to convert thermal radiation images into visible light images, allowing people to see invisible things. The thermal imager is the equipment that can detect the specific infrared band signal of the object's heat radiation, convert the signal into an image pattern that humans can visually recognize, and further calculate the temperature value. There are many thermal imaging applications, such as medical, industrial monitoring, security monitoring, military operations, and predictive maintenance of electrical equipment.

For facial recognition, compared with the visible spectrum, TIR images are less affected by posture and facial expression changes and are much easier for face detection and segmentation. The disadvantage of TIR images is the high cost of deployment. Due to the recent COVID-19 epidemic, many governments and business departments have begun to use TIR imagers to sense body temperature. The most typical application is to fix the TIR imager on the gateway and measure the passing people's facial temperature to determine whether they have a fever. With the rapid development of infrared thermal sensing equipment, we believe TIR imagers' cost will drop significantly in the near future.

Face PAD is usually located in front of the sensor before the face recognition process, and thus the detection speed is critical in practice. This paper presents the face PAD in TIR images using morphological filtering. Morphological filtering is based on mathematical morphology theoretic and operations for image analysis, such as segmentation, feature extraction, template matching, and object recognition. Morphological operations can simplify image data while retaining its basic shape features and eliminating errors. It can be applied to greyscale images and is particularly suited for binary image processing. The main advantage of morphological filtering is its low computational complexity and efficiency in representing and describing the shape of the region.

The face emits thermal radiation as an intrinsic characteristic to distinguish a living face. This implies that the face PAD can be regarded as a simple classification problem in discriminating TIR and non-TIR face images. The proposed method uses TIR images to segment the face contours and uses elliptical bit templates for template matching. The template bitmask is designed to be the same size as the input image, and the matching process can be simplified to bitwise operations to improve calculation performance. The bit masking results and the circularity shape feature are fed into the support vector machine (SVM) machine learning classifier for training in performing face PAD detection.

The contributions are summarized as follows:

- Morphological filtering is very effective for facial contour segmentation of TIR images.
- Template matching can be simplified as a bitwise operation using facial ellipse bitmask to improve computation performance.
- Thermal Infrared Face Attack Database (TIFADB) provides TIR images of photo attacks using displayed mobile phones, displayed laptops and printed photos for PAD performance evaluation.

In Section 2, a brief review of related works is provided. The morphological filtering is described in Section 3. The detailed of the proposed algorithm is presented in Section 4, and the experimental results are provided in Section 5. Section 6 gives the conclusions.

2 Related Works

The PAD on the biometric system is also called *anti-spoofing* or*liveness detection*, which means that captured traits' genuineness must be effectively distinguished. There are three main types of face spoofing attacks [10]:

1) Photo attack

   The attacker presents a photo of the legitimate user's face in front of the face recognition system sensor to deceive the authentication system. Photo attacks are the cheapest and easy means of spoof attacks. The photos of legitimate users may be secretly taken by an attacker using a digital camera or even downloaded from legitimate users' social media. Then the photo images are printed out or displayed on the screen of a smartphone or tablet. The spoof photo can then be presented in front of the face recognition system sensor to implement a spoof attack. It is still challenging to detect whether the face traits in front of the sensor comes from a real person or not.

2) Video attack

   This type of attack is also known as a replay attack. The attacker does not use static facial images but uses digital devices (such as smartphones, tablets, or laptops) to play legitimate users' facial dynamics. The video deceives the face recognition system. Because the video has a high resolution and has many physiological cues that are not in the static photo, such as head movements, facial expressions, blinking, etc., it is an advanced attack technique.

3) 3D mask attack

   This type of attack involves the attacker wearing a legitimate user's 3D face silicone mask to carry out a face recognition system's spoof attack. Since the silicone mask imitates the complete 3D structure of the legitimate user's face, using depth clues as a solution to detect 2D planar image attacks (photo attacks and video attacks) becomes ineffective in the face of this special threat.

There have been many studies on TIR's application in face detection and face recognition [6], but the application

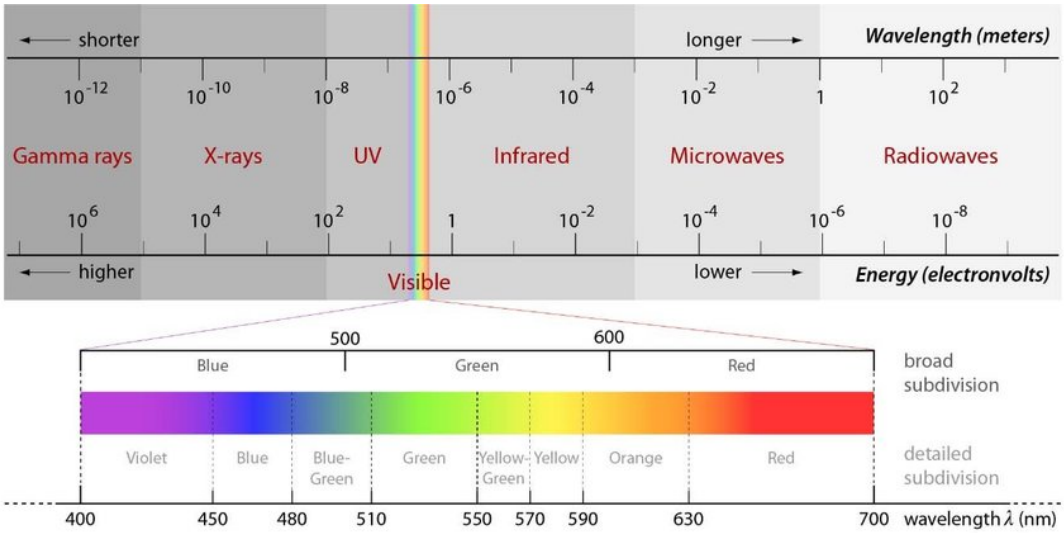


Figure 1: The radiation spectrum [17]

of TIR in PAD is less prevalent. In recent years, methods of TIR applications on fingerprint PAD [16] and 3D mask PAD [8] have been investigated.

Sun et al. proposed a face liveness detection based on cross-modality of visible and TIR image pair by canonical correlation analysis [15]. The face is divided into eight partitions, and the differences are calculated separately. Use the predefined threshold for liveness detection and use the OTCBVS database [18] for verification. The results showed that the accuracy of subjects wearing glasses was 85.1%, and the accuracy of subjects without wearing glasses was 90.8%. The main limitation of this method is that it required a visible and TIR image pair simultaneously.

Seo and Chung proposed TIR face PAD using a thermal face convolutional neural network (Thermal Face-CNN) with external knowledge [13]. The external knowledge is that the real face temperature is average near 36 to 37 degrees. The input data will be filtered according to the external knowledge first and then calculated using CNN with two convolutional layers and two pooling layers and one hidden layer. The experiment uses a self-built database in 844 color thermal images, including 338 real faces and 506 non-face objects. The results showed the best accuracy of 83.67% by using CNN.

Singh and Arora proposed a computer-aided face liveness detection with the facial thermography algorithm (CAFLD) for TIR face PAD [14]. The face is divided into seven regions (forehead, right eye, left eye, right cheek, left cheek, nose, and mouth). The average temperature distribution of the collected database of 100 subjects is calculated as the threshold. The accuracy of the seven areas of the face is between 91% and 98%, with an average of 96.57%. This method's limitation is that the sample is small, and the results are based on different face areas rather than the number of objects. Hence, the actual performance may need to be verified.

The above methods used common public thermal image face DB or self-built, but none of the testing datasets includes actual thermal IR attack images. As our best knowledge, the proposed method is the first to collect three real photo attack scenarios in thermal IR images for performance evaluation. This will make our proposed method performance closer to the actual results.

The facial PAD design needs to be considered in actual needs, including cost, accuracy, and convenience. Another critical factor is the detection speed, which must avoid taking up too much time and delaying the overall face recognition performance.

3 Morphological Filtering

Morphology is a mathematical tool used to extract image components related to the geometric structure of an object. It can be done by smoothing the object's outline, filling small holes, eliminating small protrusions, etc. The two principal morphological operations are dilation and erosion [7]. Dilation fills in small holes and connecting disjoint objects to expand objects while erosion shrinks objects by eroding their boundaries. These operations can be customized for the application by choosing the correct structural element, determining how the object will expand or corrode.

The dilation of a binary image A by a structural element B, denoted as $A \oplus B$, is defined as Equation (1).

$$A \oplus B = \{x | (B)_x \cap A \neq \phi\} \tag{1}$$

where x is the displacement of the original center of B.

The erosion of a binary image A by a structural element B, denoted as $A \ominus B$, is defined as Equation (2).

$$A \ominus B = \{x | B_x \subseteq A\} \tag{2}$$

where x is the displacement of the original center of B.

These two basic operations, dilation and erosion, can be combined into more complex sequences. The most typical morphological filtering is *opening* and *closing*. The opening is composed of an erosion followed by dilation and can be used to eliminate all pixels in too small areas to accommodate structural elements, is defined as Equation (3). The closing consists of a dilation followed by erosion and can be used to fill holes and small gaps, is defined as Equation (4).

$$A \circ B = (A \ominus B) \oplus B. \quad (3)$$

$$A \bullet B = (A \oplus B) \ominus B. \quad (4)$$

The Morphological Filtering process is performed by laying the *structural element* on the image and sliding it across the image in a manner similar to convolution.

A bitwise operation is based on Boolean logic and is a fast and simple action that operates directly on a bit string at the level of its individual bits. Typically, bitwise operations are faster than higher-level arithmetic operations, such as division, multiplication, and addition. The most common bitwise operators are bitwise AND, bitwise OR, and bitwise XOR. Bitwise AND is to extract a subset of the bits in the value; bitwise OR is to set a subset of the bits in the value; bitwise XOR is to toggle a subset of the bits in the value.

The bitwise AND operation is usually considered a bitmask because it defines a mask of bits to be retained and cleared. For example, applying the mask (00001111) to the value means that to clear the first (higher) 4 bits and keep the last (lower) 4 bits; conversely, applying the mask (11110000) to the value means that to keep the first 4 bits and clear the last 4 bits, as shown in Figure 2(a) and Figure 2(b), respectively.

4 The Proposed Method

The proposed method uses morphological filtering to calculate the face contour's geometric shape after the thermal infrared image's binarization. The binary bitmask of the same size as the original image is established using an ellipse's characteristics. The results of the bitwise AND operation on the input image and bitmask is regarded as a feature value and fed into machine learning training to determine whether it is a fake face. The flowchart of the proposed method is shown in Figure 3, and the detailed steps are described as follows.

Step 1. Data input.

The input image $I(x, y)$ is captured by using a thermal infrared camera. If the input image is a color image, then converted it into grayscale space RGB values to grayscale values by using Equation (5). If the input image is grayscale, the original image is directly used.

$$I(x, y) = 0.299 \times R + 0.587 \times G + 0.144 \times B. \quad (5)$$

Step 2. Image normalization.

The input thermal image should be normalized to the same size as the template to increase template matching processing speed. In this way, the matching score can be calculated by bitmask operation and reduce the computation cost significantly. Suppose $I(x_{dst}, y_{dst})$ denotes the normalized image from the original input image $I(x_{src}, y_{src})$ by using Equation (6).

$$\begin{aligned} x_{dst} &= x_{src} \times \left(\frac{\text{Width}_{dst}}{\text{Width}_{src}} \right), \\ y_{dst} &= y_{src} \times \left(\frac{\text{Height}_{dst}}{\text{Height}_{src}} \right). \end{aligned} \quad (6)$$

Step 3. Image binarization.

Image binarization is the process of obtaining grayscale images and converting them into black and white. Here, we use Otsu's method [11] to determine a threshold value and divide the normalized grayscale image into a binary image. The binary image consists of two parts, the object (face shape) and the background.

Step 4. Image opening.

The proposed method uses Equation (3) to perform the opening operation to remove possible noise inside the face shape after image binarization. Figure 4 shows the result of the normalized input image after binarization and image opening operation.

Step 5. Template mask construction.

The shape of the human face can be considered to be an ellipse. To facilitate feature extraction, we create an ellipse shape similar to the human face shape as a matching template. Suppose that there are two points F_1 and F_2 on the plane, a is the length of the semi-major axis, b is the length of the semi-minor axis, and$\overline{F_1F_2} < 2a$, then the ellipse formed by all P points satisfying $\overline{PF_1} + \overline{PF_2} = 2a$ as shown in Figure 5. F_1 and F_2 are called the focal points of the ellipse, and the midpoint of the connecting segment between the two focal points is called the center point O. We use Equation (7) to construct the template mask $M(H, W)$ as shown in Figure 6.

$$\frac{\left(x - \frac{H}{2}\right)^2}{a^2} + \frac{\left(y - \frac{W}{2}\right)^2}{b^2} = 1. \quad (7)$$

Step 6. Bit masking.

The normalized binary image $I(H, W)$ and the constructed template mask $M(H, W)$ with the same size will perform the bitwise AND operation. f_R denotes the ratio of bit masking results to the image size that also regards as the ratio between white pixels (bit is "1") and total pixels within the image $I(x, y)$ as Equation (8) shows.

$$f_R = \frac{\sum \text{AND}(I(x, y), M(x, y))}{H \times W}, \ 1 \leq x \leq H, 1 \leq y \leq W \quad (8)$$

	Value:	0	1	1	0	1	1	1	0
AND	Mask:	0	0	0	0	1	1	1	1
	Result:	0	0	0	0	1	1	1	0

(a) The mask to keep lower 4 bits.

	Value:	0	1	1	0	1	1	1	0
AND	Mask:	1	1	1	1	0	0	0	0
	Result:	0	1	1	0	0	0	0	0

(b) The mask to keep higher 4 bits.

Figure 2: Example of bitwise AND operation

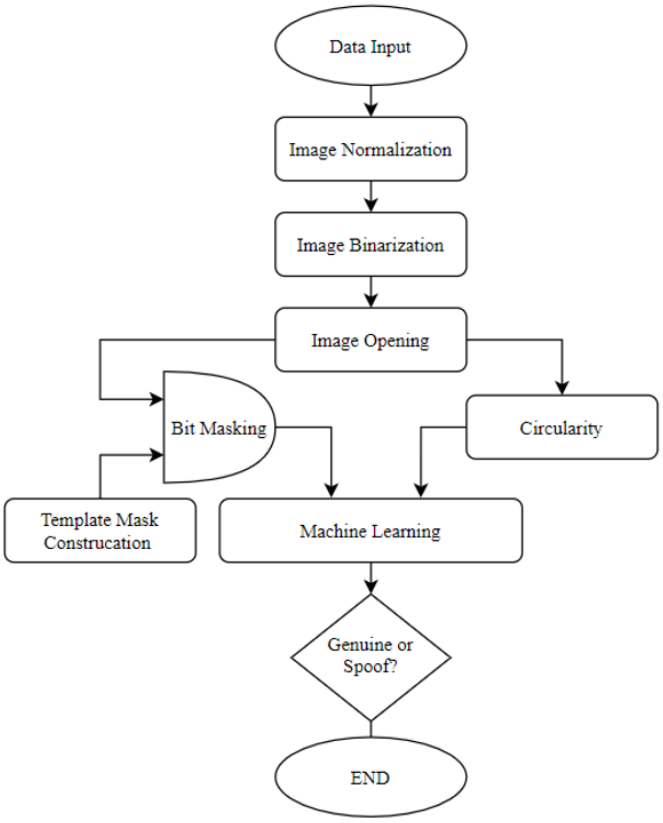
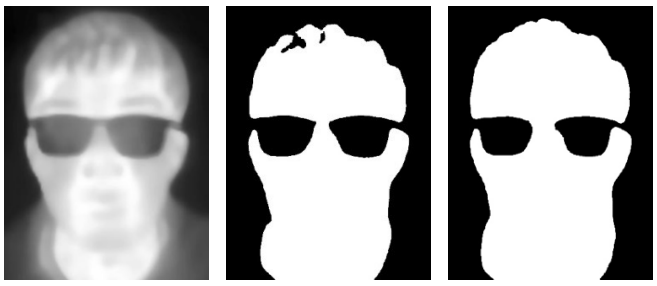


Figure 3: Flowchart of the proposed method



(a) (b) (c)

Figure 4: Example of input image preprocessing. (a) Normalized. (b) Binarization. (c) Opening

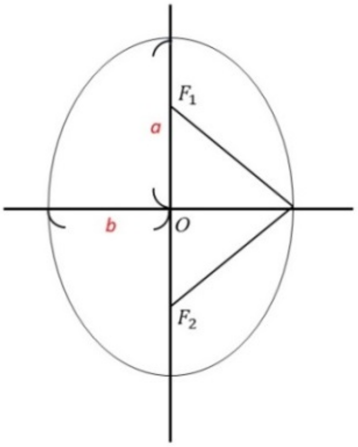


Figure 5: Example of an ellipse. F_1 and F_2 are the ellipse's focal points, O is the center point, and a is the length of the semi-major axis, b is the length of the semi-minor axis.

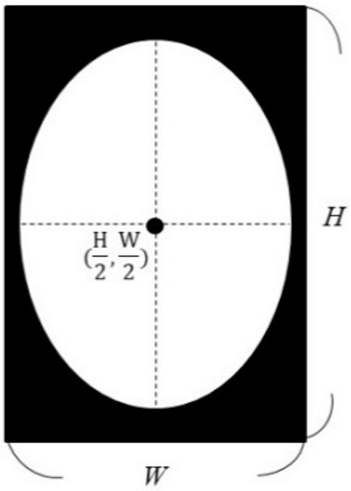


Figure 6: Example of the bitmask with width W and height H. The center point is denoted as $(H/2, W/2)$

Step 7. Circularity.

Circularity is the measure of how closely the shape of an object approaches that of a mathematically perfect circle, as Equation (9) shows.

$$f_C = \frac{4 \times \pi \times A}{P^2} = \frac{4\pi \times R^2\pi}{(2\pi R)^2},\ C \in (0,1], \qquad (9)$$

where A is the circular area, R is the circle radius, p is the circumference, and π is the circular constant.

The SVM is used for training and testing in the stage [3]. SVM is a supervised learning model for classification and regression analysis. We use SVM mainly because it can solve nonlinear classification problems and has good performance in small samples. Here, the genuine images are labeled as "one", and the spoof images are labeled as "zero".

After the input image has gone through the previous steps, it will be classified into "genuine" or "spoof" according to the model trained by machine learning.

Figure 7 shows an example of the results of the genuine image and the spoof image (displayed laptop) are respectively subjected to the template mask. The bit masking is conducted by bitwise AND operation, and the feature vectors obtained by the genuine image are $f_R = 0.86$ and $f_C = 0.78$, and that are $f_R = 0.27$ and $f_C = 0.60$ by the spoof image.

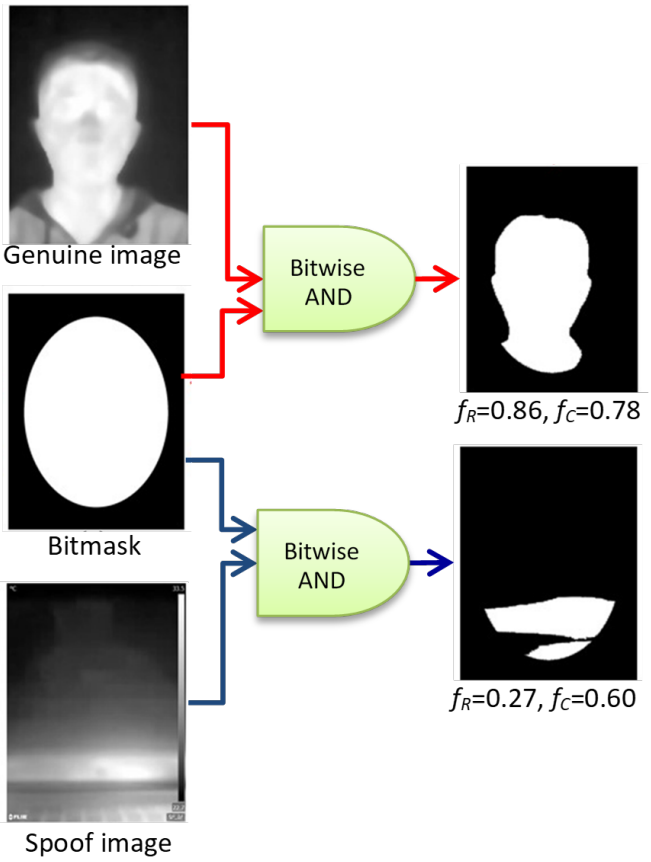


Figure 7: Example of the genuine image and the spoof image (displayed laptop) subjected to the template mask by the bitwise operation. The obtained features are $f_R = 0.86$ and $f_C = 0.78$ for the genuine image, and that is $f_R = 0.27$ and $f_C = 0.60$ for the spoof image.

5 Experiments and Discussion

5.1 Dataset Collection and Experimental Setup

The common public databases of facial thermal images are the OTCBVS [18] and Carl databases [5]. However, the existing databases contain only facial thermal images, and there are no spoofing scenes to simulate attack scenarios. To ensure effective evaluation of the proposed method's performance, we built a thermal image database named "Thermal Infrared Face Attack Database" (TIFADB).

The capturing device used FLIR ONE, which can be connected to Android or iOS smartphones and tablet devices. The spectral range is 8-14 μm, a long-wave infrared image (LWIR), and the detection temperature difference can be as small as 0.1°C. TIFADB collected a total of 1,970 images, including 1,379 real images and 591 deceptive images. The size of each image in TIFADB is the same as 1,080×1,440 pixels. There are 197 objects, of which 145 are male, and 52 are female. Each object has been collected ten images, seven of which represent facial images toward different directions: (a) front, (b) up, (c) down, (d) right, (e) right front, (f) left front, and (g) left, as shown in Figure 8. The remaining are three images representing different photo attacks that (a) displayed photo using mobile phones, (b) displayed photo using laptops, and (c) printed photos using a laser printer, as shown in Figure 9.

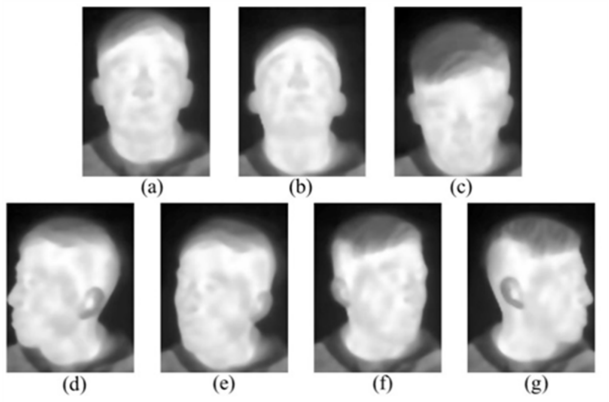


Figure 8: Example of genuine images in TIFADB. (a) front, (b) up, (c) down, (d) right, (e) right front, (f) left front, and (g) left.

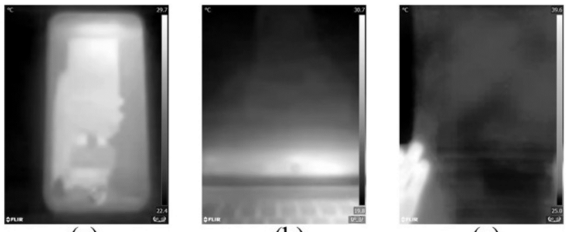


Figure 9: Example of spoofing images in TIFADB. (a) Displayed photo using mobile phones, (b) Displayed photo using laptops, and (c) Printed photo using a laser printer.

The experiments use a laptop with Intel CPU Core i7-8550U and 16 GB RAM, and the software are OpenCV 3.4.2.16 and Python 3.7.3 with packages including pandas 0.25.1, scikit-learn 0.21.3, matplotlib 3.1.1, dlib 19.9.0, and numpy 1.16.4. The threshold T used in the Otsu's method is set as 138.

To evaluate the performance of the proposed method, we uses accuracy (ACC), false rejection rate (FRR), false acceptance rate (FAR), and half total error rate (HTER) [10] as the evaluation criteria defined as Equation (10) to Equation (13). The higher the ACC, the lower the FAR, FRR and HTER, denoting better performance. Note that TP (True Positive) represented an object is positive and also be correctly recognized, TN (True Negative) represented an object is negative as also correctly recognized, FP (False Positive) represented an object is negative but wrongly recognized as positive, FN (False Negative) represented an object is positive but wrongly recognized as negative.

$$\text{ACC} = \frac{TP+TN}{TP+TN+FP+FN} \tag{10}$$

$$\text{FAR} = \frac{FP}{FP+TN} \tag{11}$$

$$\text{FRR} = \frac{FN}{TP+FN} \tag{12}$$

$$\text{HTER} = \frac{FRR+FAR}{2} \tag{13}$$

There is a total of three photo attack scenarios in the experiments. The first one uses a mobile phone to display the face, the second one uses a laptop to display the face, and the last one prints the face with a laser printer. The experiments randomly select 60% of genuine and spoof images in the TIFADB as the training data and use the remaining 40% data for testing. Since the testing result is either true or false, the SVM classifier used in the experiments adopts one-class with a linear kernel.

5.2 Experimental Results and Performance Comparison

The experimental results show in Table 1. It can be seen that the method proposed method has the best effect against photo attack by the printed photos with an accuracy of 97%. The accuracy of detecting photo attacks by displayed using mobile phones and laptops is 94% and 96%, respectively, and the average accuracy is 95.67%. In terms of the HTER, the best case is about 2% against the printed photos. The HTER of detecting photo attacks by using displayed mobile phones and laptops is 16% and 5%, respectively, and the average HTER is about 7.67%. The performance of detecting the photo attacks by using displayed mobile phones is significantly lower than that of the others. The main reason is that the shape of the mobile phone's thermal image after morphological filtering is closer to the ellipse bitmask used in this paper. Therefore, it is easier to cause misjudgments in classification. Overall, the proposed method can still effectively detect three different types of photo attacks. Because the face PAD using thermal infrared images can be regarded as a simple classification problem to distinguish TIR and non-TIR facial images, similar to determine whether the face detection of thermal images is successful or not. The experiment is compared with typical face detection methods: LBP (local binary pattern) [9], HOG (oriented gradient histogram) [4] and Haar cascade [19]. The results are shown in Table 2. From Table 2, the proposed method significantly outperforms LBP, HOG, and Haar in terms of accuracy and HTER in all three photo attack scenarios.

Table 1: Results parameters ACC, FAR, FRR, and HTER

Source	ACC	FAR	FRR	HTER
Mobile phones	0.94	0.27	0.04	0.16
Laptops	0.96	0.05	0.04	0.05
Printed photos	0.97	0	0.04	0.02
Average	0.95	0.10	0.04	0.07

The experiment is further compared with LBP, HOG and Haar in terms of computation time. Table 3 shows that the average computation time of the proposed method is 0.11 seconds per frame, which is better than LBP's 0.13 seconds per frame about 18%, better than HOG's 0.86 seconds per frame about 780%, and better than Haar's 0.86 seconds per frame 330%. In summary, we have the following findings:

1) Good accuracy: The proposed method has an average accuracy of 95.67% and an average HTER of 7.67%, with excellent performance and outperforms face detection methods such as LBP, Haar, and HOG.

2) Fast calculation: The proposed method performs well in calculation cost, and the processing time of a single image frame is 0.11 second, which is suitable for real-time applications, and also outperforms LBP, HOG, and Haar.

3) The ellipse mask is susceptible to interference with similar shapes: Among the three photo attacks scenarios, since the shape of the mobile phone shape

Table 2: Comparisons of performance results in using LBP, HOG, Haar and the proposed method

Source	Mobile phones				Laptops				Printed photos				Overall			
Methods	ACC	FAR	FRR	HTER	ACC	FAR	FRR	HTER	ACC	FAR	FRR	HTER	ACC	FAR	FRR	HTER
LBP	0.67	0.52	0.30	0.41	0.71	0.21	0.30	0.26	0.68	0.44	0.30	0.37	0.68	0.39	0.30	0.34
HOG	0.38	0.27	0.25	0.39	0.39	0	0.70	0.35	0.33	0.43	0.70	0.56	0.36	0.23	0.55	0.43
Haar	0.52	0	0.54	0.42	0.52	0	0.54	0.27	0.52	0.01	0.54	0.28	0.52	0	0.54	0.32
Proposed Method	0.94	0.27	0.04	0.16	0.96	0.05	0.04	0.05	0.97	0	0.04	0.02	0.95	0.10	0.04	0.07

Table 3: Comparisons of execution time between LBP, HOG, Haar and the proposed method (in seconds)

Methods	Mobile phones	Laptops	Printed photos	Overall
LBP	0.13	0.13	0.14	0.13
HOG	1.01	1.01	0.58	0.86
Haar	0.34	0.34	0.35	0.34
Proposed Method	0.11	0.11	0.11	0.11

may be very close to an ellipse, the performance of using mobile phones will be significantly reduced compared with that using laptops and printed photo. Therefore, the proposed method will be interfered with by similar elliptical heating objects, which is a limitation to be considered.

6 Conclusions

This research proposed a face PAD technology in thermal IR images using morphological filtering. The input facial image through binarization and opening operation, then the results of bit masking with ellipse template mask and circularity are fed into SVM classifier to determine whether it is a real face. The experimental results show that the overall performance is the high average accuracy of 95.67% and the low average HTER of 7.67%. It takes only 0.11 seconds to conduct the PAD, which implies that the proposed method's main features are fast and simple to implement. The experimental results also found that this method's elliptical bitmasks may be easy to misclassify with similar object shapes. We will leave this future research issue to enhance the robustness of the proposed method. Also, TIFADB will continue to be expanded, adding various environmental scenarios, such as wearing glasses, hats, masks, indoors, outdoors, etc., simulating more realistic s scenarios for face PAD evaluation.

Acknowledgments

This work was supported by the National Science Council of Taiwan under grant NSC 111-2221-E-606-013. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] Z. Akhtar, C. Micheloni, and G. L. Foresti, "Biometric liveness detection: Challenges and research opportunities," *IEEE Security & Privacy*, vol. 13, no. 5, pp. 63–72, 2015.

[2] A. Anjos, M. M. Chakka, and S. Marcel, "Motion-based counter-measures to photo attacks in face recognition," *IET Biometrics*, vol. 3, no. 3, pp. 147–158, 2014.

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

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, IEEE, pp. 886–893, 2005.

[5] V. Espinosa-Duró, M. Faundez-Zanuy, and J. Mekyska, "A new face database simultaneously acquired in visible, near-infrared and thermal spectrums," *Cognitive Computation*, vol. 5, no. 1, pp. 119–135, 2013.

[6] R. S. Ghiass, O. Arandjelović, A. Bendada, and X. Maldague, "Infrared face recognition: A comprehensive review of methodologies and databases," *Pattern Recognition*, vol. 47, no. 9, pp. 2807–2824, 2014.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing, 4ed.*, New York: Pearson, 2018.

[8] M. Kowalski, "A study on presentation attack detection in thermal infrared," *Sensors*, vol. 20, no. 14, p. 3988, 2020.

[9] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using texture and local shape analysis," *IET biometrics*, vol. 1, no. 1, pp. 3–10, 2012.

[10] S. Marcel, M. S. Nixon, and S. Z. Li, *Handbook of biometric anti-spoofing, 2ed.* Springer, 2019.

[11] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[12] R. Raghavendra, K. B. Raja, and C. Busch, "Presentation attack detection for face recognition using light field camera," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1060–1075, 2015.

[13] J. Seo and I.-J. Chung, "Face liveness detection using thermal face-cnn with external knowledge," *Symmetry*, vol. 11, no. 3, p. 360, 2019.

[14] M. Singh and A. S. Arora, "Computer aided face liveness detection with facial thermography," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2465–2476, 2020.

[15] L. Sun, W. Huang, and M. Wu, "Tir/vis correlation for liveness detection in face recognition," in *International Conference on Computer Analysis of Images and Patterns*, Springer, pp. 114–121, 2011.

[16] R. Tolosana, M. Gomez-Barrero, C. Busch, and J. Ortega-Garcia, "Biometric presentation attack detection: Beyond the visible spectrum," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1261–1275, 2019.

[17] G. Verhoeven, "The reflection of two fields–electromagnetic radiation and its role in (aerial) imaging," *AARGnews*, vol. 55, no. 55, pp. 10–18, 2017.

[18] Vcipl-okstate, *OTCBVS Benchmark Dataset Collection*, 11 Sep. 2022. (https://vcipl-okstate.org/pbvs/bench/)

[19] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition (CVPR'01)*, vol. 1. IEEE, 2001.

[20] T. Wang, J. Yang, Z. Lei, S. Liao, and S. Z. Li, "Face liveness detection using 3d structure recovered from a single camera," in *International Conference on Biometrics (ICB'13)*, IEEE, pp. 1–6, 2013.

Biography

Tai-Hung Lai received the B.S. degree in computer science from the Chung Cheng Institute of Technology, in 1999, and the M.S. degree in computer science and information engineering and the Ph.D. degree in electrical and electronics engineering from the Chung Cheng Institute of Technology, National Defense University, Taiwan, in 2005 and 2012, respectively. Since 2019, he has been an Assistant Professor with the Computer Science and Information Engineering Department, Chung Cheng Institute of Technology, National Defense University, Taiwan. His research interests include data hiding, network attack and defense, image processing, and computer vision.

Ching-Yu Peng received bachelor degree from Management College National Defense University, Taiwan, in 2012, and a master's degree in Computer Science and Information Engineering from Chung Cheng Institute of Technology in 2020. During 2014–2018, she worked in Information, Communications and Electronic Force Command as an Information Security Research Assistant. She published research at the 28th Conference on National Defense Science and Technology in 2019. Her research interests include biometrics, presentation attacks detection, thermal image, and machine learning.

Chao-Lung Chou received the Ph.D. degree in electrical and electronics engineering from the Chung Cheng Institute of Technology, National Defense University, Taiwan, in 2012. Since 2015, he has been an Associate Professor with the Computer Science and Information Engineering Department, Chung Cheng Institute of Technology, National Defense University. His research interests include information security, image processing, machine learning, and biometrics.

Chaotic Maps-based Privacy-Preserved Three-Factor Authentication Scheme for Telemedicine Systems

Tzu-Wei Lin[1] and Chien-Lung Hsu[2,3,4]

(Corresponding author: Tzu-Wei Lin)

Information Security Office, Office of Information Technology, Feng Chia University[1]
No. 100, Wenhwa Rd., Seatwen Dist., Taichung City, Taiwan (R.O.C.)
Email: tweilin@fcu.edu.tw
Department of Information Management, Chang Gung University[2]
Graduate Institute of Business and Management, Chang Gung University[3]
Healthy Aging Research Center, Chang Gung University[4]
No.259, Wenhua 1st Rd., Guishan Dist., Taoyuan City, Taiwan (R.O.C.)

(Received Aug. 15, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)
The Special Issue on Trusted ICT Technologies on the Smart Society and Secure Multimedia Applications
Special Editor: Prof. Chin-Feng Lee (Chaoyang University of Technology)

Abstract

5G has notable features which can provide reliable and speedy connectivity to the future Internet of Things and improve users' perceived quality of service, and IoT with a 5G environment can solve challenges of smart medical healthcare solutions. Medical privacy is essential because the economy and credibility of medical institutions lose if privacy cannot be protected. Moreover, patients will become victims of such incidents. We proposed a chaotic maps-based privacy-preserved three-factor authentication (CMPP-3FA) scheme for telemedicine systems which can provide privacy preservation while online consulting between patient and medical professional teams.

Keywords: Chaotic Maps; Privacy Preservation; Telemedicine Systems; Three-Factor Authentication

1 Introduction

5G (the fifth generation) networks is being deployed on the earth which provides high-speed network, big capacity, and scalability [2, 8]. The vision of next generation 5G wireless communications lies in providing very high data rates, extremely low latency, manifold increase in base station capacity, and significant improvement in users' perceived quality of service [12, 17]. 5G can significantly increase capacity and speed to provide reliable and speedy connectivity to the future Internet of Things (IoT) [10]. Nowadays, medical healthcare systems face many challenges. IoT with 5G environment provides solutions of network layer, including enhancing quality of service, router and jamming control, resource optimization, etc., to solve challenges of smart medical healthcare solutions [2, 12, 15]. Medical privacy is important because that economic and credibility to medical institutions losses if privacy cannot be protected [12, 15]. Moreover, patients will become victims in such incidents. Increasingly, people interact with health-care providers, using digital media technologies [19]. Accompanying the acceleration of medical data collection are rapid advancements in algorithmic computing capacities to aggregate, analyze, and draw sensitive inferences about individuals from their health data [19, 20].

2 Related Works

Telemedicine systems is a technology of electronic message and telecommunication related to healthcare [1]. Patient sends important, sensitive, and private healthcare related information to healthcare services through public networks when using telemedicine technology [1]. Medical professionals can know users' health condition if they are able to view the information immediately including online consulting [1]. Data transmission security will be discussed, such as eavesdropping, man-in-the-middle (MITM) attack, data tempering attack, message modification attack, data interception attack, etc. [27]. Telemedicine system is implemented in public networks, so privacy protection is one of notable security issues [12, 15]. Mishra *et al.* and Renuka *et al.* utilized biometric feature to design authentication schemes for telemedicine systems [18, 21]. Sureshkumar *et al.* de-

signed authentication and key agreement for telemedicine system [22]. In summary, three keys to the question must be solved for assuring telemedicine environments. First, storing amount of image should be highly efficient. Second, transmitting sensitive image should satisfy confidence, integrity, and accessibility. Finally, encryption progress should be efficient especially for end point.

Passwords have been chosen to use for identification authentication in network and computer applications [16]. Passwords are suggested being as strong as possible, but, generally, users choose weak password, which is short or easy to remember and be guessed easily [15]. Biometric authentication over cloud and network applications demands a wide range of solutions against increasing cybercrimes and digital identity thefts and improves weakness of password-based authentication, such as password-missing, weak password, etc. [7]. Moreover, multi-factor authentication has been proposed and utilized for raising security instead of using single factor authentication. Although two-factor authentication, which utilizes password and smart card in lots of cases, has been proposed and applied for years, two-factor authentication has been proven that is still not secure enough to protect privacy of users [5]. Three-factor authentication, which integrates smart card, password and biometric features, has been proven more secure than two-factor authentication and applied widely [5]. Jiang*et al.* proposed a privacy preserving three-factor authentication scheme for e-Health clouds, which may be attacked by replay attack because of using the same key in authentication phase [6]. Zhang*et al.* proposed a privacy protection for e-Health systems by means of dynamic authentication and three-factor key agreement using biometrics dynamic authentication [26]. However, adversary can guess a legitimate user's biometric feature if random number is long enough. In addition, server stores dynamic verification table in Zhang*et al.* proposed scheme [26], and adversary can steal dynamic verification table.

Chaotic system has properties, a sensitive dependence on initial conditions, pseudo-randomness, and ergodicity, which can correspond to cryptosystem's properties calles confusion and diffusion [9, 13–15, 23, 24]. First, result is unpredictable if small changes in initial values happen=. Second, chaotic system is a complex oscillation. Third, chaotic system has qualitative change of character of solutions. Mathematical definitions of Chebyshev chaotic maps can be refereed to these previous studies [9, 11, 13–15, 23, 24].

3 Proposed Scheme

Proposed scheme consists of two rules: user U_i and server S_j. Proposed scheme include system initialization phase, registration phase, login and the first-time authentication phase, the γ time authentication phase, offline password change phase, and revocation phase. Table 1 summarizes used notations of proposed scheme, respectively. Note that smartcard in proposed scheme supports public key infrastructure and X.509 certificate. The details of each phase will be described as below.

Table 1: Notations

Notations	Definitions
ID_i	Identity of user U_i.
SID_j	Identity of server S_j.
$E_k(.)$, $D_k(.)$	A symmetric encryption/decryption algorithm with secret key k.
x_{S_j}	Secret value of server S_j.
$h_k(.)$	Collision-resistance secure one-way keyed chaotic hash function.
PW_i	Password of user U_i.
Bio_i	Biometric template of user U_i.
$\bigoplus$	Exclusive OR (XOR) operation.
Bio_i	Biometric template of user U_i.
$H(.)$	Collision-resistant one-way hash functions.
MAC_A	Message authentication code algorithm of A.
b_{ij}	Number of the authentication time.
Bio_i	Biometric template of user U_i.

3.1 Registration Phase

User U_i needs to register to server S_j as a legitimate user via a secure channel. User U_i first enters (ID_i, PW_i, Bio_i), then uses smartcard to choose a random number $y_i \in Z_p^*$ and compute $\alpha_i = T_{y_i}(x) \bmod p$ and $A_i = h_{\alpha_i}(PW_i, Bio_i) \bigoplus h_{\alpha_i}(y_i, SID_j)$. Then, smartcard stores y_i and sends (ID_i, A_i) to server S_j. After receiving message from user U_i, server S_j computes $\beta_j = T_{x_{S_i}}(x) \bmod p$, $u_i = h_{\beta_j}(ID_i)$, $u_j = h_{\beta_j}(SID_j)$, $B_i = u_i \bigoplus A_i$, and $B_j = u_j \bigoplus A_i$. After that, server S_j returns (B_i, B_j) to user U_i, and user U_i stores (B_i, B_j) in USB or smartcard.

3.2 Login and the First Time Authentication Phase

To complete mutual authentication and session key confirmation and obtain services, user U_i and server S_j perform following steps (see Figure 1). First, user U_i uses smartcard to generate ($h^{(1)}(v_{ij})$, $h^{(2)}(v_{ij})$, ..., $h^{(b+1)}(v_{ij})$) and $P_1 = h^{(b+1)}(v_{ij})$. After entering (ID_i, PW_i, Bio_i), smartcard checks PW_i and Bio_i. If PW_i and Bio_i are correct, smartcard utilizes (y_i, x) to compute A_i, retrieves (B_i, B_j) to recover u_i, and computes $K_i = A_i \bigoplus h_{\alpha_i}(y_i)$ and $R_i = B_l \bigoplus h_{\alpha_i}(y_i)$. Then, smartcard chooses integer $\rho \in Z_p^*$ and a big prime N_i to compute $\mu_i = T_{y_i}(\rho_i) \bmod N_i$, $b_i = E_{u_i}(N_i \text{——} \rho_i \text{——} P_1)$, and $C_i =$

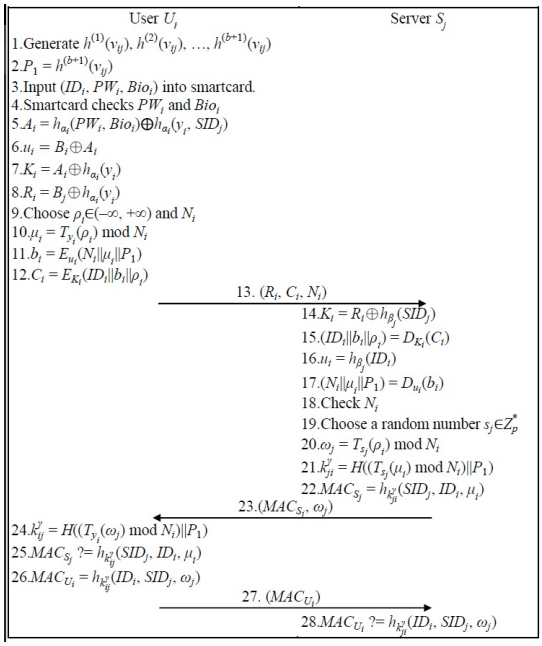


Figure 1: Login and the first-time authentication phase

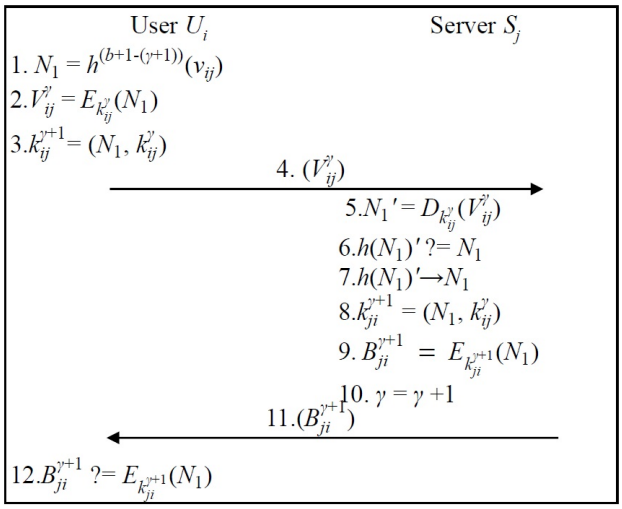


Figure 2: The γ time authentication phase

$E_{K_i}(ID_i——b_i——\rho_i)$ and sends (R_i, C_i, N_i) to server S_j. After receiving (R_i, C_i, N_i), server S_j computes $K_i = R_i \bigoplus h_{\beta_j}(SID_j)$, $(ID_i——b_i——\rho_i) = D_{K_i}(C_i)$, $u_i = h_{\beta_j}(ID_i)$, and $(N_i——\rho_i——P_1) = D_{u_i}(b_i)$. If server S_j can decrypt b_i successfully , server S_j authenticates user U_i successfully. For establishing a shared session key, server S_j chooses a random number s_j, utilizes (ρ_i, N_i, μ_i) compute $\omega_j = T_{s_j}(\rho_i) \bmod N_i$, $k_{ij}^{\gamma} = H((T_{s_j}(\mu_i) \bmod N_i)——P_1)$, and $MAC_{S_j} = h_{k_{ij}^{\gamma}}(SID_j, ID_i, \mu_i)$ and sends (MAC_{S_j}, ω_j) to user U_i. Upon receiving (MAC_{S_j}, ω_j), user U_i's smartcard computes k_{ij}^{γ} and checks whether MAC_{S_j} is correct. If it holds, mutually shared session key is correct. Then, user U_i's smartcard computes $MAC_{U_i} = h_{k_{ij}^{\gamma}}(ID_i, SID_j, \omega_j)$ and sends it to server S_j. Upon receiving MAC_{U_i}, server S_j checks whether MAC_{U_i} is correct. If it holds, shared session key confirmation is complete.

3.3 The γ Time Authentication Phase

After the first time authentication, user U_i and server S_j perform following steps where authentication time γ starts from 1 to b_{ij} which has been defined by user U_i in previous phase (see Figure 2). First, user U_i enters PW_i and Bio_i, and smartcard checks PW_i and Bio_i. If it holds, user U_i computes $N_1 = h^{(b+1+(\gamma+1))}(v_{ij})$, $V_{ij}^{\gamma} = E_{k_{ij}^{\gamma}}(N_1)$, $k_{ij}^{\gamma+1} = (N_1, k_{ij}^{\gamma})$ and sends V_{ij}^{γ} to server S_j. Then, server S_j obtains N_1 by decrypting V_{ij}^{γ} using session key k_{ij}^{γ} and verifies $h(N_1)$ with N_1. If it holds, $h(N_1)$ replaces N_1. After that, server S_j computes a new session key $k_{ji}^{\gamma+1}$ to replace k_{ij}^{γ} and $B_{ij}^{\gamma+1}$. Then, server S_j stores N_1 and sends $B_{ij}^{\gamma+1} = E_{k_{ij}^{\gamma+1}}(N_1)$ to user U_i. After receiving $B_{ij}^{\gamma+1}$, user U_i uses $k_{ij}^{\gamma+1}$ to verify $B_{ij}^{\gamma+1}$. If it holds, the γ time authentication phase is completed, and user U_i can communicate with server S_j using new session key $k_{ij}^{\gamma+1}$.

3.4 Offline Password Change Phase

User U_i can change password after entering (PW_i, Bio_i) and new password PW_i'. Then, smartcard updates $A_i = h_{\alpha_i}(PW_i', Bio_i) \bigoplus (PW_i, Bio_i) \bigoplus h_{\alpha_i}(y_i, SID_j)$ and stores updated A_i.

3.5 Revocation Phase

If user U_i no longer wants to use service of server S_j, user U_i server S_j can perform revocation phase to remove identity and authority of user U_i in server S_j. After user U_i enters (ID_i, PW_i, Bio_i), smartcard checks PW_i, Bio_i, utilizes (y_i, x) to compute A_i and sends $(ID_i, A_i, SmartcardRevocationRequest)$ to server S_j. Then, server S_j searches ID_i in revocation table, computes $x^{new} = x+1$, and stores (ID_i, x^{new}) in revocation table. Server S_j computes $u_i^{new} = h_{\beta_j}(x^{new}, ID_i)$, $u_j^{new} = h_{\beta_j}(x^{new}, SID_j)$, $B_i^{new} = u_i^{new} \bigoplus A_i$, and $B_j^{new} = u_j^{new} \bigoplus A_i$ and sends (B_i^{new}, B_j^{new}) to user U_i. Finally, user U_i's smartcard replaces (B_i, B_j) with (B_i^{new}, B_j^{new}).

4 Security Analysis

This paper applies BAN logic [4] for formal security proof. Also, we present theoretical analyses to prove proposed scheme could achieve security requirements.

4.1 Formal Security Proof Using BAN Logic

The process of proof is similar with some schemes, because BAN logic [4] aims to prove that principles in schemes

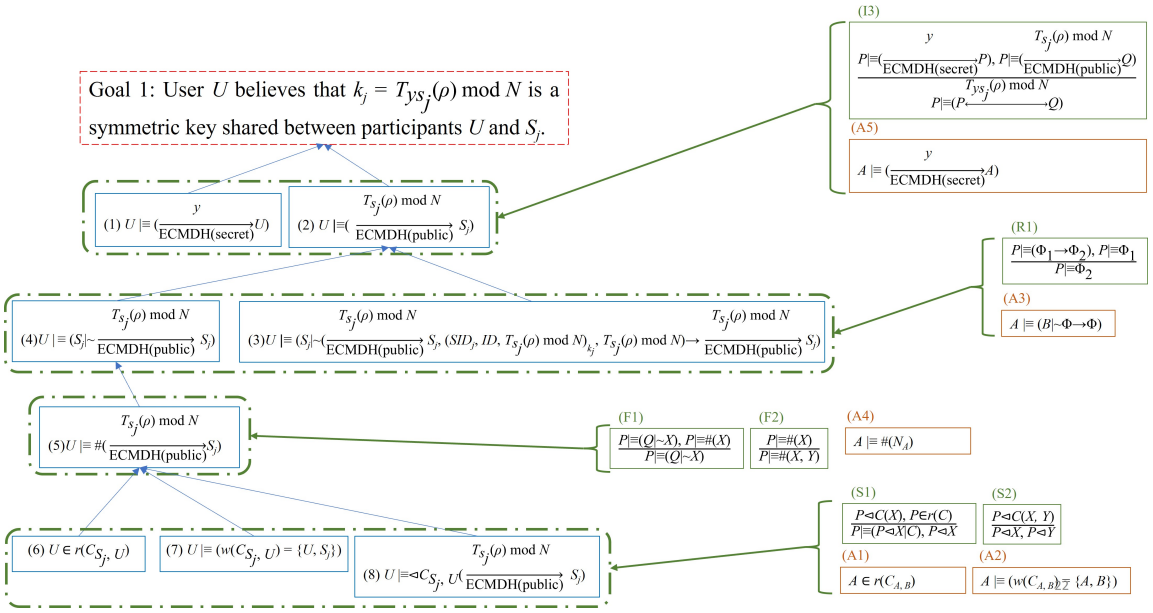


Figure 3: The process of proving Goal 1

can believe the established session keys. In BAN logic, goals have to be achieved, and proposed scheme have four goal. Goal 1 is that user U_i believes that k_{ij}^{γ} is a symmetric key shared between participants U_i and S_j; Goal 2 is that server S_j believes that k_{ij}^{γ} is a symmetric key shared between participants U_i and S_j; Goal 3 is that user U_i believes that server S_j believes k_{ij}^{γ} which is a symmetric key shared between participants U_i and S_j; Goal 4 is that server S_j believes that user U_i believes k_{ij}^{γ} which is a symmetric key shared between participants U_i and S_j. We show the process in Figures 3 and 4 because of pages limitation. Notice that proposed scheme realizes Goal 2 by using the same arguments of Goal 1, and proposed scheme realizes Goal 4 by using the same arguments of Goal 3.

4.2 Preventing MITM Attack

In order to prevent MITM attack, user U_i and server S_j can confirm the message is resent, modified, and replaced or not by checking information in message authentication codes MAC_{S_j} and MAC_{U_i}. User U_i verifies MAC_{S_j}, and server S_j verifies MAC_{U_i} in authenticated key exchange phase of proposed scheme. By this way, adversary cannot modify message authentication codes MAC_{S_j} and MAC_{U_i} without session key k_{ij}^{γ}.Thus, proposed scheme can prevent MITM attack.

4.3 Key Confirmation

User U_i can check session key k_{ij}^{γ} by verifying MAC_{S_j}, and server S_j can also check session key k_{ji}^{γ} by verifying MAC_{U_i}. As a result, proposed scheme allows for key confirmation.

4.4 Preventing Impersonating and Server Spoofing Attacks

We store a user's random number y_i in the smartcard, so an adversary can only obtain B_j even getting USB. If an adversary wants to attack, adversary should have user U_i's smartcard, correct password, and even biometric features. We store user U_i's random number y_i in smartcard, and it is hard to obtain information from smartcard. Furthermore, the number of times that a password can be entered is limited; if the number of attempts to enter a password exceeds allowable number of attempts, smartcard will get locked. As a result, proposed scheme can prevent impersonation and server spoofing attacks.

4.5 User Anonymity

A user U_i's identity is protected by encrypting ID_i in C_i with K_i, and server S_j must obtain K_i first. An adversary cannot obtain ID_i even if adversary obtains R_i and C_i because only server S_j knows secret x_{S_j}. The adversary cannot obtain K_i without knowing x_{S_j} and decrypt C_i, so adversary cannot obtain ID_i. In this way, proposed scheme provides user anonymity.

4.6 Resistant to Bergamo*et al.*'s Attack

Bergamo*et al.*'s attack is based on two situations [3]. First, adversary is able to obtain related elements (x, ρ_i, μ_i, ω_j). Second, several Chebyshev polynomials pass through the same point due to periodicity of cosine function. In proposed scheme, adversary is unable to obtain any related elements (x, ρ_i, μ_i, ω_j) because of being encrypted in transmitted messages which only user U_i and

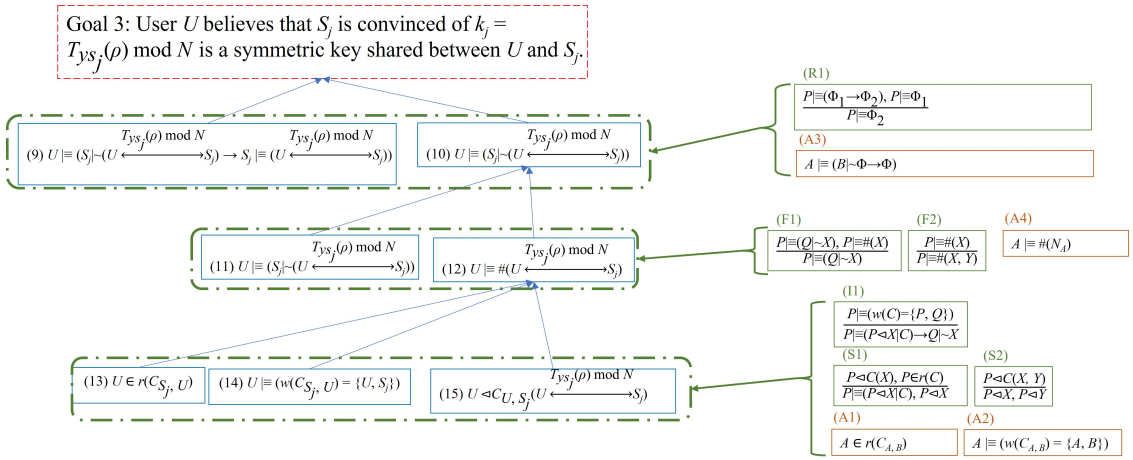


Figure 4: The process of proving Goal 3

server S_j can retrieve decryption key. Moreover, proposed protocol utilizes extended Chebyshev polynomials, in which periodicity of cosine function is avoided by extending interval of x to $(-\infty, +\infty)$ [25]. As a result, proposed scheme can resist Bergamo*et al.*'s attack [3].

5 Performance Analysis

We prove that proposed scheme can be more efficient than other similar schemes. Results of proposed scheme can refer to Lin *et al.*'s [15] because of similar results.

6 Conclusions

We proposed a CMPP-3FA scheme for telemedicine systems, which could achieve some general security requirements, such as preventing MITM attack, preventing impersonation and server spoofing attacks, providing user anonymity, and resisting Bergamo*et al.*'s attack [3]. Proposed scheme establishes a secure communication channel using authenticated session keys between patients and services of telemedicine systems, without threats of eavesdrop, impersonation, etc., and allow patients and medical professionals access to multiple telemedicine services with password, smartcard, and biometric feature. Patients have data ownership because patient can control and decide data's destination and time of transmission. If a patient would like to have an appointment with medical professionals, no matter individual or group meeting, through telemedicine system, patient and medical professionals have to use smartcards and complete mutual authentication and session key establishment. Communications between patient and medical professionals is protected. Such scenario can apply proposed scheme which can protect privacy of patients and measured biodata while transmitting data. Some issues can be discussed in the future, such as communication between wearable devices and server of medical institute, approach of applying three-factor authentication with user-friendly interface and quality of services while maintaining privacy preservation, etc.

References

[1] G. Abel, P. Istvan, and A. Attila, "Revolutionizing healthcare with iot and cognitive, cloud-based telemedicine," *Acta Polytechnica Hungarica*, vol. 16, no. 2, SI, 2019.

[2] A. Ahad, M. Tahir, and K. A. Yau, "5g-based smart healthcare network : Architecture , taxonomy , challenges and future research directions," *IEEE Access*, vol. 7, pp. 100747–100762, 2019.

[3] P. Bergamo, P. D'Arco, A. De Santis, and L. Kocarev, "Security of public-key cryptosystems based on chebyshev polynomials," *IEEE Transactions on Circuits and Systems I-Regular Papers*, vol. 52, no. 7, pp. 1382–1393, 2005.

[4] M. Burrows, M. Abadi, and Roger Michael Needham, "A logic of authentication," *Proceedings of the Royal Society of London Series A*, vol. 426, no. 1871, p. 233–271, 1989.

[5] C. L. Hsu, T. V. Le, M. C. Hsieh, K. Y. Tsai, C. F. Lu, and T. W. Lin, "Three-factor ucsso scheme with fast authentication and privacy protection for telecare medicine information systems," *IEEE Access*, vol. 8, pp. 196553–196566, 2020.

[6] Q. Jiang, M.K. Khan, X. Lu, J. Ma, and D. He, "A privacy preserving three-factor authentication protocol for e-health clouds," *The Journal of Supercomputing*, vol. 72, no. 10, pp. 3826–3849, 2016.

[7] H. Kaur and P. Khanna, "Privacy preserving remote multi-server biometric authentication using cancelable biometrics and secret sharing," *Future Generation Computer Systems*, vol. 102, pp. 30–41, 2020.

[8] C. Lalit and B. Rabindranath, "A comprehensive survey on internet of things (iot) toward 5g wireless

systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.

[9] T. F. Lee, C. H. Hsiao, S. H. Hwang, and T. H. Lin, "Enhanced smartcard-based password-authenticated key agreement using extended chaotic maps," *PLOS ONE*, vol. 12, no. 7, 2017.

[10] S. Li, L. D. Xu, and S. Zhao, "5G internet of things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.

[11] H. Y. Lin, "Improved chaotic maps-based password-authenticated key agreement using smart cards," *Communications in Nonlinear Science and Numerical Simulation*, vol. 20, no. 2, pp. 482–488, 2015.

[12] T. Z. Lin, "A privacy-preserved id-based secure communication scheme in 5g-iot telemedicine systems," *Sensors*, vol. 22, no. 18, 2022.

[13] T. Z. Lin and C. L. Hsu, "Anonymous group key agreement protocol for multi-server and mobile environments based on chebyshev chaotic maps," *The Journal of Supercomputing*, vol. 77, no. 9, pp. 4521–4541, 2018.

[14] T. Z. Lin and C. L. Hsu, "Faidm for medical privacy protection in 5g telemedicine systems," *Applied Sciences-Basel*, vol. 11, no. 3, 2021.

[15] T. Z. Lin, C. L. Hsu, T. V. Le, C. F. Lu, and B. Y. Huang, "A smartcard-based user-controlled single sign-on for privacy preservation in 5g-iot telemedicine system," *Sensors*, vol. 21, no. 8, 2021.

[16] Z. Liu, Y. Hong, and D. Pi, "A large-scale study of web password habits of chinese network users," *Journal of Software*, vol. 9, pp. 293–297, 2014.

[17] A. Mamta, R. Abhishek, and S. Navrati, "Next generation 5g wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[18] D. Mishra, S. Mukhopadhyay, S. Kumari, M. K. Khan, and A. Chaturvedi, "Security enhancement of a biometric based authentication scheme for telecare medicine information systems with nonce," *Journal of Medical Systems*, vol. 38, no. 5, 2014.

[19] Y. J. Park, J. E. Chung, and D. H. Shin, "The structuration of digital ecosystem, privacy, and big data intelligence," *American Behavioral Scientist*, vol. 62, no. 10, pp. 1319–1337, 2018.

[20] Y. J. Park and D. H. Shin, "Contextualizing privacy on health-related use of information technology," *Computers in Human Behavior*, vol. 105, 2020.

[21] K. Renuka, S. Kumari, and X. Li, "Design of a secure three-factor authentication scheme for smart healthcare," *Journal of Medical Systems*, vol. 43, no. 5, 2019.

[22] V. Sureshkumar, R. Amin, M. S. Obaidat, and I. Karthikeyan, "An enhanced mutual authentication and key establishment protocol for tmis using chaotic map," *Journal of Information Security and Applications*, vol. 53, pp. 2214–2126, 2020.

[23] E. J. Yoon and I. S. Jeon, "An efficient and secure diffie-hellman key agreement protocol based on chebyshev chaotic map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 6, pp. 2383–2389, 2011.

[24] E. J. Yoon and K. Y. Yoo, "Cryptanalysis of group key agreement protocol based on chaotic hash function," *IEICE Transactions on Information and Systems*, vol. E94D, no. 11, pp. 2167–2170, 2011.

[25] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos Solitions & Ftactals*, vol. 37, no. 3, pp. 669–674, 2008.

[26] L. Zhang, Y. Zhang, S. Tang, and H. Luo, "Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2795–2805, 2018.

[27] I. A. Zriqat and A. M. Altamimi, "Security and privacy issues in ehealthcare systems: Towards trusted services," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 9, 2016.

Biography

Tzu-Wei Lin is an assistant professor of *i.* School, Feng Chia University (FCU), Taiwan, who is also director of Information Security Office, Office of Information Technology, FCU. received PhD. He received PhD. degree in Graduate Institute of Business and Management, Chang Gung University (CGU), Taiwan, at June 2021. He received B.S. and M.S. degree in Department of Information Management from CGU, Taiwan, in 2011 and 2013, respectively. He worked in Information Security Group, Information Technology Services, Academia Sinica, Taiwan from 2013 to 2016. His research interests are computer and communication security, information security, applied cryptography, Internet of Things, and wearable healthcare system.

Chien-Lung Hsu received his M.S and Ph.D. degree in Information Management from National Taiwan University of Science and Technology in 1997 and 2002 respectively. He is currently a professor of Department of Information Management and Graduate Institute of Business and Management, CGU, Taiwan. His research interests include smart home, mobile commence, computer and communication security, information security, applied cryptography, healthcare, digital right management, auto identification technology, and user centered service. He received lots of honors, awards, certificates in term of information security in his research and had a great number of publications in the related fields. He is also the member of Institute of Information & Computing Machinery (IICM) and Chinese Cryptology and Information Security Association (CCISA)..

Overlapping Difference Expansion Reversible Data Hiding

Chin-Feng Lee[1], Jau-Ji Shen[2], and Chin-Yung Wu[2]
(Corresponding author: Chin-Feng Lee)
Department of Information Management, Chaoyang University of Technology[1]
Taichung 41349, Taiwan, ROC
Email: lcf@cyut.edu.tw
TManagement Information Systems, National Chung Hsing University[2]
Taichung, Taiwan, ROC
(Received Aug. 15, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)
The Special Issue on Trusted ICT Technologies on the Smart Society and Secure Multimedia Applications
Special Editor: Prof. Chin-Feng Lee (Chaoyang University of Technology)

Abstract

Rapid technological development has made information transmission convenient by using the internet in the shortest possible time. Despite this, there is an increase in information security problems. Therefore, the need for information hiding is of paramount importance to the public. The difference expansion (DE) embedding technology has good embedding capacity in reversible data hiding (RDH) technology. However, the DE embedding technique creates problems in the multi-layer embedding process. This study proposes an embedding strategy of horizontal overlapping and vertical overlapping difference expansion (also called HoVo_DE for short) to enhance the information embedding of the DE series methods and improve the problem of difference expansion cascading caused by multiple DE embedding. The experimental results show that the proposed strategy reduces the distortion of image quality and has a higher embedding capacity than other strategies.

Keywords: Data Hiding; Reversible Data Hiding; Difference Expansion (DE); Improved Reduced Difference Expansion (IRDE)

1 Introduction

The emergence of communication software and social platforms has significantly enhanced communication and information sharing in the shortest possible time. One of the most integral parts of network operation and maintenance is network security. A large number of multimedia forensics and network security technologies have aroused extensive research attention in multimedia data integrity assessment [11]. Also, there is a rapid increase in information security problems. Thus, transmitting messages safely and freely on the internet has become a concern. Information hiding [3] involves embedding secret data in digital multimedia carriers, such as images or documents. In the case of digital images, the difference between the cover image before embedding the secret message and the stego-image after embedding should be as far as possible to ensure the secrecy and security of the embedded secret information [1, 4]. This way, the slight difference between the mask and the camouflage image can effectively prevent encrypted information from being detected or deciphered during the transmission process. Information hiding can be classified as irreversible data hiding [6] and reversible data hiding (RDH) [2, 5, 7–10, 12–17].

Among several RDH methods, the difference expansion (DE) method proposed by Tian in 2003 [17], the prediction error expansion (PEE) method proposed by Thodi *et al.* in 2004 [12], histogram shifting (HS) method proposed by Ni *et al.* in 2006 [7], and pixel value ordering (PVO) method proposed by Li *et al.* in 2013 [8]. Among these methods, the highly efficient and well-known is the DE method, which can achieve the effect of high embedding quantity through a relatively simple and low-complexity operation process. The DE method uses the difference between the adjacent pixel pair and the average value to embed the secret data at the adjacent pixel pair such that the average value at the adjacent pixel pair remains unchanged.

The intuitive and concise methods for embedding secret data through the difference expansion between pixel pair have significantly attracted much interest to optimize the DE and derive several DE series methods with better performance [5,14,15]. Liu *et al.* [2] proposed the reduced DE (RDE) strategy in 2007 to solve the image distortion problem of the DE scheme. After the secret information is embedded, the RDE strategy can reduce the original difference between the two stego-pixels to enable the re-

duction of the degree of distortion. Then, Yi et al. [9] proposed the improved reduced difference expansion (also referred as IRDE) method in 2009. The IRDE method uses a logarithmic transformation function to reduce the difference first before the secret information has been embedded so that the difference between stego-pixels can be minimized more effectively, thereby maintaining good image quality. Alattar [10] proposed the Quad-DE method which introduces the difference expansion hiding within a group of pixels on the basis of 2×2 blocks. Lee *et al.* proposed block-shiftable embedding (BSE) method [13] based on the DE scheme. The method of block displacement increases the embedded embedding capacity while retaining the image quality within an acceptable range.

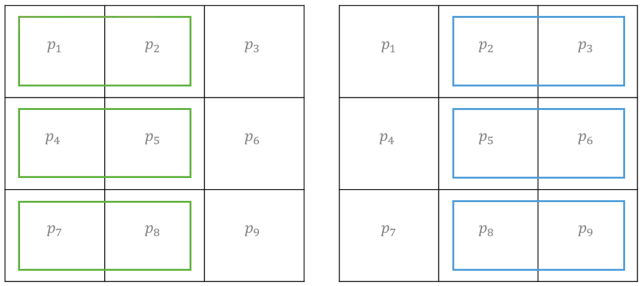
The DE embedding technology has a good embedding capacity; however, there is an inherent weakness in the DE embedding scheme, i.e., it will cause the cascading problem due to continuous differential expansion in the multi-layer embedding process. Each embedding represents an expansion of the difference value, so after the difference value of the multi-layer embedded image is expanded to a degree, the relative image quality dramatically decreases. Therefore, this study proposes an embedding strategy of horizontal overlapping and vertical overlapping DE, also called HoVo_DE for short to simultaneously enhance the embedding capacity and improve the problem of DE cascading caused by multiple-layer embedding. Exploring the relationship between different block sizes and making full use of the image quality in the blocks, the proposed embedding strategy of HoVo_DE can effectively suppress the occurrence of the DE cascading problem.

2 Block-Shiftable Embedding

Lee *et al.* proposed the block-shiftable embedding (also referred as BSE) [13] strategy for reversible data hiding. Because the traditional multi-layer embedding strategy can adversely distort the image quality, the BSE strategy shifts the block and changes the embedding starting point during the second layer embedding, which can improve the image quality and increase the embedding capacity.

The BSE strategy employs the DE manipulation with different block segmentation for embedding. The first step is to divide the original image I of size $W \times H$ into non-overlapping pixel pairs of size 1×2 individually. The pixel pairs of the first layer and second layer are as shown in Figures 1(a) and 1(b) respectively. Then the BSE performs the DE embedding operation on each pixel pair in the sequence.

In the previous method, the secret data will be embedded in one layer and then start to embed the remaining information into the second layer. However, the BSE strategy spreads the secret data in two layers according to the division mode of different blocks with a proper threshold. By using different partitioning schemes for each embedding layer, the predictions become adaptive, creating more room for improvement over previous state-of-the-art methods. More, secret data will be hidden in smoother blocks and thus have better image quality.



(a) Segmentation of First Layer (b) Segmentation of Second layer

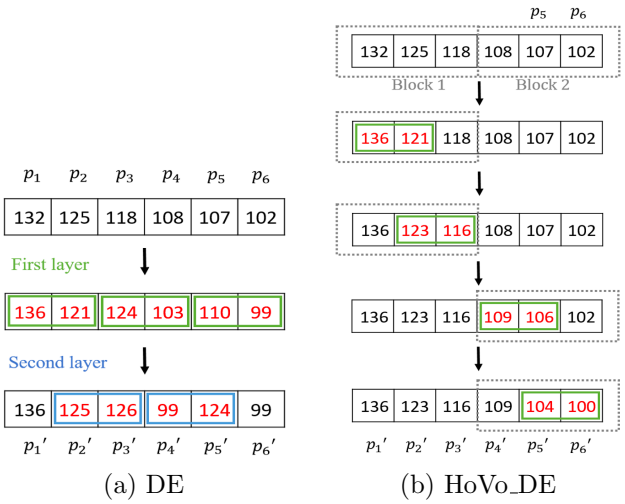
Figure 1: The Block-Shiftable embedding strategy.

3 Proposed Method

The problem of difference expansion cascading exists in the traditional DE and the IRDE methods, after multiple-layer embedding has been performed to a certain extent. The differential expansion cascade problem will prevent adjacent pixel pairs from being able to hide data or cause serious image distortion. We propose an embedding strategy of horizontal overlapping and vertical overlapping DE strategy, also called HoVo_DE embedding strategy for short, which is different from the previous DE based methods and performs overlapping embedding in the horizontal and vertical directions, respectively. In the experiment, we applied the HoVo_DE embedding strategy under different block segmentation modes and discussed the cover image's embedding capacity and the stego-image's quality.

3.1 HoVo Difference Embedding Strategy

The HoVo difference embedding strategy (also called as the HoVo_DE embedding strategy for short) proposed in this study divides the original image into non-overlapping blocks with $M \times N$ pixels before embedding. Independent embedding is performed in each block. Thus, the differential expansion cascading can be limited to the current block to avoid affecting the pixels of other blocks. The division of blocks can suppress the chain reaction of differential expansion cascading and reduce the degree of image quality damage. Figure 2 shows that, compared with the traditional multi-layer embedding mode, our proposed HoVo_DE embedding strategy, performing information hiding after block division can suppress the rapid expansion of camouflaged pixel values caused by DE cascades to a certain extent.



(a) DE (b) HoVo_DE

Figure 2: Difference expansion cascading methods. (a)DE method (b) the HoVo_DE embedding method

The proposed HoVo_DE embedding strategy uses four different block division modes, namely, 3×3, 4×4, 5×5, and 6×6. Figure 3 below shows the group overlapping modes with the block size of 6×6.

3.2 Embedding Procedure

The embedding steps are as follows.

Algorithm 1 The embedding procedure

1: Obtain the cover image I with an image size of $W \times H$.
2: Divide I into non-overlapping blocks of size $M \times N$ in sequence from top to bottom and left to right.
3: Each block is divided into groups. There are $M \times (N-1)$ groups in the horizontal direction and $N \times (M-1)$ groups in the vertical direction; that is, a total of $M(N-1)+N(M-1)$ groups in a block.
4: According to the HoVo_DE embedding strategy, the information hiding method of DE or IRDE is employed to hide the secret data bits in each group of every row by overlapping groups from both ends of the block to the middle of the block.
5: The DE or IRDE method hides the secret data bits in each group of every column by overlapping groups from both ends of the block to the middle of the block.

3.3 Example of HoVo_DE Embedding Strategy

The following example comes from the HoVo_DE embedding strategy applied to the 3×3 block size. There are a total of $3 \times (3-1) + 3 \times (3-1) = 12$ groups can be used in one block.

Assume that the first group, denoted by $g_1 = (p_1, p_2) = (132, 125)$. Assuming that the to-be hidden secret data bit is 1, after the DE method is applied to the first group, we can obtain $g_1 = (p_1', p_2') = (136, 121)$. Next, the embedding operation of g_2 is performed. Therefore, the pixel values used in the group g_2 are (p_2', p_3). Repeat the embedding step in sequence until all groups are embedded. Figure 4 presents the flowchart of HoVo_DE embedding strategy in block size 3×3.

While taking out the secret information and restoring the original image, the operations are performed on each group individually in reverse order, the order should be $g_{12}', g_{11}', g_{10}', \ldots, g_2'$, and g_1', respectively.

4 Experimental Results

This experiment was performed using MATLAB R2020a installed on the Intel (R) Core (TM) i7-9750 H CPU 2.60 GH-z, 16 GB RAM environment. To quantify the effectiveness of this experimental method and compare it with other related methods, eight standard grayscale images of size 512×512 are used to test the data hiding performance. The test images are Lena, Airplane, Baboon, Boat, Peppers, and Elaine, which can be seen in Figure 5. Additionally, the secret messages hidden in this experiment were all generated using binary random numbers (0 and 1).

This experiment uses two metrics to evaluate the performance of the experimental results, namely, embedding capacity (EC) and peak signal-to-noise ratio (PSNR). The EC represents the amount of secret information in bits that can be embedded in an image. The higher the value of EC, the more information is embedded. The pixel similarity between the cover image and the stego-image is calculated using the PSNR metric. If the PSNR value is high, the more similar the stego-image will be to the original image.

The following sections explore the embedding methods based on DE [17] and IRDE [9] and use the HoVo_DE embedding strategy to achieve the embedding effect under different block sizes.

4.1 Embedding capacity of HoVo_DE Embedding Strategy

This section presents the results of a series of methods using the proposed HoVo_DE embedding strategy. We used the DE and IRDE methods to conduct embedding experiments under different block sizes and compared the results of the DE [17], IRDE [9], and BSE [13] methods. This study mainly explores whether the HoVo_DE embedding strategy can obtain better embedding capacity and image quality with the block size expansion. Under the experiments of various block sizes, the average performance of the IRDE is better than that of the DE. The detailed experimental data below also confirm this findings.

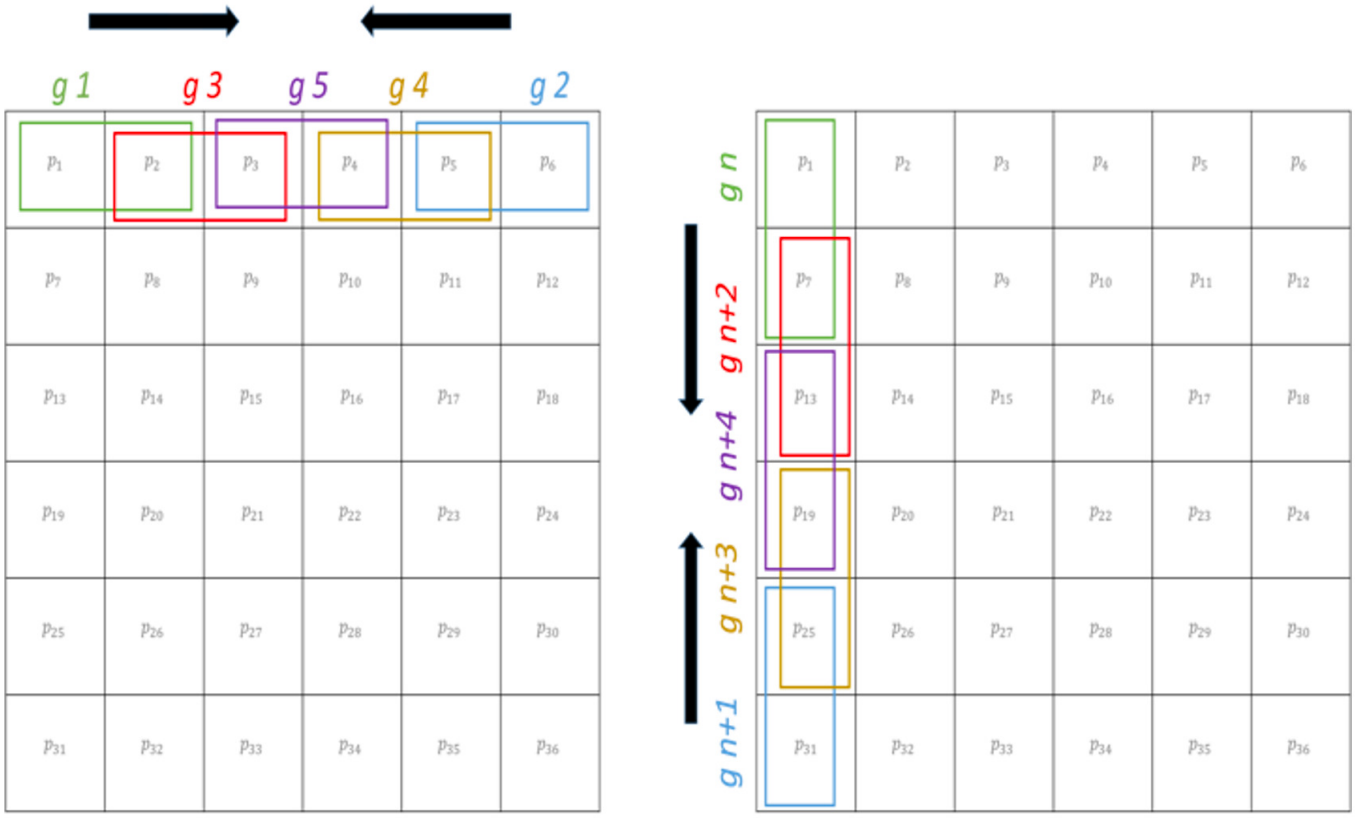


Figure 3: HoVo_DE embedding strategy in block size 6×6

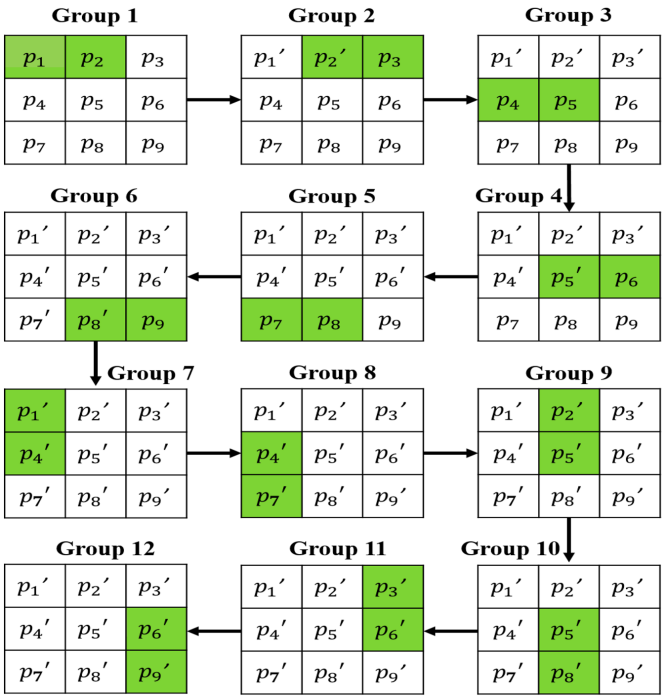
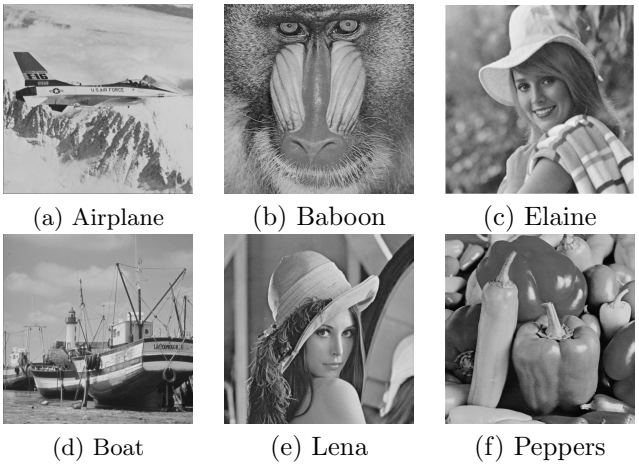


Figure 4: Embedding flowchart of HoV_DE embedding strategy in block size 3×3.



(a) Airplane (b) Baboon (c) Elaine

(d) Boat (e) Lena (f) Peppers

Figure 5: Testing images: (a) Airplane, (b) Baboon, (c) Elaine, (d) Boat, (e) Lena, and (f) Peppers.

4.1.1 HoVo_DE Embedding Strategy Using the DE Method

We implemented the HoVo_DE embedding strategy using the DE and measured the size of different blocks based on the concept of dicing. Furthermore, we determined whether larger blocks can be used to obtain higher embedding capacity while maintaining a certain level of storage image quality. In the following table, DE represents the original DE method; BSE_M×N represents the BSE method implemented in the $M \times N$ block by the DE method; HoVo_DE_M×N represents the use of the HoVo_DE embedding strategy to implement the DE method based on the $M \times N$ block division mode. By implementing HoVo_DE_3×3, HoVo_DE_4×4, HoVo_DE_5×5, and HoVo_DE_6×6, with Lee *et al.*'s BSE_1×4 and BSE_2×2 methods [13], and the original DE method [17]. The table below shows the embedding capacity and image quality for method comparison.

From Table 1, the average embedding capacity of HoVo_DE_3×3, HoVo_DE_4×4, HoVo_DE_5×5, and HoVo_DE_6×6 are 344,951, 390,761, 413,258, 430,226 bits, respectively. We observed that the embedding capacity can be effectively increased when the HoVo_DE embedding strategy is implemented in a larger block division mode. Moreover, compared with our proposed method, the original DE method has only 130,720 bits, and the BSE_2×2 method implemented by DE has 261,675 bits in terms of the average embedding capacity. In this study, the average embedding capacity of the method in the block division modes of HoVoDE_3×3, HoVoDE_4×4, HoVoDE_5×5, and HoVoDE_6×6 are 2.64 times, 2.99 times, 3.16 times, 3.29 times than that of the DE method, respectively. They are also 1.32, 1.49, 1.58, and 1.65 times than that of the BSE_2×2 method, respectively.

Table 2 shows the stego-image quality measured by PSNR values by the original DE method, BSE_1×4 method, BSE_2×2 method, and our proposed methods with four block division modes which are HoVo_DE_3×3, HoVo_DE_4×4, HoVo_DE_5×5, and HoVo_DE_6×6, respectively.

Table 1: Embedding Capacity of Original DE, BSE and the proposed HoVo DE Embedding Methods

EC	DE	BSE_ 1 × 4	BSE_ 2 × 2	HoVo DE_ 3 × 3	HoVo DE_ 4 × 4	HoVo DE_ 5 × 5	HoVo DE_ 6 × 6
Airplane	130700	196249	262045	346252	392637	415566	432859
Baboon	130954	196597	261061	341373	386262	408236	424544
Elaine	131060	196560	262012	345374	390707	412548	429477
Boat	130307	196030	261759	345287	391277	413895	430811
Lena	131014	196590	262113	346604	393034	415944	433263
Peppers	130283	196060	261059	344814	390648	413361	430403
Average	130720	196348	261675	344951	390761	413258	430226

When comparing the BSE and HoVo_DE embedding methods for the Airplane, Lena, Peppers images with clear black and white boundaries, in Table 2, the PSNR performance of BSE_1×4 is the best. To the best of our knowledge, this is due to the different block-division structures. As a result, the BSE_1×4 blocks have a rectangular structure, whereas the blocks of BSE_2×2, HoVo_DE_3×3, HoVo_DE_4×4, HoVo_DE_5×5, and HoVo_DE_6×6 are all square structures. Therefore, BSE_2×2, HoVo_DE_3×3, HoVo_DE_4×4, HoVo_DE_5×5, and HoVo_DE_6×6 have a high probability of overlapping with the black and white boundaries in the images when calculating the pixel differences. The difference at the boundary will be relatively large and the image quality will be degraded. Also, from the above experimental results, the proposed method can successfully obtain a larger embedding capacity than other methods using a larger block size division strategy. With an increase in the block size, the image quality declines but maintains a certain level.

Table 2: PSNR of Original DE, BSE and the proposed HoVo DE Embedding Methods

PSNR	DE	BSE_ 1 × 4	BSE_ 2 × 2	HoVo DE_ 3 × 3	HoVo DE_ 4 × 4	HoVo DE_ 5 × 5	HoVo DE_ 6 × 6
Airplane	32.63	31.86	30.58	29.22	26.86	26.79	26.46
Baboon	27.16	28.55	22.53	20.50	19.78	19.53	19.33
Elaine	33.41	30.01	26.57	21.84	20.21	19.51	19.16
Boat	30.59	28.56	27.94	25.54	24.80	24.22	23.95
Lena	33.15	33.54	30.58	29.15	28.59	28.29	27.98
Peppers	33.63	34.69	31.12	29.00	27.92	27.50	27.16
Average	31.76	31.20	27.94	25.62	24.69	24.29	23.98

4.1.2 HoVo_DE Embedding Strategy Using the IRDE Method

We implemented the proposed HoVo_DE embedding strategy using the IRDE method [9] and measured the size of different block divisions. Furthermore, we test whether larger blocks can be used to obtain higher embedding capacity while maintaining a certain level of storage image quality. In the following table, IRDE represents the IRDE method; BSE_M×N represents the BSE method implemented in the $M \times N$ block by the IRDE method; HoVo_IRDE_M×N represents the use of the HoVo_DE embedding strategy to implement the IRDE method based on the $M \times N$ block division mode. By implementing HoVo_IRDE_3×3, HoVo_IRDE_4×4, HoVo_IRDE_5×5, and HoVo_IRDE_6×6, with Lee *et al.*'s BSE_1×4 and BSE_2×2 methods [9], and Yi *et al.*'s IRDE methods [9]. The table below shows the embedding capacity and image quality for method comparison.

From Table 3, the average embedding capacity of HoVo_IRDE_3×3, HoVo_IRDE_4×4, HoVo_IRDE_5×5, and HoVo_IRDE_6×6 are 346,786, 393,197, 416,154, 433,496 bits, respectively. We observed that the embedding capacity can be effectively increased when the HoVo_DE embedding strategy is implemented in a larger block division mode. Moreover, compared with our proposed method, the IRDE method has only 131,053 bits, and the BSE_2×2 method implemented by IRDE has 262,111 bits, in terms of the average embedding capacity. In this study, the average embedding capacity of the method in the block division modes of HoVo_IRDE_3×3, HoVo_IRDE_4×4, HoVo_IRDE_5×5,

and HoVo_IRDE_6×6 are 2.65 times, 3.00 times, 3.18 times, 3.30 times that of the IRDE method, respectively. They are also 1.32, 1.50, 1.59, and 1.65 times that of the BSE_2×2 method, respectively. Table 4 shows the stego-image quality measured by PSNR values by the IRDE method, BSE_1×4 method, BSE_2×2 method, and our proposed methods with four block division modes.

Consider the Lena image with BSE_2×2 as an example; Table 3 shows the maximum embedding capacity of 262,145 bits, and Table 4 shows that the PSNR value is 44.28 dB. Table 3 shows that the embedding capacity of HoVo_IRDE_3×3 is increased by 84,656 (=346801-262145) bits compared with that of BSE_2×2. In Table 4, the Lena image using HoVo_IRDE_3×3 shows that its PSNR value is only 1.31 dB lower than that of BSE_2×2. It can be seen in Table 4 that the embedding capacity of the Lena image implemented by HoVo_IRDE_6×6 is increased by 171,356 (=433501-262145) bits compared with the BSE_2×2 method.

Tables 3 and 4 show that the HoVo_DE embedding strategy conducted by IRDE method achieves excellent results in embedding capacity and image quality. Consider the Airplane image as an example, we observe that the embedding capacity of HoVo_IRDE_6×6 in Table 3 is nearly 300% higher than that of the original IRDE, and Table 4 shows that the PSNR of HoVo_IRDE_6×6 can still reach an image quality of about 40 dB without significant degradation. Generally, the higher the embedding capacity, the greater the degree of image distortion. These data show that despite doubling the amount of information embedded, the level of image distortion did not drop in proportion to the sharp drop but decreased slightly to maintain good image quality.

Table 3: Embedding Capacity of IRDE, BSE and the proposed HoVo_IRDE Embedding Methods

EC	IRDE	BSE_1×4	BSE_2×2	HoVo DE_3×3	HoVo DE_4×4	HoVo DE_5×5	HoVo DE_6×6
Airplane	131072	196608	262144	346804	393217	416154	433473
Baboon	131070	196617	262107	346801	393215	416139	433516
Elaine	131071	196609	262128	346805	393117	416191	433506
Boat	131025	196573	262098	346764	393186	416182	433455
Lena	131072	196609	262145	346801	393207	416161	433501
Peppers	131006	196546	262046	346738	393127	416098	433524
Average	131053	196594	262111	346786	393197	416154	433496

Table 4: PSNR of IRDE, BSE and the proposed HoVo_IRDE Embedding Methods

PSNR	IRDE	BSE_1×4	BSE_2×2	HoVo DE_3×3	HoVo DE_4×4	HoVo DE_5×5	HoVo DE_6×6
Airplane	45.47	44.36	42.41	41.84	40.98	40.08	38.49
Baboon	42.11	41.70	37.92	36.49	35.91	35.60	34.02
Elaine	46.62	44.55	42.67	40.79	40.22	39.78	38.67
Boat	43.18	42.53	42.00	40.47	39.93	39.59	38.07
Lena	45.83	45.53	44.28	42.97	42.34	42.07	40.84
Peppers	46.70	46.44	44.40	42.90	42.38	42.06	40.81
Average	44.98	44.19	42.28	40.91	40.29	39.86	38.48

4.2 Image Quality Comparison of Related Methods

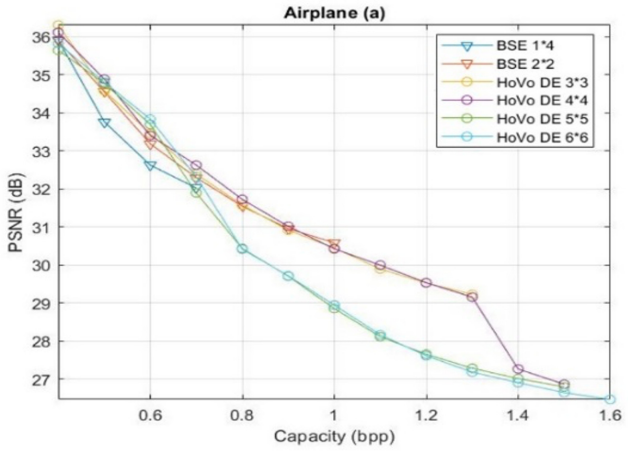
This section compares the embedding capacity based on each method and observes the value of PSNR after hiding the embedded secret information. Figures 6 (a) (f) present the visualized data graphs of the relative changes of EC and PSNR when the six test images were gradually embedded with data until the maximum embedding capacity attained that each image can carry.

4.2.1 HoVo_DE Embedding Strategy Applied to the DE Method

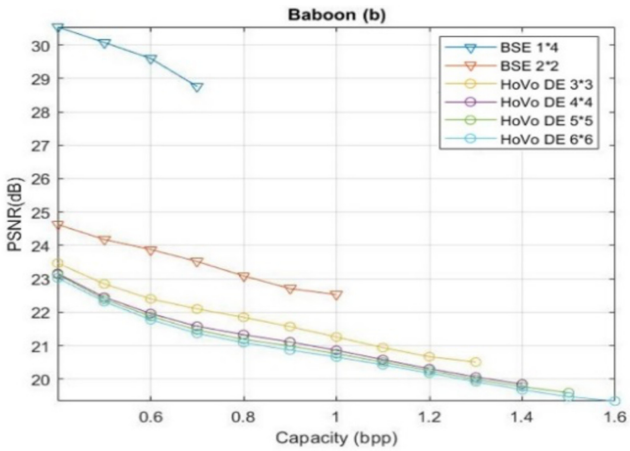
Figures 6 (a)-(f) show that in the HoVo_DE_3×3, HoVo_DE_4×4, HoVo_DE_5×5, or HoVo_DE_6×6 methods, the maximum embedding capacity of each proposed method is much higher than that of BSE_1×4 or BSE_2×2. Also, the maximum embedding capacity of the HoVo_DE_6×6 method can achieve 1.654 bpp on average. Although the image quality decreases with an increase in embedding capacity, the PSNR of the HoVo_DE embedding strategy cannot be far from the methods of BSE_1×4 or BSE_2×2. Consider the Airplane image in Figures 6(a) as an example when the embedding capacity is 1 bpp, the PSNR of the BSE_2×2 method is 30.58 dB, and the PSNR of the HoVo_DE_6×6 method is 28.9 dB, where the image quality difference between the two methods under the same EC is less than 2 dB. However, our proposed HoVo_DE_6×6 method can achieve a maximum embedding capacity of 1.654 bpp.

4.2.2 HoVo_DE Embedding Strategy Applied to the IRDE Method

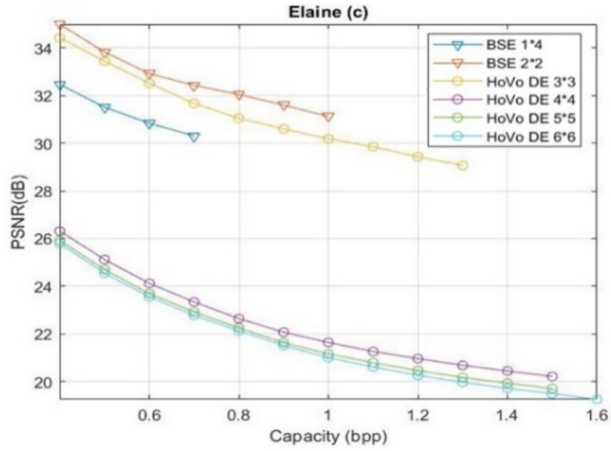
Figures 7(a)-(f) show that the proposed method proposed can significantly increase the EC. The maximum EC of the HoVo_IRDE_6×6 methods can reach 1.6 bpp, and the image quality can still maintain the PSNR above 40 dB, such as in Airplane, Lena, and Peppers. This means that our proposed block segmentation can suppress the effect of the difference expansion cascading to a certain extent. Also, the overlapping mechanism can offset the pixel differential value of the upper layer expansion. Consider Figure 7(a) as an example when the test image is an airplane and the embedding capacity is 1 bpp, the PSNR of the BSE_2×2 method is 42.41 dB. The PSNR values of our method slightly outperform those of the BSE_2×2 methods. This shows that the HoVo_DE embedding strategy has better image quality and larger embedding capacity than the BSE method at the same level of EC.



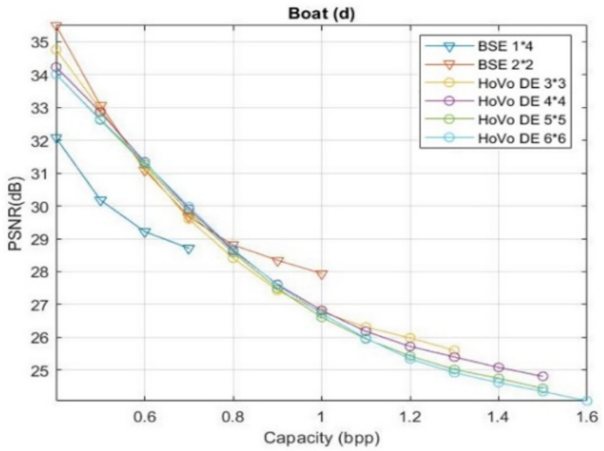
(a) Airplane



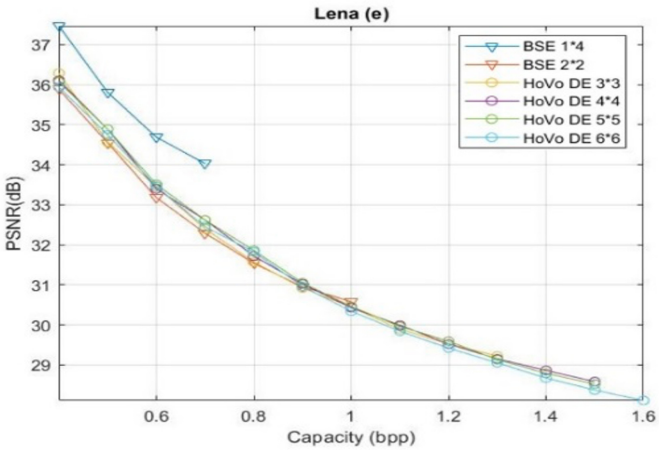
(b) Baboon



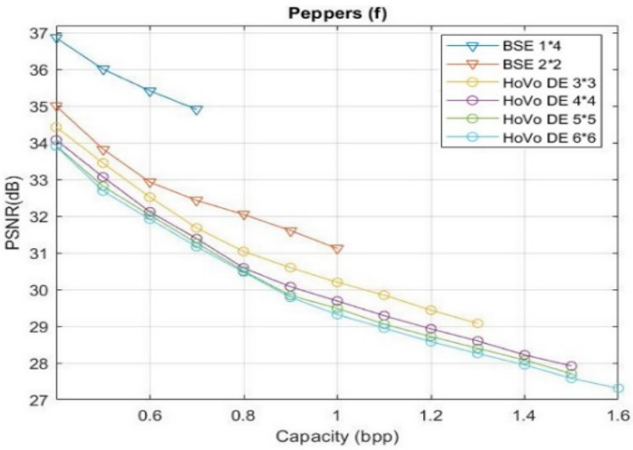
(c) Elaine



(d) Boat

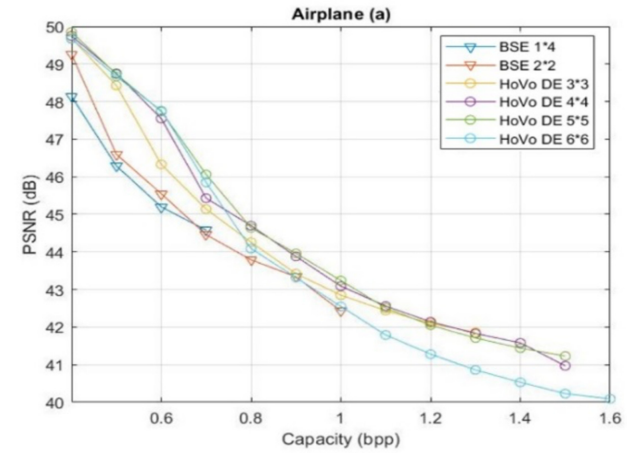


(e) Lena

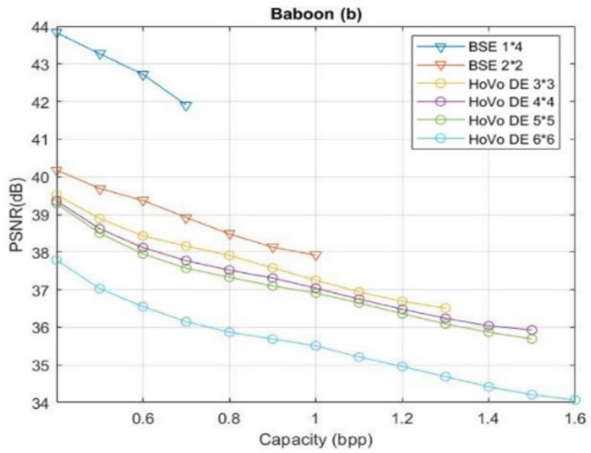


(f) Peppers

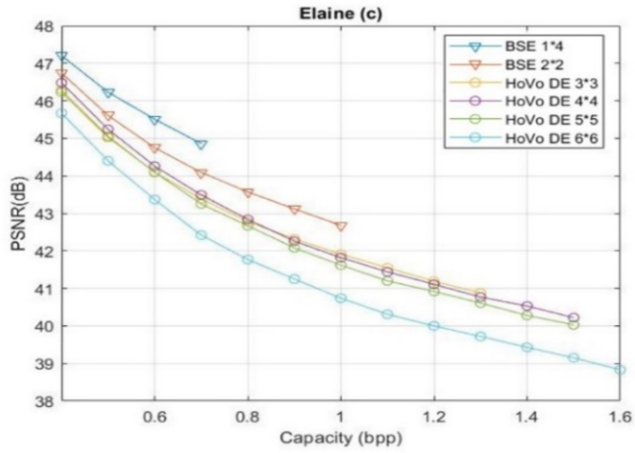
Figure 6: The performance of BES method and HoVo_DEmethod in image quality under different embedding capacity



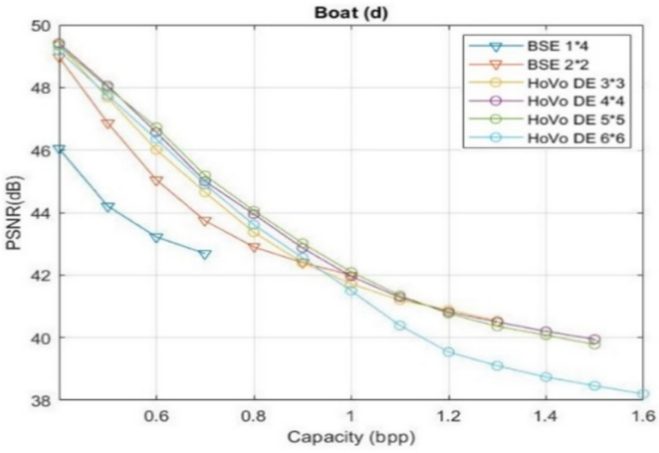
(a) Airplane



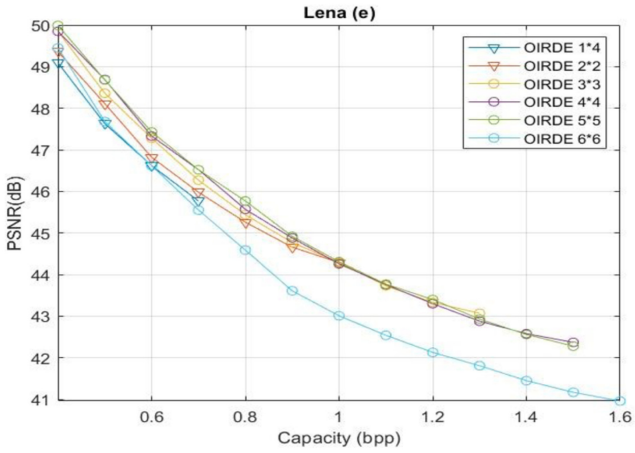
(b) Baboon



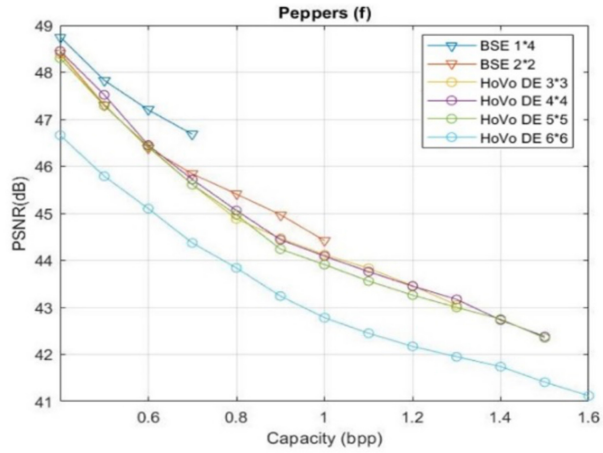
(c) Elaine



(d) Boat



(e) Lena



(f) Peppers

Figure 7: The performance of BES method and HoVo_IRD Emethod in image quality under different embedding capacity

4.3 Comparison of Image Quality Effects of HoVo_DE Embedding Strategy Implemented in Multi-layer Embedding

This study applies the HoVo_DE embedding strategy to multi-layer embedding. The stego-image generated by the original image using the HoVo_DE strategy through one layer of embedding becomes the input of the second layer of data hiding. This embedding method is called two-layer embedding. The following presents the image quality and embedding capacity obtained by the two-layer embedding based on the proposed HoVo_DE embedding strategy under different block-division modes.

Table 5 shows that the average embedding capacity of HoVo_DE_3×3 in a single layer embedding manner is 344951 bits, and the average embedding capacity of HoVo_DE_3×3 in a two-layer embedding manner can reach 643779 bits, which is an increase of 298828 bits compared with a single layer embedding. Taking the HoVo_DE_6×6 as an example, the average embedding capacity of the HoVo_DE_6×6 method is 430226 bits in a single layer embedding and 803300 bits in a two-layer embedding, which is an increase of 373074 bits compared with a single layer. However, the image quality also decreases with a larger EC. As shown in Table 5, whether it is the HoVo_DE_3×3, HoVo_DE_4×4, HoVo_DE_5×5, or HoVo_DE_6×6 methods, although the block sizes are different, their average embedding capacity in the two layers is 200% more than that in only one-layer embedding.

Figure 8 shows the average performance of six test images when the HoVo_DE embedding strategy is applied to two layers based on the DE method and as HoVo_DE_3×3(2L) when using the HoVo_DE_3×3 method to hide data with two-layer embedding. Similarly, when using HoVo_DE_4×4, HoVo_DE_5×5, and HoVo_DE_6×6 to hide data with two-layer embedding, they can be represented as HoVo_DE_4×4(2L), HoVo_DE_5×5(2L), and HoVo_DE_6×6(2L), respectively. In Figure 8, although the image quality suffers, the embedding capacity of the two layers can be improved significantly. Taking the HoVo_DE_6×6 method as an example, the average embedding capacity of a single layer HoVo_DE_6×6 is 1.6 bpp, and its PSNR value is about 25 dB. The average maximum embedding capacity of HoVo_DE_6×6(2L) can reach 3 bpp.

Table 5: Comparison of EC of single-layer and two-layer embedding based on HoVo DE method

EC	HoVo DE 3×3 1 Layer	HoVo DE 3×3 2 Layer	HoVo DE 4×4 1 Layer	HoVo DE 4×4 2 Layer	HoVo DE 5×5 1 Layer	HoVo DE 5×5 2 Layer	HoVo DE 6×6 1 Layer	HoVo DE 6×6 2 Layer
Airplane	346252	674841	392637	769230	415566	812326	432859	843257
Baboon	341373	599241	386262	695117	408236	731742	424544	751410
Elaine	345374	593427	390707	683417	412548	716995	429477	734892
Boat	345287	649330	391277	741300	413895	782065	430811	808887
Lena	346604	677374	393034	772859	415944	815799	433263	847457
Peppers	344814	668460	390648	761519	413361	803383	430403	833896
Average	344951	643779	390761	737240	413258	777052	430226	803300

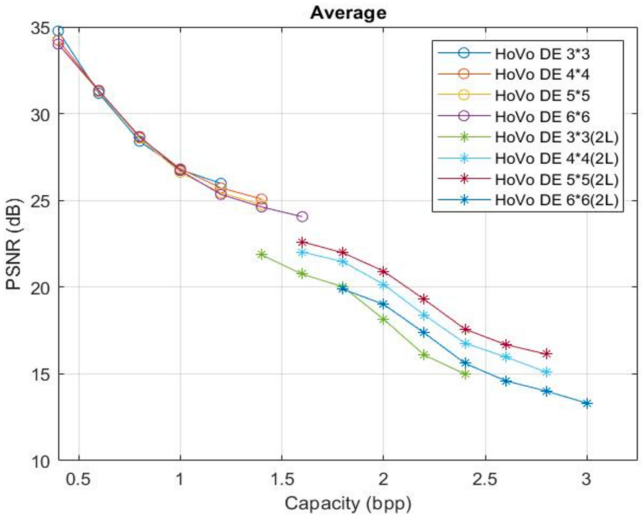


Figure 8: The average image quality vs embedding capacity of the first and second layer embedding based on the DE method, the HoVo_DE embedding strategy is applied to different block division modes

Table 6 shows that the average embedding capacity of HoVo_IRDE_3×3 in a single layer is 346786 bits, while the average embedding capacity of HoVo_IRDE_3×3 using two layers can reach 693332 bits, which is an increase of 346546 bits compared to a single layer embedding; the average embedding capacity of HoVo_IRDE_4×4 in a single layer is 393197 bits, while the average embedding capacity of HoVo_IRDE_4×4 using two layers can reach 786602 bits, which is an increase of 393405 bits compared to a single layer embedding; the average embedding capacity of HoVo_IRDE_5×5 in a single layer is 416154 bits, while the average embedding capacity of HoVo_IRDE_5×5 using two layers can reach 832570 bits, which is an increase of 416416 bits compared to a single layer embedding; the average embedding capacity of HoVo_IRDE_6×6 in a single layer is 433496 bits, while the average embedding capacity of HoVo_IRDE_6×6 using two layers can reach 867025 bits, which is an increase of 433529 bits compared to a single layer embedding. The embedding capacity can be improved significantly compared to the single layer and the PSNR values using two-layer embedding has can attain a level above 30 dB as shown in Figure 9.

Table 6: Comparison of EC of single-layer and two-layer embedding based on HoVo_IRDE method

EC	HoVo DE 3×3 1 Layer	HoVo DE 3×3 2 Layer	HoVo DE 4×4 1 Layer	HoVo DE 4×4 2 Layer	HoVo DE 5×5 1 Layer	HoVo DE 5×5 2 Layer	HoVo DE 6×6 1 Layer	HoVo DE 6×6 2 Layer
Airplane	346804	693605	393217	786434	416154	832180	433473	866974
Baboon	346801	693355	393215	786799	416139	833031	433516	867113
Elaine	346805	693537	393117	786320	416191	832352	433506	866817
Boat	346764	693433	393186	786685	416182	832705	433455	867147
Lena	346801	692906	393207	786678	416161	832458	433501	867288
Peppers	346738	693159	393127	786695	416098	832691	433524	866813
Average	346786	693332	393197	786602	416154	832570	433496	867025

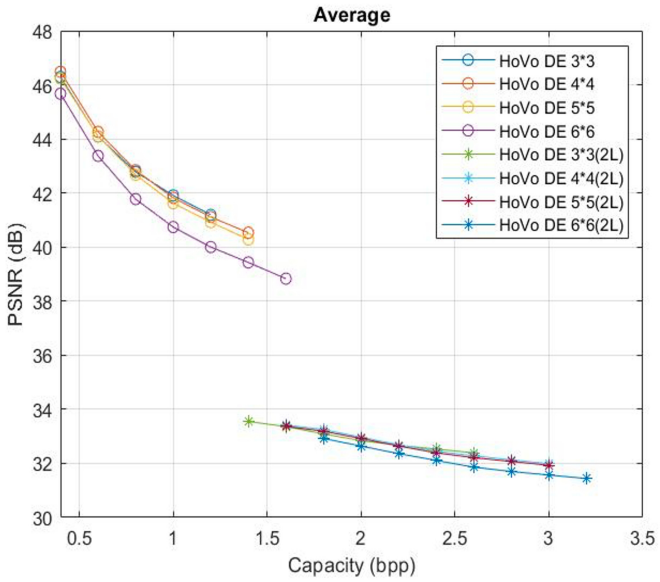


Figure 9: The average image quality vs embedding capacity of the first and second layer embedding based on the IRDE method, the HoVo_DE embedding strategy is applied to different block division modes

5 Conclusions

The traditional DE embedding method is usually in sequential order when it is used to embed secret data; for example, the pixels are embedded individually from left to right. However, with each embedding, the difference continues to expand, causing the difference expansion cascading problem which results in bad embedding capacity. Therefore, when performing the embedding operation, the proposed HoVo_DE embedding strategy abandons the traditional left to right order to perform pixel embedding individually and uses a block as a unit which is then subdivided into groups. Then, the information from the groups at both ends of the block is embedded into the center position group in an overlapping manner. Simultaneously, this strategy checks whether there is an overflow of the camouflaged pixel value each time the secret data are embedded, which can maximize the embedding capacity of secret data and offset the subsequent embedding to a certain extent.

We implemented the HoVo_DE embedding strategy in 3×3, 4×4, 5×5, and 6×6 block division modes. The experimental results indicated that the size of the different blocks increases the embedding capacity and maintains a similar level of image quality. The study also found that the additional embedding capacity decreases each time with a linear curve when larger block size is adopted. This means that the effect of the exchange of the embedding capacity with a larger block-division mode becomes increasingly limited. We also found that when the block size is greater than 5×5, the embedding capacity does not continue to increase significantly, especially when observing the 5×5 and 6×6 blocks. Also, the image quality slows down with the linear curve reduction. Also, experiments have shown that using multi-layer embedding to obtain additional embedding capacity will also face the problem of a sharp drop in image quality. To sum up, the HoVo_DE embedding strategy indeed addresses the difference expansion cascading problem, enhances the embedding capacity, and maintains good image quality.

Acknowledgments

This research was partially supported by the Ministry of Science and Technology, Taiwan, Republic of China under the Grant [MOST 111-2221-E-324-019-MY2]. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the quality.

References

[1] A. M. Alattar, "Reversible watermark using difference expansion of quads," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. iii–377–80, Montreal, QC, Canada, May 2004.

[2] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, pp. 1147–1156, 8 2004.

[3] C. K. Chan and L. M. Cheng, "Hiding data in images by simple lsb substitution," *Pattern Recognition*, vol. 37, pp. 469–474, 3 2004.

[4] M. Hussain, A. W. A. Wahab, Y. I. BinIdris, A. T. S. Ho, and K. H. Jung, "Image steganography in spatial domain: A survey," *Signal Processing: Image Communication*, vol. 65, pp. 46–66, 7 2018.

[5] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, pp. 26–34, 2 1998.

[6] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research," *Neurocomputing*, vol. 335, pp. 299–326, 3 2019.

[7] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. Choo, "A novel difference expansion transform for reversible data embedding," *IEEE Transactions on Information Forensics and Security*, vol. 3, pp. 456–465, 9 2008.

[8] C. F. Lee, H. L. Chen, and H. K. Tso, "Embedding capacity raising in reversible data hiding based on prediction of difference expansion," *The Journal of Systems and Software*, vol. 83, pp. 1864–1872, 10 2010.

[9] C. F. Lee, J. J. Shen, Y. J. Wu, and S. Agrawal, "Reversible data hiding scheme based on difference

expansion using shiftable block strategy for enhancing image fidelity," in *," IEEE 10th International Conference on Awareness Science and Technology (iCAST)*, pp. 1–6, Morioka, Japan, October 2019.

[10] C. F. Lee, C. Y. Weng, C. H. Wang, G. Chakraborty, K. Sakurai, and K. Y. Tsai, "Research on multimedia application on information hiding forensics and cybersecurity," *International Journal of Network Security (IJNS)*, vol. 23, pp. 1093–1107, 11 2021.

[11] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," *Signal Processing*, vol. 93, pp. 198–205, 1 2013.

[12] C. L. Liu, D. C. Lou, and C. C. Lee, "Reversible data embedding using reduced difference expansion," in *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, pp. 433–436, Kaohsiung, Taiwan, November 2007.

[13] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," in *Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS)*, vol. 2, pp. 912–915, Bangkok, Thailand, May 2003.

[14] D. M. Thodi and J. J. Rodriguez, "Reversible watermarking by prediction-error expansion," *6th IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 21–25, 3 2004.

[15] J. Tian, "Image steganography in spatial domain: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 890–896, 8 2003.

[16] D. C. Wu and W. H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, pp. 1613–1626, 2003.

[17] H. Yi, S. Wei, and J. Hou, "Improved reduced difference expansion based reversible data hiding scheme for digital images," in *2009 9th International Conference on Electronic Measurement & Instruments*, pp. 4–315–4–318, Beijing, China, August 2009.

Biography

Chin-Feng Lee received her Ph.D. in Computer Science and Information Engineering from National Chung Cheng University, Taiwan in 1998. She is currently a professor of Information Management at Chaoyang University of Technology, Taiwan. Her research interests include steganography, image processing, information retrieval and data mining.

Jau-Ji Shen received his Ph.D. in Computer Science and Information Engineering from National Taiwan University, Taiwan in 1988. He is currently a professor of Information Management at Chung Hsing University, Taiwan. His research interests include image techniques, data techniques and software engineering.

Chin-Yung Wu received his Master degree in Information Management at Chung Hsing University, Taiwan. His research interests include image techniques, and data hiding.

Ransomware Detection and Prevention through Strategically Hidden Decoy File

Yung-She Lin and Chin-Feng Lee
(Corresponding author: Chin-Feng Lee)

Department of Information Management, Chaoyang University of Technology
Taichung 41349, Taiwan, ROC
Email: asirlin@gmail.com; lcf@cyut.edu.tw

(Received Aug. 15, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)
The Special Issue on Trusted ICT Technologies on the Smart Society and Secure Multimedia Applications
Special Editor: Prof. Chin-Feng Lee (Chaoyang University of Technology)

Abstract

Today's antivirus software has various methods to detect new and unknown malware, offering a very high detection rate and protection ability for virus-type malware. However, this detection rate is significantly reduced or even provides no protection capability for ransomware good at hiding, which is a highly severe threat to the computer files stored by users. Current antivirus software uses machine or deep learning mechanisms to effectively improve the detection rate of new and unknown malware. However, the news still reports ransomware incidents from enterprises or government units. This study implements a honeypot technique, a secret pot mechanism, where decoys are placed in the computer to detect ransomware. The detection program monitors the decoy files used at any time. Once the file is damaged by ransomware, the protection mechanism is triggered immediately, forcing the computer to shut down, preventing the ransomware from encrypting and destroying the files, which can protect the user's files and minimize losses.

Keywords: Cyberattack; Honeypot; Ransomware Detection and Prevention

1 Introduction

One of the required applications for computers are antivirus software, which is used to prevent malicious software (such as computer viruses, trojans, and ransomware) from invading our devices. The advancements of information technology enable hackers to use new technologies and methods that keep pace with the times to develop malicious software or even find weaknesses in antivirus software, break through its detection and protection, and cause computer damage through poison or invasion. This problem is a highly significant information security threat for enterprises.

In 1989, Joseph Popp developed the first ransomware AIDS Trojan in human history [2]. He distributed 20,000 posters labeled "AIDS Information–Introductory Diskettes" to attendees at the World Health Organization (WHO) AIDS Conference. The disks were infected with the AIDS Trojan, which replaces the AUTOEXEC.BAT file and uses it to count computer boot times. Once the boot count reaches 90, the AIDS Trojan hides the directory and encrypts all file names on the hard drive C, making the computer unbootable. Then, a screen appears asking the user to pay ($189 to a PO Box in Panama).

Ransomware has been developed for over 25 years, and the encryption technology used by ransomware is also constantly improving. Today's ransomware uses asymmetric encryption technology that is difficult to crack.

In addition to stealing the confidential information in the user's computer, ransomware also encrypts the computer files such that the user cannot obtain access. Then, the user is asked to pay a certain amount of Canadian currency as a ransom to obtain the decryption key. If the user fails to pay the ransom according to the hacker's instructions, the latter does not provide the decryption key to the user. Without the correct decryption key, the file cannot be restored. In several cases, even when the user has paid the ransom according to the hacker's instructions, the hacker still did not provide the decryption key. Cases of repeated extortion by the same ransomware have also occurred.

Today's antivirus software has good detection rate and protection against known ransomware, but most are almost powerless against new and unknown ransomware. Antivirus software can become useless because ransomware has multiple ways to evade detection. This is also the reason why users install antivirus software and often update the latest virus patterns, but they are still unable to escape malicious ransomware. Four types of ransomware incidents have been reported:

1) In March 2021, Acer was attacked by ransomware,

which stole company files and encrypted them. Hackers extorted $50 million from Acer [5]. In July 2021, Kaseya Software suffered a ransomware attack during the National Day holiday in the United States, which affected about 1,500 companies around the world, compromised more than 1 million computers, and was extorted $70 million [6]. Occurred in August 2021 Gigabyte was attacked by hackers, it was confirmed that confidential files were stolen, and the hackers demanded a ransom in cryptocurrency [7].

2) Cause network interruption and leakage of customer data. In 2021, Insurance giant CNA reports data breach after ransomware attack [8], and Fimmick ransomware attack puts over 35,000 people's data at risk [11].

3) Attack cloud infrastructure. In 2021, Python ransomware script targets ESXi server for encryption [13]. In 2022 ASUSTOR NAS been hit by Deadbolt Ransomware [14].

4) Create national security problems such as in 2021, the Colonial Pipeline cyberattack [15] shuts down pipeline that supply 45% of East Coast's fuel.

In recent years, due to its great progress, artificial intelligence-related technologies, such as machine or deep learning, have been used to effectively improve the detection rate of new computer viruses and malware. According to Poudyal and Dasgupta, the detection rate of ransomware can reach 99.54% by using artificial intelligence-related technologies [17]. Adamov and Carlsson applied a Reinforcement Learning approach to anti-ransomware testing, and helps to improve weaknesses in anti-ransomware defenses and fixes them before a real attack occurs [19].

A deception strategy commonly used to detect network intrusion is the honeypot mechanism, which lures hackers or malicious software to attack the honeypot server. This method has shown good performance and can effectively reduce or prevent the server from being attacked. Pascariu and Barbu use Honeypot solution designed to detect a ransomware infection identify the ransomware family [9]. Moore [3] deployed a Honeypot server on the network to detect any ransomware activity.

In the present study, we refer to the deception strategy of the honeypot mechanism commonly used in network intrusion detection. According to the characteristics of the Windows operating system, a decoy file is planted to detect the ransomware. Once the honeypot file used as bait has been damaged by ransomware, it shuts down the computer immediately to prevent the ransomware from encrypting and destroying files, which can protect the user's data to the greatest extent and minimize losses.

This paper is divided into five parts. The first section briefly introduces the research background and the current ransomware threats. The second section mainly discusses the honeypot mechanism used to lure the enemy. The third section presents how we strategically use a hidden decoy file to detect and prevent the ransomware attack. The fourth section describes the results and the comparison of the protection effect of antivirus software commonly used by users against new and unknown ransomware. The fifth section presents the conclusion.

2 Literature Review

Honeypot is a deception strategy mechanism used to lure attackers. The purpose of honeypot is not to prevent or mitigate attacks, but rather to pretend to be a real environment, deceive attackers, and lure them into exhibiting aggressive behavior. When the attacker appears, they are caught and dealt with later.

A good example of a honeypot is proposed by Pascariu and Barbu, who used Raspberry Pi to disguise an SMB server secret jar to lure and catch ransomware attacks on the Internet. This method has a good effect when the ransomware is intended to destroy the files on the server in the network [9]. Moore deployed a honeypot server by creating a secret can folder on the server and monitoring the activities related to the files to detect any ransomware activity on the network [3]. Venkatesh *et al.* established a sealed container environment for file servers to test and analyze any malicious network behavior or file destruction [1]. However, the methods of detecting ransomware by the encrypted server deployed on the Internet cannot identify when the ransomware starts to destroy the files. The sabotage of servers by ransomware is not detected until the files on the secret can are encrypted.

Zhuravchak *et al.* used an file symbolic linking honeypots to detect and prevent ransomware attacks in Linux operating systems [4]. A honeypot archive is deployed to monitor ransomware encryption activity on the file, and deploying one or more symbolic links pointing to the sealed and monitored can file in the folder. Thus, the file encryption activities of the ransomware on the Linux operating system can be completely and effectively monitored.

Fan *et al.* deployed a high-efficiency canister system to block various attacks from the network [12]. According to different network security requirements, Eliot *et al.* proposed a flexible network security laboratory environment that uses Raspberry Pi and VMWare virtual machines as honeypot deployment to effectively detect and prevent intrusions and attacks from the Internet [10]. Fan et al mentioned an intrusion prevention system (IPS) integrated with a honeypot can used to detect and block attacks from internal or external networks [16]. Lee *et al.* [18] suggested that when a malicious program has successfully invaded a computer system and gotten administrator privileges, the hidden interface is used to implement file-based Phantom FS spoofing technology to provide simulation and camouflage and hide real files, maximizing the chance of successfully deceiving malware into corrupt file.

The above cases show that the honeypot mechanism

used to deceive and lure hackers or malicious software to attack itself has various setting methods according to different baiting requirements. The secret can server has shown good protection on the server and the network.

3 Proposed Method

With a focus on the behavior pattern of ransomware encrypting files, this study refers to the method of deceiving the enemy that is commonly used in network intrusion detection. The encrypted canister file is set as a bait in the computer with the Windows operating system to attract and deceive ransomware. Once the file is damaged by ransomware, the protection mechanism is triggered immediately, forcing the computer to shut down and preventing the ransomware from encrypting and destroying files, which can protect the user's data and minimize losses.

3.1 Behavior Patterns of Ransomware

This study has repeatedly verified the behavior of several kinds of ransomware on file encryption and has confirmed its behavior mode. The ransomware found in the Windows system, when encrypting the files of the compromised computer, starts looking for the files from the root directory of drive C: to encrypt and destroy. Then, the ransomware repeatedly searches for the files in its lower subfolders. In encrypting and destroying files, ransomware uses an ascending order by file name. Ransomware does not destroy the normal operations of the computer system, and does not encrypt executable files including exe, com, dll, and sys. Rather, the main targets for encryption are the following file types:

1) Document File: txt, doc, docx, pdf, xls, xlsx, ppt, pptx, htm, html, ...
2) Video files: mp3, wav, mp4, dat, avi, ...
3) Program source code files: c, cpp, java, js, css, ...
4) Graphic files: bmp, jpg, gif, dwg, ...
5) Database files: dbf, mdf, mdb, accdb, db, sql, xml, json, ...

This study finds that the ransomware may process the file names of the encrypted files in the following three modes: (1) The file name remains unchanged after encryption; (2) After encryption, a specific file name is added to the file name, varying with different ransomware. For example, assuming that the original file name is A.txt, the encrypted file name becoms A.txt.xxx; (3) After encryption, the file name is changed to a random English + numeric string file name. Suppose the original file name is A.txt, the encrypted file name may become Asdcifksmc.xk354.

Given the difficulties to obtain new and unknown ransomware, all of the obtained ransomware are known and can be effectively identified and protected by antivirus software. However, the protective effect on new and unknown ransomware is impossible to measure. To simulate the new and unknown ransomware, we refer to the behavior patterns of three different types of ransomwares and develop three simulation programs to simulate new and unknown ransomware. Then, these three simulation programs are used to test the proposed method and the common antivirus software in the market for purposes of comparison. When facing new unknown ransomware, the proposed method and each antivirus software can effectively protect computer files.

The ransomware processing for encrypting files is a complex and hard-to-break asymmetric encryption algorithm. In this study, the ransomware simulation program only mimics the effect of encrypting files, and does not use a high-strength asymmetric encryption algorithm. Instead, a simple symmetric encryption algorithm is used to simulate the effect of ransomware encrypting and breaking files. Table 1 shows the software and hardware of the development environment where Microsoft Visual Studio 2019 and Microsoft Visual C# are used to develop a simulated ransomware.

Table 1: The software and hardware configuration of development environment

CPU	Intel i7-8750H
RAM	DDR4 16GB
HDD	SSD 256GB
Operating System	Microsoft Windows 10 Home
Development Tools	Microsoft Visual Studio 2019
Programming Language	Microsoft Visual C#

3.2 The Protection Mechanism

This study addresses the behavior patterns of ransomware encrypted files, with reference to the honeypot method of deceiving the enemy commonly used in the network intrusion detection. The proposed method proposed sets up a monitoring program in Windows OS and a honeypot file in the computer as bait to lure and deceive ransomware attacks. If the monitoring program finds that the honeypot file used as bait is encrypted by ransomware, it triggers emergency protection, forcing the computer to shut down and stop working, to prevent the ransomware from encrypting other files and thereby protect the data. Figure 1 shows the proposed protection architecture diagram.

Figure 2 shows the proposed protection workflow chart. The monitoring program first checks whether the bait honeypot file exists, given that ransomware may change the file name synchronously during encryption. If the monitoring program does not find the bait honeypot file, then it assumes that the ransomware has invaded the computer, changed the file name of the decoy file, and has

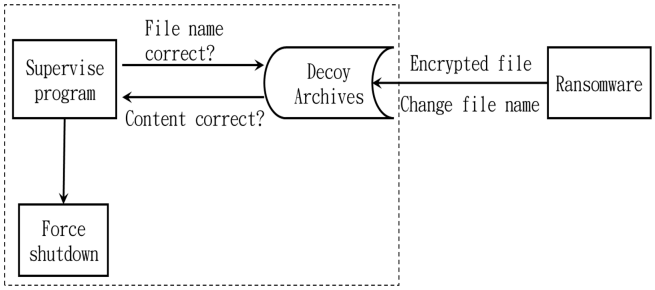


Figure 1: The proposed protection architecture diagram.

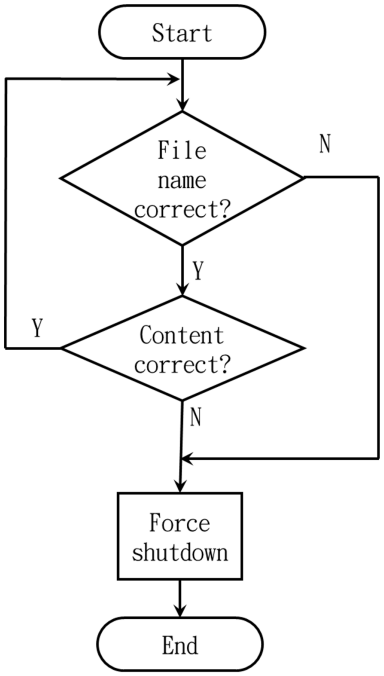


Figure 2: The proposed protection workflow chart.

begun to destroy files. At this point, the monitoring program triggers the protection mechanism to force the computer to shut down, and thus that the ransomware cannot work, preventing the continuous destruction of files. Thus, To further determine if there is any ransomware intrusion, the monitoring program opens the honeypot file and reads its content for comparison. If the read content is the correct preset identification key, then the computer has not been corrupted by ransomware and remains safe at this time. To avoid occupying system resources, the monitoring program enters the sleep mode (60 seconds by default), and restarts the detection after resting.

3.3 The Testing Method

This study adopts the ransomware behavior model to implement new and unknown simulated ransomware to test the proposed methods and antivirus software regarding the detection and protection capabilities of

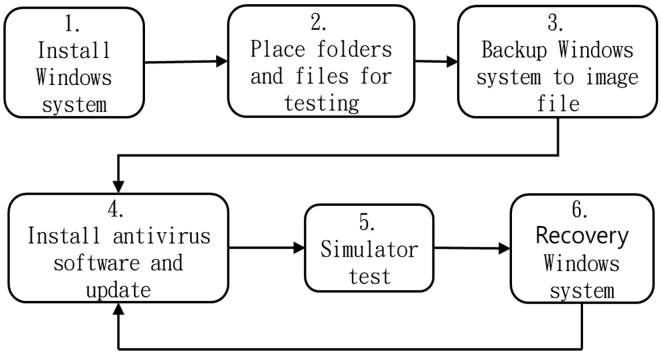


Figure 3: The flowchart of this test assignment.

new and unknown ransomware.
A desktop computer is used for testing and verification. Table 2 shows the software and hardware specifications of the test computer.

Table 2: The software and hardware specifications of the test computer.

CPU	AMD Ryzen R7-3700X
RAM	DDR4 16GB
Motherboard	ASUS B450M-A
HDD	SSD 256GB
Operating System	Microsoft Windows 10 Home
Development Tools	Microsoft Visual Studio 2019

The test proceeds as follows: Step 1 installs Windows 10. Step 2 simulates general users storing folders and files in the disk drive, with a total of 850 files. Step 3 backs up the hard disk as an image file. Step 4 reboots and installs the antivirus software for testing, and updates the antivirus software version and virus pattern to the latest version. Step 5 executes simulated ransomware to test the antivirus software the detection and protection capabilities. Step 6 restores the image file to this hard disk and returns to step 4 for the next test. Figure 3 shows the flowchart of this test assignment.

We prepared commonly used 10 files, including document files (.doc, .docx, xlsx, ods, ppt, pdf, odt, txt), video files (.mp4), compression files (.zip), as shown in Table 3. In Table 4, we create 8 different folders, each of which contains a different number of files, for a total of 850 files.

4 Research Results

Facing the ever-evolving ransomware, ensuring that the computer files are not damaged is a problem for enterprises and government units. Although everyone knows that file backup is the best method to deal with ransomware, such operation is often not performed due to human negligence, equipment problems, or network is-

Table 3: 10 different file type.

Item	File	File type
1	01.jpg	Image file
2	02.mp4	Video file
3	03.dot	Document file
4	04.doc	Document file
5	05.xlsx	Spreadsheet file
6	06.ods	Spreadsheet file
7	07.ppt	Presentation file
8	08.pdf	Document file
9	09.txt	Plain text file
10	10.zip	Compressed file

Table 4: 8 different folders, each of which contains a different number of files, for a total of 850 files.

Item	Folder	File count
1	C:\	10
2	C:\TEST	120
3	C:\Users\User\Desktop	120
4	C:\Users\User\Documents	120
5	C:\Users\User\Download	120
6	C:\Users\User\Music	120
7	C:\Users\User\Pictures	120
8	C:\Users\User\Videos	120
Total files		850

Figure 4: Screenshot of the original content of “a.txt” file.

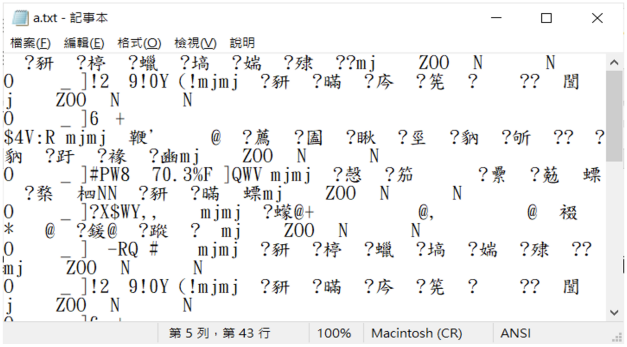


Figure 5: Screenshot of encrypted content of a.txt

sues. This lack of backup is the main cause of huge losses for businesses when faced with ransomware attacks.

Antivirus software has a good protection effect against general computer viruses, but can it effectively protect computer files from new and unknown ransomware? This issue deserves attention. Therefore, well-known antivirus software on the market are selected and compared to test the detection and defense capabilities of each method against various ransomware attacks.

This study prevents the continuous destruction of archives from three different types of ransomware threats. The protection rate of stored files is 98.82%. If the user does not place the file in C:\, then the protection rate can reach 100%, which is an excellent performance.

4.1 Simulate and Verify the Behavior of Ransomware

The behavior patterns of three different ransomware types are imitated in simulations A, B, and C to represent three new and unknown ransomwares for testing purposes. In addition, we create a plain text file named ”a.txt” to illustrate the file changes after going through different types of ransomware emulators. Figure 4. Screenshot of the original content of “a.txt” file.

Simulation A does NOT change file names or extensions but only encrypts the file contents. Figure 5 presents the encrypted content of “a.txt” file. After encryption,

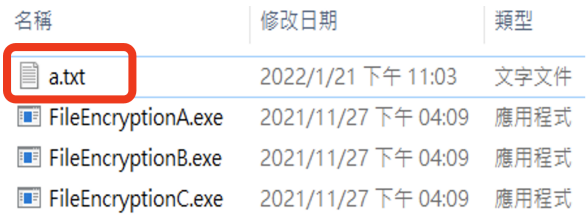


Figure 6: Screenshot of the file list after simulation program A is executed

Figure 7: Screenshot of the file list after simulation program B is executed

the file name has not been changed. Figure 6 shows the screenshot of the file list after simulation A. In simulation B, which encrypts files and appends the encrypted filename with ".xxx." Thus, the "a.txt" file is be renamed as "a.txt.xxx," as shown in Figure 7. Simulation C not only encrypts file contents but also renames file name by using random English + numeric strings, as shown in Figure 8.

4.2 Method to Prevent Ransomware from Continuously Destroying Files and Their Protective Effects

We proposes a method to prevent ransomware from continuously destroying files. Table 5 shows the results of attack tests from three different types of ransomware simulators. The numbers represent the files that were attacked, with 0 indicating that no files in that folder were attacked by the ransomware simulation. Faced with three different types of ransomware threats, the proposed method garners a protection rate of 98.82% for the computer files. If the user does not place the file in C:\, then the protection rate can reach 100%, which is an excellent performance.

Figure 8: Screenshot of the file list after the execution of the Type C emulator

Table 5: The results of attack tests from three different types of ransomware simulators.

Folder	A simulation program	B simulation program	C simulation program
C:\	10	10	10
C:\TEST	0	0	0
C:\Users\User\Desktop	0	0	0
C:\Users\User\Documents	0	0	0
C:\Users\User\Download	0	0	0
C:\Users\User\Music	0	0	0
C:\Users\User\Pictures	0	0	0
C:\Users\User\Videos	0	0	0
Total damage	10	10	10
Total not destroyed	840	840	840
Destruction rate	1.18%	1.18%	1.18%
Protection rate	98.82%	98.82%	98.82%
Stop time	< 60 second	< 60 second	< 60 second
	Force shutdown	Force shutdown	Force shutdown

4.3 Comparison Between the Proposed Method and Other Well-known Antivirus Software for Ransomware Protection

Antivirus software has a good protection effect against general computer viruses, but can it effectively protect computer files in the face of new and unknown ransomware? This issue deserves attention. Therefore, this study selects 16 commonly used sets of antivirus software in the market, as shown in the list of antivirus software to be tested in Table 6. All antivirus software is tested using the most common installation defaults for general users. Among them, the four sets of antivirus software (Avast Free Antivirus, Comodo Free Antivirus, Microsoft Defender, and Tinder Security) have anti-ransomware or advanced protection functions in their setting items, which need to be manually turned on. This study also manually starts the following settings and tests for these four sets of antivirus software, as follows: Avast Free Antivirus opens the default protected folder; Comodo Free Antivirus opens the Container; Microsoft Defender turns on controlled folder access; and Tinder Security - Turn on ransomware trapping.

Table 6: Antivirus software to be tested.

Item	Antivirus software
1	Avast Free Antivirus
2	AVG Antivirus Free
3	Avira Antivirus
4	Bitdefender
5	BullGuard Antivirus
6	Comodo Free Antivirus
7	F-Secure Safe
8	Kasperskey Free
9	McAfee Total Protection
10	Microsoft Defender
11	Panda Free Antivirus
12	PC-Cillin 2022
13	Vipre Advanced Security
14	Kinstnui
15	360 Total Security
16	Sysdiag

In the proposed method, 16 sets of antivirus software presets and four sets of antivirus software are set to manually open ransomware or advanced protection mode. Table 7 shows the test results of three different types of new and unknown ransomware simulation, and compares the protection rates of the proposed method and other well-known antivirus software.

In the results, 0% indicates that when the antivirus software faces ransomware emulators, all test files are encrypted and have no protection at all. According to the test results, 12 sets of antivirus software had no protection effect when faced with three different types of ransomware simulation. A set of antivirus software achieves no protection even when the advanced protection setting function is manually turned on. However, a different set of antivirus software shows a good protection effect against the attacks in simulations A or B, but has no protection against the attack in simulation C. Another set of antivirus software manually activates the ransomware trapping function, which has a good protection effect from attacks in simulations B and C, but has no protection against the attack in simulation A.

Meanwhile, the "Comodo Free Antivirus" software can completely block the attacks from three different types of ransomware simulations after manually enabling the Container function, yielding an impressive 100% protection rate for files. The proposed method and two other sets of antivirus software, Bitdefender and BullGuard Antivirus, achieve a protection rate of 98.82% when faced with the attacks from the three different types of ransomware simulation. Despite a very small number of test files loss, the abovementioned antivirus software also shows a good protection level.

Given the many antivirus software with no protection for the three different types of simulations, they cannot

Table 7: The test results of three different types of new and unknown ransomware simulation.

Item	Antivirus software	A simulation program	B simulation program	C simulation program
1	The proposed method	98.82%	98.82%	98.82%
2	Avast Free Antivirus default value	0%	0%	0%
3	Avast Free Antivirus enable default protected folder	0%	0%	0%
4	AVG Antivirus Free	0%	0%	0%
5	Avira Antivirus	0%	0%	0%
6	Bitdefender	98.82%	98.82%	98.82%
7	BullGuard Antivirus	98.82%	98.82%	98.82%
8	Comodo Free Antivirus default value	0%	0%	0%
9	Comodo Free Antivirus enable Container	100%	100%	100%
10	F-Secure Safe	0%	0%	0%
11	Kasperskey Free	0%	0%	0%
12	McAfee Total Protection	0%	0%	0%
13	Microsoft Defender	0%	0%	0%
14	Microsoft Defender enable Controlled Folder Access	56.47%	56.47%	56.47%
15	Panda Free Antivirus	0%	0%	0%
16	PC-Cillin 2022	70.59%	70.59%	70.59%
17	Vipre Advanced Security	98.82%	98.82%	0%
18	Kinstnui	0%	0%	0%
19	360 Total Security	0%	0%	0%
20	Sysdiag default value	0%	0%	0%
21	Sysdiag enable Ransomware lure	0%	98.82%	98.82%

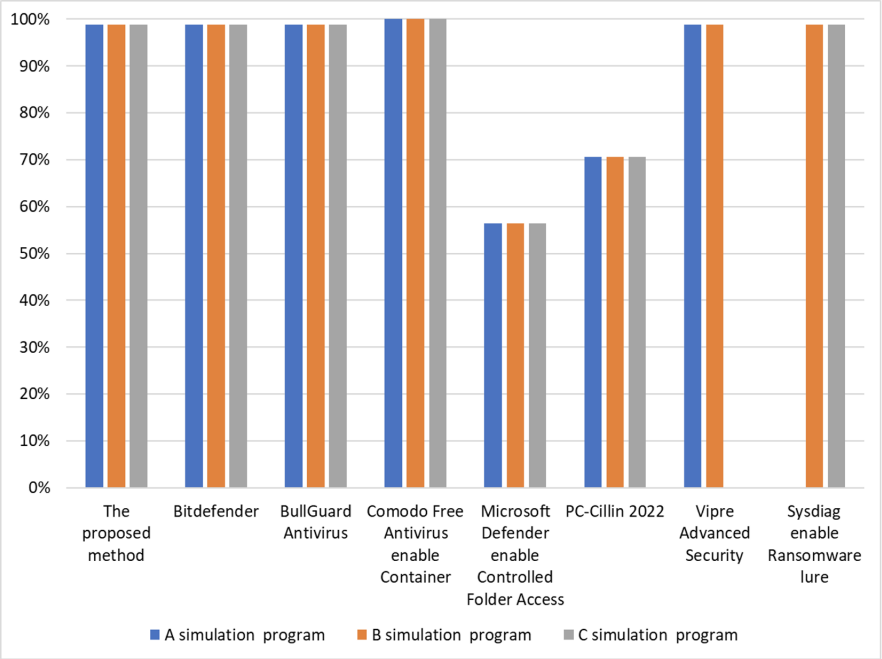


Figure 9: Comparison chart of the proposed method and other well-known antivirus software against ransomware protection (protection rate)

be clearly displayed in the comparison chart, and are thus removed. Figure 9 clearly shows the protection effect of antivirus software on new ransomware. The proposed method is comparable to well-known antivirus software in terms of protection against new and unknown ransomware.

The results of test experiments in Figure 9 present that the proposed method can effectively monitor and prevent ransomware from encrypting files. When faced with the threat of three different types of ransomware such as simulated programs A, B, and C, the method we proposed can effectively protect computer files, and the protection rate is 98.82% which is same with the methods of Bitdefender and BullGuard Antivirus. The Comodo Free Antivirus enable Container has the highest protection rate 0f 100%. However, if the files are not placed on the disk drive C:\, then this proposed method can detect ransomware and protect the files in time with the protection rate of 100%.

5 Conclusions

The results of the test experiments prove that the proposed method can effectively monitor and prevent file encryption from ransomware. When facing the threat of three different types of ransomwares, the proposed method can protect the computer files with a rate of 98.82%. If the files are not placed in C:\, then the protection rate can reach 100%, which shows excellent performance.

The test also reveals that when the antivirus software commonly used by users face new and unknown ransomware, only a few antivirus software have protective effects. Thus, the computer clearly has antivirus software installed, so why is it still infected with ransomware or computer virus? The current testing is limited to new and unknown ransomware threats, and thus may not be fair to antivirus software, which may have other good protection effects.

The proposed method can indeed protect computer files from the threat of ransomware. Thus, this method has high value, especially for users and enterprises with no file backup. This study focuses on the behavior pattern of ransomware as a method to detect and prevent it from encrypting and destroying files. The effect is the same for new and unknown ransomware. However, the proposed method does not have the functions of antivirus software and cannot replace its protection. The purpose of this study is to strengthen the existing antivirus software that cannot effectively detect and prevent computer files from being encrypted and damaged by new or unknown ransomware. The proposed method can work together with antivirus software to enable better protection effect for users' computer files.

Acknowledgments

This research was partially supported by the Ministry of Science and Technology, Taiwan, Republic of China under the Grant[MOST 111-2221-E-324-019-MY2]. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the quality.

References

[1] L. Abrams, "Computer giant acer hit by $50 million ransomware attack," Jan. 28, 2023. (https://www.bleepingcomputer.com/news/security/computer-giant-acer-hit-by-50-million-ransomware-attack)

[2] A. Adamov, A. Carlsson, "Reinforcement learning for anti-ransomware testing," in *Proceedings of 2020 IEEE East-West Design & Test Symposium (EWDTS'20)*, pp. 1–5, Sept. 2020.

[3] A. Brandt, "Python ransomware script targets esxi server for encryption," Jan. 28, 2023. (https://news.sophos.com/en-us/2021/10/05/python-ransomware-script-targets-esxi-server-for-encryption)

[4] M. Clark, "Hackers reportedly threaten to leak data from gigabyte ransomware attack," Jan. 28, 2023. (https://www.theverge.com/2021/8/9/22616882/gigabyte-technologies-ransomware-attack-data-leak-112-gb-ransomexx)

[5] L. Dignan, "Colonial pipeline cyberattack shuts down pipeline that supplies 45% of east coast's fuel," Jan. 28, 2023. (https://www.zdnet.com/article/colonial-pipeline-cyberattack-shuts-down-pipeline-that-supplies-45-of-east-coasts-fuel)

[6] N. Eliot, D. Kendall, and M. Brockway, "A flexible laboratory environment supporting honeypot deployment for teaching real-world cybersecurity skills," *IEEE Access*, vol. 6, pp. 34884–34895, 2018.

[7] W. Fan, Z. Du, D. Fernández, and V. A. Villagrá, "Enabling an anatomic view to investigate honeypot systems: A survey," *IEEE Systems Journal*, vol. 12, pp. 3906–3919, 2018.

[8] W. Fan, Z. Du, M. Smith-Creasey, and D. Fernández, "Honeydoc: An efficient honeypot architecture enabling all-round design," *IEEE Journal on Selected Areas in Communications*, vol. 37, pp. 683–697, 2019.

[9] ASUSTOR Community Forum, "Deadbolt ransomware," Jan. 28, 2023. (https://forum.asustor.com/viewtopic.php?f=45&t=12630)

[10] S. Gatlan, "Insurance giant cna reports data breach after ransomware attack," Jan. 28, 2023. (https://www.bleepingcomputer.com/news/security/insurance-giant-cna-reports-data-breach-after-ransomware-attack)

[11] J. Lee, J. Choi, G. Lee, S. W. Shim, and T. Kim, "Phantomfs: File-based deception technology for thwarting malicious users," *IEEE Access*, vol. 8, pp. 32203–32214, 2020.

[12] R. McMillan, "Ransomware hackers demand $70 million to unlock computers in widespread attack," Jan. 28, 2023. (https://www.wsj.com/articles/ransomware-hackers-demand-70-million-to-unlock-computer-in-widespread-attack-11625524076)

[13] C. Moore, "Reinforcement learning for anti-ransomware testing," in *Proceedings of 2016 Cybersecurity and Cyberforensics Conference(CCC'16)*, pp. 77–81, 2016.

[14] C. Pascariu, I. D. Barbu, "Ransomware honeypot - honeypot solution designed to detect a ransomware infection identify the ransomware family," in *Proceedings of the 11th International Conference on ELECTRONICS, COMPUTERS and ARTIFICIAL INTELLIGENCE (ECAI'19)*, pp. 1–4, 2019.

[15] S. Poudyal, D. Dasgupta, "Analysis of crypto-ransomware using ml-based multi-level profiling," *IEEE Access*, vol. 9, pp. 122532–122547, 2021.

[16] The Standard, "Fimmick ransomware attack puts over 35,000 people's data at risk," Jan. 28, 2023. (https://www.thestandard.com.hk/breaking-news/section/4/181793/Fimmick-ransomware-attack-puts-over-35,000-people%27s-data-at-risk)

[17] J. Venkatesh, V. Vetriselvi, Ranjani Parthasarathi, and G. Subrahmanya V.R.K. Rao, "Identification and isolation of crypto ransomware using honeypot," in *Proceedings of Fourteenth International Conference on Information Processing (ICINPRO'18)*, pp. 1–6, 2018.

[18] Wikipedia, "Aids (trojan horse)," Jan. 28, 2023. (https://en.wikipedia.org/wiki/AIDS_(Trojan_horse)

[19] D. Zhuravchak, T. Ustyianovych, V. Dudykevych, B. Vennyk, and K. Ruda, "Ransomware prevention system design based on file symbolic linking honeypots," in *Proceedings of 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'21)*, vol. 1, pp. 284–287, 2021.

Biography

Yung-She Lin received his Master degree in Information Management from Chaoyang University of Technology, Taiwan. Currently He is a PhD. Candidate in the department of Information Management, Chaoyang University of Technology. His research interests include cryptography and information hiding.

Chin-Feng Lee received her Ph.D. in Computer Science and Information Engineering from National Chung Cheng University, Taiwan in 1998. She is currently a professor of Information Management at Chaoyang University of Technology, Taiwan. Her research interests include steganography, image processing, information retrieval and data mining.

High Embedding Capacity Data Hiding Technique Based on Hybrid AMBTC and LSB Substitutions

Pei-Chun Lai[1], Jau-Ji Shen[1], and Yung-Chen Chou[2]
(Corresponding author: Yung-Chen Chou)
Department of Management Information Systems, National Chung Hsing University[1]
145 Xingda Rd., South Dist., Taichung City 40227, Taiwan (R.O.C.)
iSchool, Feng Chia University[2]
No. 100, Wenhua Rd. Xitun Dist., Taichung City 407102, Taiwan (R.O.C.)
Email: yungchen@gmail.come
(Received Aug. 15, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)
The Special Issue on Trusted ICT Technologies on the Smart Society and Secure Multimedia Applications
Special Editor: Prof. Chin-Feng Lee (Chaoyang University of Technology)

Abstract

How to securely send secret data to receivers throughout network transmission has been a critical research topic in recent years. Data hiding, or steganography, is a security method that uses a cover media to carry the secret message to achieve the goal of secure data transmission. In general, there are three classifications of data-hiding methods: spatial domain, frequency domain, and compression domain. In this paper, we offer a new compression domain-based steganography to increase the embedding capacity of current AMBTC-based data-hiding methods. In AMBTC, the low and high quantized values are generated with a bitmap to create the compression code of an image. Our proposed method considers the complexity of block contents to design more adaptive embedding strategies that can embed as much secret data as possible. We utilize a predefined threshold Thr to determine the image block type. Based on our simulations, 16 is the most suitable threshold level that enables the proposed method to achieve a higher embedding capacity with the good visual quality of the restored images. From the experimental results, the proposed method has more embedding capacity than other AMBTC-based data hiding methods and can maintain an excellent visual quality of the restored images.

Keywords: *Absolute Moment Block Truncation Coding (AMBTC); Data Hiding; Irreversible Data Hiding (IDH); QVD; Modified LSB Substitution*

1 Introduction

Information technologies and the Internet have provided the conveniences of our modern life. Nowadays, we can send documents, images, or messages to one another in an instance. However, the Internet is a double-edged sword. Data in transition on the computer network is always at risk of been stolen by malicious or unauthorized users. For data protection, we have used various data encryption algorithms (for example, AES and RSA) to hide a secret message as cipher datagram, which makes the data look like random noise. Data encryption algorithms are used in various applications like wireless network communication and secret data storing, etc. Using data encryption algorithms to protect data from malicious users is an important research topic.

Data hiding, or steganography, is a security technique that protects data transmitting over the computer network. Key to data hiding methods is the cover media, which can be a text file, image, video, audio, or web page, etc. A cover media must contain redundant space and provide imperceptibility to humans. For example, an image is composed of multiple pixel values some of which can be made redundant, and minute changes in image clarity can be hard to detect for the human eye. Based on these characteristics, an image is one of the most useful cover media for secret data embedding for secure data transmission.

In general, there are three main types of data hiding methods utilizing digital images as the cover media: Spatial Domain, Transform Domain, and Compression Domain [11]. Embedding capacity and stego-image quality are two of the most important factors used for evaluating the performance a data hiding method. In spatial

domain data hiding techniques (such as Pixels-Value Difference (PVD) [32], Least Significant Bit Substitution [1], and Difference Expansion (DE) [31]), researchers focus on modifying image pixel values to embed as much secret message as possible while still maintaining a good stego-image visual quality.

In transform domain data hiding methods, improving stego-image visual quality takes precedence over data embedding capacity. Transforming digital images from the spatial domain to the frequency domain is frequently used in image processing operations. After transforming images to the frequency domain, the coefficient matrix is calculated based on three frequency bands (high frequency band, middle frequency band, and low frequency band). Low frequency band coefficients preserve the basic information of the image content. Mid-frequency band coefficients store a portion of the image details while high frequency band coefficients contain most of the image details. Coefficients in the high frequency band can be treated as redundant data. In other words, removing high frequency band coefficients will not significantly affect the visual quality of an image observed by the human eye. Considering the effect on a stego-image due to common image processing operations (for example, image compression), stego-images generated by transform domain data hiding methods are more robust than spatial domain data hiding methods. Discrete Cosine Transform (DCT), and Discrete Fourier Transform (DFT) are the two popular transformation functions for converting from the spatial domain to the frequency domain.

There are two types of data hiding methods that concern cover image completeness after data extraction: Reversible data hiding (RDH) and Irreversible data hiding (IDH) [9]. In the IDH data hiding method, the original image cannot be restored from the stego-image after data extraction. Compared with RDH, IDH provides more data embedding capacity since fewer data bits are needed to preserve the information of the original cover image. Two popular IDH methods are Pixel Value Differencing (PVD) [32] and Least Significant Bit (LSB) substitution. RDH methods consider both data embedding capacity and quality of the original image restored from the stego-image after data extraction. In remote medical care or military image applications, the cover image usually contains information that is important to the receiver. This means that no distortion of any kind is allowed in the restored cover image. Two well-known RDH methods are Histogram Shifting (HS) [22] and Difference Expansion (DE) [32].

In 2003, Wu and Tsai proposed Pixel Value Differencing (PVD) [32] steganography, an irreversible information hiding method. This method treats two adjacent pixels as a block, divides the image into multiple sets of non-overlapping blocks (1×2 a block), and then calculates the pixel difference value of the block. A larger error value indicates that the block is closer to the image edge or the complex area. Conversely, a smaller error value means that the block is near a smooth area. This method utilizes the amount of storage that can be tolerated by the change between the pixel value differences in the image to hide the secret data. Therefore, changes in the image after data embedding are difficult for the human eye to detect and the goal of low-distortion images can be achieved.

Furthermore, PVD steganography is simple and fast, enabling more applications and making it easier for further research [3, 5, 10, 12, 23, 26, 27, 30, 34]. For example, Wu [33] combined PVD and LSB substitution methods to achieve high embedding capacity for each adjacent pixel. Liao's method [16] divides the image into non-overlapping blocks each sized 2×2 and calculates the average difference of each block value. According to the threshold value, blocks are divided into smooth blocks and complex blocks. Smooth blocks use fewer LSB bits to embed the secret data while complex blocks use more LSB bits. This method improves the embedding capabilities. Swain's method [29] divides the image into non-overlapping blocks sized 3×3. For each block the average differences of each block value are calculated. Blocks are then classified into four ranges used to embed 2 to 5 secret bits, respectively. This method not only increases the embedding capacity but also improves the visual quality of the stego image.

Absolute Moment Block Truncation Coding (AMBTC) [14] is a high compression rate image compression method. According to the design of AMBTC, pixels in a block are separated into two groups using the pixel mean value of the block as the threshold. The compression code of a block is generated based on the block bitmap and two quantized values. AMBTC is easy to implement and provides high compression performance and a good visual quality of decompressed images. Based on these qualities, some researchers use AMBTC to develop new data hiding methods for secret data transmission [2, 4, 6, 15, 17–21, 28, 35].

Ou *et al.* [25] presented a data hiding method utilizing the property of AMBTC's quantized values for blocks to detect whether a block contains complex or smooth content. For smooth blocks, secret data is embedded into the block bitmap directory; otherwise, the quantized values of a block are used to carry one secret data only. Obviously, a complex content image block can only be used to carry one secret bit, which can impact the performance of data embedding capacity. However, Ou *et al.*'s method significantly improves the visual quality of decompressed images.

In 2016, Huang *et al.* proposed a hybrid AMBTC data hiding method [8] to improve both data embedding capacity and image visual quality compared to Ou *et al.*'s method. The main idea of Huang's method is to conceal the secret data into the difference between two quantized values of a block. In 2019, Kumar *et al.* presented an AMBTC based data hiding method [13] by including PVD and Hamming distance techniques. Kumar *et al.*'s method provides a rule to utilize the difference between two quantized values of a block to classify the block into one of three types: smooth block, low complex block, and

high complex block. Then, PVD is adopted to hide the secret data into the difference value of two quantized values of a block. According to their design, data embedding capacity is improved.

In 2020, Horng *et al.* presented Quotient Value Differencing (QVD) that includes LSB substitution to enhance the PVD method for concealing more secret bits into the difference value of two quantized values of a block. Horng *et al.*'s method successfully improves the data embedding capacity and maintains acceptable visual quality of stego-images.

Irreversible data hiding techniques has higher data embedding capacity than reversible data hiding (IDH) techniques. However, there is an inverse relationship with the visual quality of stego-images. This means that the better the visual quality of a stego-image, the lower the data embedding capacity. Considering the purpose of secret data transmission, how to maximize data embedding capacity while maintaining a good visual quality of stego-images is an important issue. The proposed method is a hybrid version that combines the image compression process in AMBTC and LSB substitution techniques to adaptively determine the number of secret bits to embed into cover images. The experimental results demonstrate that the proposed method not only successfully improves the performance of the data embedding capacity but is also able to maintain a good visual quality of restored images after data extraction.

This paper presents a novel secret data hiding method that incorporates AMBTC and LSB substitution techniques to improve the data embedding capacity with good visual quality. Section 2 provides the background information and previous related contributions. The proposed method will be described in detail in Section 3. Section 4 includes the experimental results to compare the performance of the proposed method and previous works. Finally, the conclusions are described in Section 5.

2 Related Works

In this Section, we will briefly describe the main concept of AMBTC data compression and some previous AMBTC based data hiding methods (such as the methods presented by Ou *et al.*, Huang *et al.*, Kumar *et al.*, and Horng *et al.*).

2.1 Absolute Moment Block Truncation Coding (AMBTC)

Lema and Mitchell presented an image compression method named Absolute Moment Block Truncation Coding (AMBTC) [14] that is used to improve the compression performance of Block Truncation Coding (BTC). First, an image is divided into non-overlapping blocks and denoted as $I = \{B_i | i = 0, 1, \ldots, N_B - 1\}$ where B_i represents the i-th image block, N_B is the total number of blocks after division, and each block B_i is sized $w \times w$.

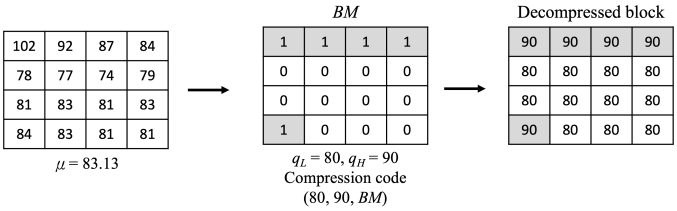


Figure 1: Example of AMBTC in data compression and decompression

Then, for each block B_i compute the mean pixel value μ_i by using Equation (1). Note that, p_j is the j-th pixel value in block B_i. After that, use μ_i as a threshold to compare all pixels in B_i and generate B_i's bitmap BM_i. Here, BM_i is the bitmap of block B_i, which is used to store the state of pixel p_j in B_i. If p_j is greater than or equal to μ_i, then BM_i is denoted as 1, otherwise it is denoted as 0.

$$\mu_i = \frac{1}{w \times w} \sum_{j=0}^{w \times w - 1} p_j. \tag{1}$$

Thus, every pixel in B_i is classified into two groups: higher-than-and-equal-to-μ_i group (denoted as $G_H = \{p_k | k = 0, 1, \ldots, n_h\}$, and $p_k \geq \mu_i$) or lower-than-μ_i group (denoted as $G_L = \{p_k | k = 0, 1, \ldots, w \times w - n_h\}$, and $p_k < \mu_i$). Further, to compute quantized values b and a (refer to Equations (2) and (3)) for G_H and G_L groups, respectively. After that, $\{b, a, BM_i\}$ is used as the compression code of block B_i.

$$b = \left\lfloor \frac{1}{n_h} \sum_{p_k \in G_H} p_k \right\rfloor. \tag{2}$$

$$a = \left\lfloor \frac{1}{w \times w - n_h} \sum_{p_k \in G_L} p_k \right\rfloor. \tag{3}$$

In the data decompression phase, for each block, b is used to fill pixel values in the restored block where the positions correspond to '1' in BM. While a is used to fill pixel values in the restored block in positions that map to '0' in BM.

The following simple example illustrates how AMBTC is applied to compress an image block (refer to Figure 1). As shown in the figure, the mean block value is $\mu = 83.13$ and the bitmap (BM) is 1 to indicate the locations of pixels bigger than μ; otherwise, the value 0 indicates the locations of pixel values smaller than μ. Then, the quantized values $b = 90$ and $a = 80$ are calculated by adopting Equations (2) and (3). According to the design in AMBTC, the compression code of the block is (80, 90, 1111 0000 0000 1000). In the decompressed block, BM locations with the value '1' are replaced by the high quantized value b (i.e., $b = 90$), while BM positions with the value '0' are replaced by the low quantized value a (i.e., $a = 80$). Finally, a decompressed block is obtained.

2.2 Data Hiding Method Using AMBTC

In 2015, Ou *et al.* [25] proposed a data hiding method using AMBTC that divides blocks into smooth blocks and complex blocks according to the difference between the low quantized value a and high quantized value b. Considering the characteristics of smooth blocks, the difference between the low and high quantized values is small, meaning that the content of associated BM locations does not significantly affect the visual quality of the block as observed by the human eye. Thus, taking $w \times w$ secret bits to replace the content of BM directly. On the other hand, for complex blocks, where the difference between the low and high quantized values is big, adjustment is made to the encoding sequence of quantized values to embed one secret bit. Since Ou *et al.*'s method takes the BM content and blocks into consideration and applies an adaptive strategy to modify the compression code to maintain a higher visual quality of decompressed images. The following section summarizes the key steps of Ou *et al.*'s data hiding method.

First, a cover image is divided into non-overlapping blocks sized $w \times w$. For each block, the method applies AMBTC to generate quantized values a and b, and bitmap BM. Then, the difference between a and b is calculated and denoted as d by using Equation (4). If $d \leq Thr$, where Thr is the predefined threshold, the block is treated as a smooth block; otherwise, it is a complex block.

$$a = d - a. \tag{4}$$

For smooth blocks (i.e., $d \leq Thr$), take $w \times w$ secret bits to form a temporary bitmap BM' and apply Equations (2) and (3) to compute new quantized values a' and b'. If the new difference $d' = b' - a' \leq Thr$, then the compression code of the block is (a', b', BM'); otherwise, (a, b, BM') is used as the compression code.

For complex blocks, taking one secret bit at a time, if the secret bit is '0', then the compression code of the block is (a, b, BM). If the secret bit is '1', then each bit in BM is inverted (i.e., from 0 to 1 and 1 to 0) to form a new bitmap BM', and (b, a, BM') is used as the compression code of the block.

2.3 Hybrid AMBTC Data Hiding Method

In 2017, Huang *et al.* presented a new AMBTC based hybrid data hiding method to improve the performance of Ou *et al.*'s method [8]. The main idea is to utilize the difference between two quantized values to embed the secret data. According to their design, the performance of the data embedding capacity has been improved while a good visual quality of the decompressed image is maintained. The key steps of Huang's method are summarized as follows.

First, a cover image is divided as non-overlapping block sized $w \times w$. For each block apply AMBTC to get the low quantized value a, high quantized value b, and bitmap BM. A predefined threshold Thr is used to determine whether a block is a smooth or complex block. Then, the difference between two quantized values is calculated by $d = b - a$, and the embeddable secret bits are obtained by $\log_2 Thr$. Subsequently, take $\log_2 Thr$ bits and convert the secret bits to a decimal value D and obtain the remaining value by adopting $R = d \mod Thr$.

For smooth blocks (i.e., $d \leq Thr$), apply Equations (5) and (6) to adjust the quantized values to obtain a' and b' according to the secret data. Also, similar with Ou *et al.*'s method, bitmap BM can be used to embed $w \times w$ bits of secret data. Therefore, the total number of embedded secret bits is calculated as $\log_2 Thr + w \times w$. Finally, the compression code of the block is obtained as (a', b', BM').

$$a' = \begin{cases} a - \left\lfloor \frac{D-R}{2} \right\rfloor, & \text{if } R \leq D, \\ a - \left\lfloor \frac{D-R}{2} \right\rfloor, & \text{if } R > D \text{ and } d + R - D \leq Thr, \\ a - \left\lfloor \frac{D-R-Thr}{2} \right\rfloor, & \text{otherwise .} \end{cases} \tag{5}$$

$$b' = \begin{cases} b + \left\lceil \frac{D-R}{2} \right\rceil, & \text{if } R \leq D, \\ b + \left\lceil \frac{D-R}{2} \right\rceil, & \text{if } R > D \text{ and } d + R - D \leq Thr, \\ b + \left\lceil \frac{D-R-Thr}{2} \right\rceil, & \text{otherwise .} \end{cases} \tag{6}$$

For complex blocks (i.e., $d > Thr$), apply Equations (7) and (8) to calculate two quantized values a' and b' based on a and b. To avoid affecting the visual quality of decompressed images, this method embeds one secret bit into the compression code. If the secret bit is '0', the compression code is (a', b', BM). If the secret bit is '1', then each bit in BM is inversed (i.e., from 0 to 1 and 1 to 0) to form the new bitmap BM' and the compression code of the block is obtained as (b', a', BM'). Thus, a complex block can contain $\log_2 Thr + 1$ secret bits.

$$a' = \begin{cases} a - \left\lfloor \frac{D-R}{2} \right\rfloor, & \text{if } R \leq D, \\ a - \left\lfloor \frac{D-R}{2} \right\rfloor, & \text{if } R > D \text{ and } d + R - D > Thr, \\ a - \left\lfloor \frac{D-R+Thr}{2} \right\rfloor, & \text{otherwise .} \end{cases} \tag{7}$$

$$b' = \begin{cases} b + \left\lceil \frac{D-R}{2} \right\rceil, & \text{if } R \leq D, \\ b + \left\lceil \frac{D-R}{2} \right\rceil, & \text{if } R > D \text{ and } d + R - D > Thr, \\ b + \left\lceil \frac{D-R+Thr}{2} \right\rceil, & \text{otherwise .} \end{cases} \tag{8}$$

2.4 Kumar *et al.*'s Data Hiding Method [13]

In 2019, Kumar *et al.* presented an AMBTC-based data hiding method. Kumar *et al.*'s method uses two thresholds to classify blocks into three types: smooth block, low complex block, and high complex block. In addition, the method employs a different data embedding strategy for each block type. The procedure of Kumar *et al.*'s method is summarized as follows.

First, a cover image is divided into non-overlapping block sized $w \times w$ and AMBTC is applied to obtain the low quantized value a, high quantized value b, and bitmap BM. Two predefined thresholds $Thr1$ and $Thr2$ combined with the difference between the quantized values

(i.e., $d = b - a$) classify a block into one of three types. Thus, for different block types as much secret data is concealed as possible using the associated data embedding strategy.

Smooth block: If $d < Thr1$, the block is a smooth block. Replace bitmap BM with $w \times w$ secret bits to form the new bitmap BM'. Then, apply Equations (2) and (3) to get the new quantized values a' and b', and the new difference value $d' = b' - a'$. If $d' < Thr1$, then use (a', b', BM') as the compression code of the block, otherwise use (a, b, BM').

Low complex block: If $Thr1 \leq d \leq Thr2$, the block is a low complex block. Create a new bitmap EBM by replacing data in the even-numbered columns of bitmap BM with $\frac{w \times w}{2}$ (i.e., $\frac{w}{2}$ is half the number of columns in BM) secret bits. Similarly, replace the data in the odd-numbered columns of bitmap BM to create another bitmap OBM. After that, compute the Hamming distance $h1$ between BM and EBM and Hamming distance $h2$ between BM and OEM. If $h1 \leq h2$, use (a, b, EBM) as the compression code of the block; otherwise, invert the data in OBM (i.e., from 0 to 1, and from 1 to 0) to form OBM' and use (b, a, OBM') as the compression code.

High complex block: If $d > Thr2$, the block is a high complex block. Apply the PVD method to embed secret data into the quantized values a and b. If $a - 3 \geq 0$, set $a = a - 3$, and if $b + 2 \leq 255$, then set $b = b + 2$ to ensure that the difference between a and b (i.e., $d' = b' - a'$) is always greater than $Thr2$. Refer to Table 1 to determine the number of secret bits (i.e., denoted as n) to be embedded. In Table 1, $r_j = [r_j^L, r_j^H]$ is the range segment, and r_j^L and r_j^H represent the low and upper bounds of the range segment, respectively. Next, convert n bits of secret data to decimal format (i.e., denoted as D). Apply $r_j^L + D$ to obtain the new difference value $d2$ that is used to adjust the quantized values a and b; then, use Equation (9) to get a' and b'.

$$(a', b') = \begin{cases} a - \left\lceil \frac{d2-d'}{2} \right\rceil, b + \left\lfloor \frac{d2-d'}{2} \right\rfloor & \text{if } |d' - d2| \text{ is even,} \\ a - \left\lfloor \frac{d2-d'}{2} \right\rfloor, b + \left\lceil \frac{d2-d'}{2} \right\rceil & \text{if } |d' - d2| \text{ is odd.} \end{cases} \quad (9)$$

Table 1: PVD range table of Kumar *et al.*'s method [13]

Difference range ($r_j = [r_j^L, r_j^H]$)	Embeddable length(n)
0 7	3
8 15	3
16 31	4
32 63	5
64 127	6
128 255	7

2.5 Horng *et al.*'s Data Hiding Method [7]

In 2020, Horng *et al.* utilized the concept of QVD with AMBTC and LSB substitution to create a new data hiding method [7]. Horng *et al.*'s method not only embeds secret data into the difference of the two quantized values but also uses the 2-LSB strategy to embed secret data into the quantized values to improve data embedding capacity. The key steps of the Horng *et al.*'s method are summarized as follows.

First, the cover image is divided into non-overlapping blocks sized $w \times w$ and, for each block, use AMBTC to obtain the two quantized values a and b, and bitmap BM. Data embedding is performed in three phases. In the first phase, the method computes the quotient of a and b divided by 4 (i.e., denoted as Q_a and Q_b) and the remainders of a and b (i.e., denoted as R_a and R_b) using Equations (10) and (11). Then, the difference value between Q_a and Q_b (i.e., $q_d = Q_b - Q_a$) is obtained to find the embeddable length n (see Table 2). Next, convert n secret bits to decimal format denoted as D. Then, apply the PVD strategy to embed the secret data by $d2 = r_j^L + D$, where r_j^L represents the lower bound of the difference d_q located in the difference range. Based on the PVD strategy, $d2$ helps to adjust the quantized values Q_a and Q_b to generate Q'_a and Q'_b for storing the secret data (refer to Equation (12)). Subsequently, 2-LSB strategy is adopted to embed the secret data into the values R_a and R_b and generate the quantized values a' and b' (refer to Equations (13) and (14)).

$$Q_a = \frac{a}{4}, Q_b = \frac{b}{4}. \quad (10)$$

$$R_a = a \mod 4, R_b = b \mod 4. \quad (11)$$

Table 2: PVD table of Horng *et al.*'s method [7]

Difference range ($r_j = [r_j^L, r_j^H]$)	Embeddable length(n)
0 3	2
4 7	2
8 15	3
16 31	4
32 63	5

$$(Q'_a, Q'_b) = \begin{cases} Q_a - \left\lceil \frac{d2-d_q}{2} \right\rceil, Q_b + \left\lfloor \frac{d2-d_q}{2} \right\rfloor & \text{if } d_q \text{ is even,} \\ Q_a - \left\lfloor \frac{d2-d_q}{2} \right\rfloor, b + \left\lceil \frac{d2-d_q}{2} \right\rceil & \text{if } d_q \text{ is odd.} \end{cases} \quad (12)$$

$$a' = Q'_a \times 4 + R'_a. \quad (13)$$

$$b' = Q'_b \times 4 + R'_b. \quad (14)$$

If $a' \neq b'$, then embed one secret bit into the compression code sequence. If the secret bit is '0', use (a', b', BM) as the compression code. On the other hand, if the secret bit is '1', invert the bit data in BM (i.e., form 0 to 1, and from 1 to 0) to form a new bitmap BM', then use (b', a', BM') as the compression code.

In the last phase, a pre-defined threshold Thr is used to check the complexity of a block. If $d' = |a' - b'| < Thr$, then the block belongs to a smooth block and $w \times w$ secret bits are used to overwrite data in the bitmap BM to generate a new bitmap.

$$d' = |a' - b'|. \tag{15}$$

3 Proposed Method

AMBTC is a useful image compression method. As mentioned previously, researchers have developed new data embedding methods based on AMBTC. However, these methods can still be further improved. Here, we present a hybrid AMBTC-based data hiding method that incorporates LSB substitution to improve data embedding capacity. The key idea in our proposed method is to utilize the property of the image contents, that is, using smooth image blocks to conceal more secret data than complex image blocks. In addition, our proposed method slightly modifies the compression code to embed one extra secret bit in an image block. Subsection 3.1 describes the key steps of the data embedding and compression code generation processes. Subsection 3.2 illustrates a simple example where the proposed data hiding method is used. Finally, the main steps of the data extraction and image restoration procedures are shown in Subsection 3.3.

3.1 Data Embedding and Image Compression

For security purposes, secret data must be encrypted using a crypto system before transmission to protect the data from malicious users. After the secret data is modified into cipher data using a crypto system, the receiver must complete the encryption key negotiation to ensure that the extracted data can be decrypted successfully. The proposed method modifies the procedure of AMBTC to conceal secret data for secure data transmission. First, an image is divided into non-overlapping blocks sized $w \times w$ and denoted as $I = \{B_i | i = 0, 1, \ldots, N_B - 1\}$. For each block B_i, compute the pixel mean value of B_i (denoted as μ_i). Next, compare the value of pixels in B_i with μ_i to classify the pixels into two groups: high group ($> \mu_i$) and low group ($< \mu_i$). After that, compute the mean values of the low group (denoted as a) and high group (b). The matrix BM, being the same size as B_i, is used to store the locations of the high group pixels (i.e., marked as '1') and low group pixels (i.e., marked as '0').

Observing the content of the block, a smooth block has a small difference value between a and b. On the other hand, a complex block has a large difference value between a and b. Thus, after our simulation, we discover that $d = |a - b| \leq 15$ indicates a smooth block and the embeddable secret bits $n = 3$; otherwise, B_i belongs to a complex block and $n = 4$. Then, the mean values a and b are modified as a' and b' by using n-LSB substitution to conceal n secret bits each in a and b. To maintain a good visual quality of the restored image, Equations (16) and (17) are adopted to reduce image distortion due to data embedding. Normally a' must be smaller than b', but after data embedding a' may be greater than b'. Thus, Equation (18) is used to modify a' to ensure that a' is smaller than or equal to b'. Considering data extraction on the receiver side, n' might be different from n on the sender side. Thus, $n' = 3$ in the case of $d' = |a' - b'| \leq 15$; otherwise, $n' = 4$. If $n' \neq n$, then Equation (19) is adopted to adjust the value of a' to make sure $n' = n$.

Further, considering that a' and b' might lead to an underflow or overflow issue, Equation (20) is applied as a resolution. Our goal is to embed more secret data into the compression code. So, the proposed method embeds one extra secret data bit s_j into the compression code for the case $a' \neq b'$. If $s_j = 0$, then the compression code is (a', b', BM). If $s_j = 1$, then change 0 to 1 and 1 to 0 in BM to generate BM' and use (b', a', BM') as the compression code for block B_i. Lastly, if $n' \leq Thr$, this means that B_i is a smooth block (i.e., a' value is close to b'). If the content of the bitmap is changed, any significant damage to the visual quality of decompressed images is not likely. When data embedding is done for all blocks, the compression code is generated and sent to the receiver.

$$a' = \begin{cases} a' - 2^n, & \text{if } a' \geq a + (2^{n-1} + 1), \\ a' + 2^n, & \text{if } a' \leq a - (2^{n-1} + 1). \end{cases} \tag{16}$$

$$b' = \begin{cases} b' - 2^n, & \text{if } b' \geq b + (2^{n-1} + 1), \\ b' + 2^n, & \text{if } b' \leq b - (2^{n-1} + 1). \end{cases} \tag{17}$$

$$(a', b') = \begin{cases} (a' - 2^n, b'), & \text{if } \frac{a'}{2^n} = \frac{b'}{2^n} and a' - 2^n \geq 0, \\ (a', b' + 2^n), & \text{else if } \frac{a'}{2^n} = \frac{b'}{2^n}, \\ (\frac{b'}{2^n} \times 2^n + a' \mod 2^n, \frac{a'}{2^n} \times 2^n + b' \mod 2^n), & \text{else.} \end{cases} \tag{18}$$

$$(a', b') = \begin{cases} (a' + 2^n, b'), & \text{if } |a' + 2^n - b| \leq |b' - 2^n - b| \text{ and } r_j = r_1 \text{ and } r'_j = r_2, \\ (a', b' - 2^n), & \text{else if } r_j = r_1 \text{ and } r'_j = r_2, \\ (a' - 2^n, b'), & \text{if } |a' - 2^n - a| \leq |b' + 2^n - b| \text{ and } r_j = r_2 \text{ and } r'_j = r_1. \\ (a', b' + 2^n), & \text{else.} \end{cases} \tag{19}$$

$$(a', b') = \begin{cases} (a' + 2^n, b' + 2^n), & \text{if } a' < 0 \text{ or } b' < 0, \\ (a' - 2^n, b' - 2^n), & \text{if } a' > 255 \text{ or } b' > 255. \end{cases} \tag{20}$$

3.2 Data Embedding Example

In this section, we present a simple example to illustrate the key steps of the proposed method (i.e., refer to Figure 2). Let a predefined threshold $Thr = 16$ and the encrypted secret data $M = m_1 \| m_2 \| m_3$. First, use AMBTC to compute the pixel mean value of a block $\mu = 134.25$ and the mean values of the higher and lower groups $b = 136$ and $a = 133$, respectively. According to Step 4 of the proposed method, the number of embeddable bits is $n = 3$. Applying the operations in Step 5, the quantized values

Algorithm 1 Data embedding and image compression procedures

1: Divide cover image I into non-overlapping blocks $\{B_i|I = 0, 1, \ldots, N_B - 1\}$ sized $w \times w$.
2: For each B_i apply AMBTC to get two quantized values (a, b) and bitmap BM.
3: Compute the distance between a and b using $d = |a - b|$.
4: If $d \leq 15$, set $n = 3$ (i.e., n is the number of secret bits to embed); otherwise, set $n = 4$.
5: Apply n-LSB substitution to embed n secret bits to the quantized values a and b. The quantized values are denoted as a' and b' that correspond to a and b, respectively.
6: Apply Equations (16) and (17) to adjust the values of a' and b' to reduce the distortion generated due to data embedding.
7: If $a' > b'$, then use Equation (18) to change a' so that $a' \leq b'$.
8: If $d' = |a' - b'| \leq 15$, then set $n' = 3$; otherwise, set $n' = 4$.
9: If $n' \neq n$, then apply Equation (19) to change a' and b' until $n' = n$.
10: If $a' < 0$ or $a' > 255$ or $b' < 0$ or $b' > 255$, then use Equation (20) to adjust a' and b' to resolve any overflow or underflow issues.
11: If $a' \neq b'$, then take the next secret bit s_j. If $s_j = 0$, then output the compression code (a', b', BM); else if $s_j = 1$, then convert 0 to 1 and 1 to 0 in BM to generate BM' and generate the compression code as (b', a', BM').
12: If $n' \leq Thr$, then take the next $w \times w$ secret bits from the secret bit stream to form the new bitmap BM'.
13: Repeat Steps 2 12 until the data embedding and compression procedures are completed on all blocks.

are $a' = 141$ and $b' = 128$ by using 3-LSB substitution to embed m_1= "000 101". Next, since $a' \neq b'$ and $m_2 = 1$, the encoding sequence of (a', b', BM) is changed to (b', a', BM) and each value in BM is inverted (ie., from 0 to 1 and 1 to 0) to obtain the modified bitmap BM'. Since $|141 - 128| = 13 < Thr = 16$, then BM' is replaced by m_3. Thus, the compression code is $(141, 128, BM')$.

3.3 Data Extraction and Image Decompression

When the receiver receives the compression code, the secret data and decompressed image can be obtained by the proposed data extraction process. The data extraction procedure is the inverse operation of the data embedding procedure. First, take a segment of the compression code to extract the secret data and restore the image. Here, the segment length is the total bits of (a', b', BM'). The new difference between a' and b' is calculated by $d'' = |a' - b'|$. If $d'' \leq Thr$, then the secret data m_e^3 is extracted from BM' and the length of the secret bits is $w \times w$. If $a' < b'$, output $m_e^2 = 0$; else, output $m_e^2 = 1$. Further, if $0 \leq d'' \leq 15$, set $n' = 3$; otherwise, set $n' = 4$. Next, extract n-LSB from a' and b' to form m_e^1. Repeat the decoding and data extraction procedures from the compression code segments sequentially until secret data have been extracted completely. Finally, the receiver adopts the decryption operation with the right key to get the original data that is human readable. Here, the decryption operation must be consistent with the encryption operation used on the sender side. The key steps of the proposed data extraction process are summarized as follows.

Algorithm 2 Data extraction and image decompression procedure

1: Take the compression code for a block (a', b', BM').
2: Compute $d'' = |a' - b'|$. If $d'' \leq Thr$, output $w \times w$ bits of secret data m_e^3 from BM'.
3: If $a' < b'$, output secret bit '0'; otherwise, output secret bit '1' and swap the values of a' and b'. Use m_e^2 to represent the extracted data.
4: If $0 \leq d'' \leq 15$, set $n' = 3$; otherwise, set $n' = 4$.
5: Output $2n'$ extracted secret bits (i.e., denoted as m_e^1), which corresponds to n' bits of binary data by computing $a' \mod n'$ and another n' bits of binary data by computing $b' \mod n'$.
6: Output the extracted secret data from the compression code segment $m_j = m_e^1 \| m_e^2 \| m_e^3$. Note that "$\|$" is the concatenation operation.
7: Append m_j to M'.
8: Restore the decompressed image block by using the information of (a', b', BM') with the AMBTC decompression procedure.
9: Repeat Steps 1 8 until all compression codes have been decoded.
10: Output the extracted message M' and decompressed image I'.

4 Experimental Results

To evaluate the performance of the proposed method, we implement the proposed method and previous methods using MATLAB R2019 running on Windows 10. Nine commonly-used test images (refer to Figure 3), taken from SIPI image database [24], were used in the experiments. As mentioned previously, the embedding capacity (EC) and visual quality of stego images are the two most important factors for evaluating the performance of a data hiding method.

Peak-Signal-to-Noise-Rate (PSNR) is a simple technique to evaluate the similarity between two images. In other words, PSNR can be used to measure how a stego image would look like compared to the original cover im-

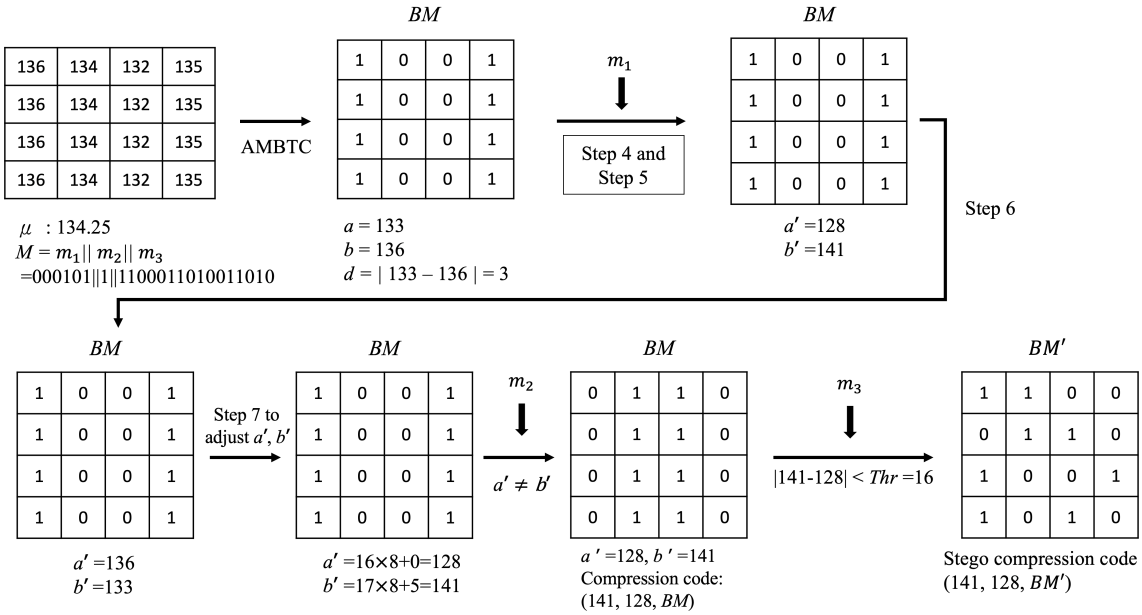


Figure 2: Hwang's scheme

(a) Airplane (b) Baboon (c) Boat

(d) Elaine (e) Lena (f) Male

(g) Peppers (h) Tank (i) Tiffany

Figure 3: Test images

age. PSNR is defined by Equations (21) and (22):

$$PSNR = 10 \times \log\left(\frac{255^2}{MSE}\right) \tag{21}$$

$$MSE = \frac{1}{W \times H} \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} (I_{ij} - I'_{ij})^2 \tag{22}$$

where MSE (Mean Square Error) calculates the distortion from the original pixel I_{ij} to the stego pixel I'_{ij}, W and H are the width and height of the image. As defined, a large PSNR value indicates that the stego image is very similar to the original image. On the contrary, a small PSNR value means the stego image is dissimilar to the original image. Generally, when the PSNR value is larger than 30dB, then the distortion on the stego image is not easily observed by the human eye.

Embedding capacity (EC) is used to evaluate the maximum bits of secret data that can be embedded into a cover image (refer to Equation (23)). Obviously, a data hiding method with a large value of EC can embed more secret data bits in a cover image than the one with a small EC value. Note that, in Equation (23), s_i is the secret data embedded into block B_i and $\|\cdot\|$ measures the total bits of s_i.

$$EC = \sum_{i=0}^{N_B-1} \|s_i\|. \tag{23}$$

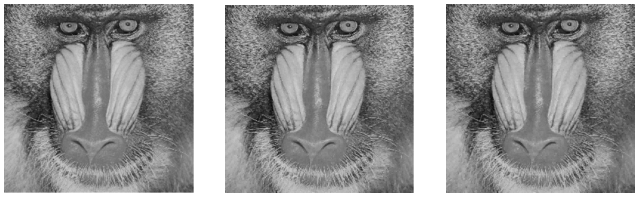
In our experiments, a random bit stream is used as secret data in data embedding and extraction simulations. Table 3 shows the results of evaluating the embedding performance using different Thr values. It clearly shows that a large Thr value provides better performance in terms of EC, but the visual quality of decompressed images is reduced. Considering the visual imperceptions of malicious

(a) AMBTC (PSNR=33.23 dB)

(b) Horng *et al.*'s method(PSNR= 29.744 dB, EC= 312,107 bits)

(c) Proposed (PSNR=30.249dB, EC= 322,428 bits)

Figure 4: Simulation results of Lena image



(a) AMBTC (PSNR=28.67 dB)

(b) Horng *et al.*'s method(PSNR= 26.76 dB, EC= 218,641 bits)

(c) Proposed (PSNR=27.211dB, EC= 23,3421 bits)

Figure 5: Simulation results of Baboon image

users on decompressed images, the $PSNR$ value must be greater than or equal to 30dB of the decompressed image. Thus, a suitable value for Thr is 8 or 16.

Figure 4(b) is the restored Lena image generated by Horng *et al.*'s method (i.e., $PSNR = 29.744$, and $EC = 312,107$ bits) and Figure 4(c) is the decompressed image generated by the proposed method (i.e., $PSNR = 30.249$dB, and $EC = 322,428$ bits). As we can see, the proposed method not only has better visual quality representation than Horng *et al.*'s method, but also can embed more secret bits. In simulations using complex images (refer to Figure 5), the $PSNR$ and EC of our proposed method (27.156dB/233,421 bits) are much better than Horng *et al.*'s (26.76dB/218,641bits).

To evaluate the performance of the proposed method, the experiment also implements Ou *et al.*'s method [25], Huang *et al.*'s method [8], Kumar *et al.*'s method and Horng *et al.*'s method, and comparing the performance between the proposed method to these methods. Normally, AMBTC based data hiding methods utilize three possible phases for data embedding. We count the occurrence of these phases and summarize the results in Table 4. The first phase is LSB substitution that embeds data into quantized values a and b directly. The second phase changes the encoding sequence of quantized values a and b, which means that (a', b', BM) is used to represent secret bit '0' and (b', a', BM) for secret bit '1'. The third phase is to embed the secret data into bitmap BM.

Since the complexity of the image content is a factor impacting the performance of data embedding, we include both complex images (e.g., Baboon) and smooth images (e.g., Lena) as test images. In the simulation, the image block size is 4×4 and blocks are classified into one of three types: smooth block, low complex block, and high complex block. Table 4 summarizes the embedding capacity for different block types using different data embedding methods. From Table 4, we can see that smooth blocks normally have better embedding capacity than the other two block types. The reason is that the bitmap of the smooth blocks can be used to embed more secret data compared to using the bitmaps for low and high complex blocks. Also, when embedding secret data into quantized values a and b, Horng *et al.*'s method can embed 6 9 bits into complex blocks, the proposed method 6 8 bits, and Kummar *et al.*'s method just 3 7 bits. The embeddability of Huang *et al.*'s method is affected by the predefined threshold setting. The proposed method is dependent on the block properties to adaptively determine the number of secret bits that can be embedded a large difference between the two quantized values means that more secret data bits can be embedded. Ou *et al.*'s method does not use special strategies for embedding secret data into quantized values, so we have no data to compare for this situation.

In the process of compression code generating of quantized values, Ou *et al.*'s and Huang *et al.*'s methods only embed one secret bit in a complex block. Kumar *et al.*'s method does not include a data embedding strategy for this situation, so there is no information to compare. For complex blocks, only Kumar *et al.*'s method can embed 8 secret bits into the bitmap while other methods can't.

Table 5 summarizes the simulation results of the embedding capacity in three phases for different data embedding methods, where the predefined threshold Thr is set to 16. In the phase of quantized value sequencing arrangement, both Ou *et al.*'s and Huang *et al.*'s methods embed 3,712 bits, which is significantly lower than Horng *et al.*'s method (i.e., 100,579 bits) and the proposed method (i.e., 106,198 bits). The reason is that Ou *et al.*'s and Huang *et al.*'s methods only perform data embedding for complex blocks. As we can see, the proposed method can embed 5,619 secret bits more than Horng *et al.*'s method.

Table 6 shows the experiment results of the data hiding methods using the Baboon test image that is more complex. In the phase of quantized value sequence arrangement, both Ou *et al.*'s and Huang *et al.*'s methods embed 10,062 bits, which is lower than Horng *et al.*'s method and the proposed method. In the phase of embedding secret data into quantized values, Huang *et al.*'s method can embed 65,536 bits since threshold Thr is used to determine the embeddable length of secret bits regardless of the block types.

Table 7 compares the performance of visual quality and embedding capacity for the nine test images using different data hiding methods. From the experimental results, the proposed method with different threshold values provides average visual quality values 30.848 dB ($Thr = 4$), 30.395 dB ($Thr = 8$), 29.232 dB ($Thr = 16$), and 26.976 dB ($Thr = 32$). Since there is an inverse relationship between visual quality and embedding capacity, the bet-

Table 3: Results of evaluating the embedding performance using different Thr values

Image		$Thr = 0$	$Thr = 4$	$Thr = 8$	$Thr = 16$	$Thr = 32$
Airplane	PSNR	30.925	30.835	30.497	29.892	28.223
	EC	133083	198999	275388	319903	347970
	bpp	0.51	0.76	1.05	1.22	1.33
Baboon	PSNR	27.878	27.867	27.729	27.211	24.728
	EC	136775	150139	180689	233421	313252
	bpp	0.52	0.57	0.69	0.89	1.19
Boat	PSNR	30.132	30.072	29.711	28.746	26.657
	EC	129677	159897	219692	295109	344415
	bpp	0.49	0.61	0.84	1.13	1.31
Elaine	PSNR	32.395	32.276	31.509	29.421	27.621
	EC	123762	148473	214027	330152	369804
	bpp	0.47	0.57	0.82	1.26	1.41
Lena	PSNR	31.908	31.78	31.251	30.249	28.142
	EC	129902	183257	261274	322428	357101
	bpp	0.5	0.7	1	1.23	1.36
Male	PSNR	30.777	30.691	30.332	29.373	26.722
	EC	132223	169116	228639	292636	350327
	bpp	0.5	0.65	0.87	1.12	1.34
Peppers	PSNR	32.054	31.974	31.293	29.652	27.632
	EC	125494	155958	224522	321563	360904
	bpp	0.48	0.59	0.86	1.23	1.38
Tank	PSNR	31.172	31.132	30.685	29.044	25.736
	EC	130049	141714	183017	277966	367715
	bpp	0.5	0.54	0.7	1.06	1.4
Tiffany	PSNR	31.11	31.008	30.545	29.496	27.321
	EC	129536	168578	236017	307174	351487
	bpp	0.49	0.64	0.9	1.17	1.34
Average	PSNR	30.928	30.848	30.395	29.232	26.976
	EC	130056	164015	224808	300040	351448
	bpp	0.50	0.63	0.86	1.15	1.34

Table 4: Embedding capacity for different block types using different data hiding methods

Mehtod	LSB for quantized values a and b		Sequence of quantized a, b		Bitmap	
	Smooth blocks	Complex blocks	Smooth blocks	Complex blocks	Smooth blocks	Complex blocks
Ou *et al.* [25]	0		0	1	16	0
Huang *et al.* [8]	$\log_2 Thr$		0	1	16	0
Kumar *et al.* [13]	0	0/3 7	0	0/0*	16	8/0*
Horng *et al.* [7]	6 9		1	1	16	0
Proposed	6 8		1	1	16	0

* The number of secret bits can be embedded in low or high complex blocks

Table 5: Embedding capacity in three phases for different data hiding methods (Lena image, $Thr = 16$)

Method	LSB for quantized values a and b	Sequence of quantized a, b	Bitmap	EC (bits)
Ou *et al.* [25]	0	3,712	202,752	206,464
Huang *et al.* [8]	65,536	3,712	202,752	272,000
Kumar *et al.* [13]	9,121	0	216,688	225,809
Horng *et al.* [7]	13,395	100,579	198,400	312,374
Proposed	15,910	106,198	200,320	322,428

Table 6: Embedding capacity in three phases for different data hiding methods (Baboon image, $Thr = 16$)

Method	LSB for quantized values a and b	Sequence of quantized a, b	Bitmap	EC (bits)
Ou *et al.* [25]	0	10,062	101,152	111,214
Huang *et al.* [8]	65,536	10,062	101,152	176,750
Kumar *et al.* [13]	27,555	0	135,648	163,203
Horng *et al.* [7]	15,134	104,627	98,880	218,641
Proposed	16,295	119,206	97,920	233,421

ter the embedding capacity the poorer in visual quality the restored images. However, the proposed method can successfully achieve the goal of enhancing the embedding capacity while maintaining a good visual quality of the restored images.

5 Conclusions

Data hiding techniques provide a secure transmission method for secret data over computer networks. The proposed AMBTC based data hiding method includes three phases for embedding as much secret data as possible while maintaining a good visual quality of decompressed images after data extraction. Also, in the design of our proposed method, there is no extra information nor any impact on the compression rate. From the experimental results, the proposed method not only maintains a good visual quality of decompressed images but also increases the embedding capacity. Compared with the previous works, the proposed method has better performance.

References

[1] C.K. Chan and L.M. Cheng, "Hiding data in images by simple lsb substitution," *Pattern Recognition*, vol. 37, pp. 469–474, 3 2004.

[2] I.-C. Chang, Y.-C. Hu, W.-L. Chen, and C.-C. Lo, "High capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding," *Signal Processing*, vol. 108, pp. 376–388, 3 2015.

[3] K.-C. Chang, C.-P. Chang, P. S. Huang, and T.-M. Tu, "A novel image steganographic method using tri-way pixel-value differencing," *Journal of Multimedia*, vol. 3, pp. 37–44, 6 2008.

[4] Y.-H. Chen, C.-C. Chang, C.-C. Lin, and Z.-M. Wang, "An adaptive reversible data hiding scheme using ambtc and quantization level difference," *Applied Sciences*, vol. 11, pp. 635–647, 1 2021.

[5] K. A. Darabkh, A. K. Al-Dhamari, and I. F. Jafar, "A new steganographic algorithm based on multi directional pvd and modified lsb," *Information Technology and Control*, vol. 46, pp. 16–36, 3 2017.

[6] W. Hong, X. Zhou, and S. Weng, "Joint adaptive coding and reversible data hiding for ambtc compressed images," *Symmetry*, vol. 10, no. 7, pp. 254–267, 2018.

[7] J.-H. Horng, C.-C. Chang, and G.-L. Li, "Steganography using quotient value differencing and lsb substitution for ambtc compressed images," *IEEE Access*, vol. 8, pp. 129347–129358, 2020.

[8] Y.-H. Huang, C.-C. Chang, and Y.-H. Chen, "Hybrid secret hiding schemes based on absolute moment block truncation coding," *Multimedia Tools and Applications*, vol. 76, pp. 6159–6174, 2017.

[9] M. Hussain, A.W. A. Wahab, Y.I. B. Idris, A.T. S. Ho, , and K.-H. Jung, "Image steganography in spatial domain: A survey," *Signal Processing: Image Communication*, vol. 65, pp. 46–66, 7 2018.

[10] K.-H. Jung, "Data hiding scheme improving embedding capacity using mixed pvd and lsb on bit plane," *Journal of Real-Time Image Processing*, vol. 14, pp. 127–136, 9 2018.

[11] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, evaluations, and trends in future

Table 7: Comparison of visual quality and embedding capacity of different data hiding methods

Image	*Thr*	Ou *et al.*		Huang *et al.*		Kumar *et al.*		Horng *et al.*		Proposed	
		PSNR	EC	PSNR	EC	PSNR	EC	PSNR	EC	PSNR	EC
Airplane	4	31.914	127054	31.721	159822	29.68	166530	29.839	192827	30.835	198999
	8	31.758	172309	31.244	221461	29.434	202752	29.692	241322	30.497	275388
	16	31.366	203764	29.777	269300	29.24	223684	29.143	311834	29.892	319903
	32	30.382	228874	26.51	310794	28.336	241061	27.791	340848	28.223	347970
Baboon	4	28.669	24589	28.624	57357	26.7	104993	27.314	154156	27.867	150139
	8	28.607	56299	28.392	105451	26.427	125081	27.212	175733	27.729	180689
	16	28.25	111214	27.412	176750	26.064	163203	26.76	218641	27.211	233421
	32	26.739	185869	24.413	267789	24.488	214833	24.675	296142	24.728	313252
Boat	4	31.143	36229	31.049	68997	28.539	122723	29.16	178257	30.072	159897
	8	30.905	107044	30.388	156196	28.333	159997	28.912	218284	29.711	219692
	16	30.204	178504	28.736	244040	28.081	207253	28.266	281046	28.746	295109
	32	28.955	222589	25.726	304509	27.143	236846	26.384	334691	26.657	344415
Elaine	4	33.886	30199	33.732	62967	30.605	129476	31.513	187816	32.276	148473
	8	33.492	87514	32.635	136666	30.396	154861	31.059	236757	31.509	214027
	16	31.539	217639	29.277	283175	30.114	232482	29.756	311395	29.421	330152
	32	30.375	250924	26.354	332844	29.586	255032	27.654	362038	27.621	369804
Lena	4	33.125	86209	32.864	118977	30.564	142276	30.905	190667	31.78	183257
	8	32.789	159424	32.039	208576	30.32	194043	30.698	238694	31.251	261274
	16	32.085	206464	30.141	272000	30.076	225809	29.795	312374	30.249	322428
	32	30.667	237664	26.495	319584	28.94	246585	28.008	347923	28.142	357101
Male	4	32.005	60424	31.866	93192	29.256	131794	29.926	177788	30.691	169116
	8	31.794	115894	31.243	165046	28.966	167717	29.753	214627	30.332	228639
	16	31.041	175939	29.36	241475	28.586	206340	28.921	278840	29.373	292636
	32	29.318	228349	25.777	310269	27.606	240633	26.693	338643	26.722	350327
Peppers	4	33.462	36214	33.295	68982	30.311	127337	30.995	188080	31.974	155958
	8	33.064	110449	32.109	159601	30.073	162330	30.642	235254	31.293	224522
	16	31.803	206449	29.616	271985	29.83	225856	29.603	306852	29.652	321563
	32	30.499	241309	26.354	323229	29.039	248891	27.542	352863	27.632	360904
Tank	4	32.869	18889	32.765	51657	29.498	115298	30.84	164591	31.132	141714
	8	32.676	52564	32.096	101716	28.956	133995	30.523	196534	30.685	183017
	16	31.103	163024	29.101	228560	28.414	199710	29.26	256829	29.044	277966
	32	28.69	245059	25.246	326979	27.751	251349	25.949	354570	25.736	367715
Tiffany	4	32.455	43099	32.307	75867	29.662	124911	29.888	182018	31.008	168578
	8	32.097	130864	31.403	180016	29.419	175533	29.569	227370	30.545	236017
	16	31.329	190954	29.562	256490	29.06	215845	28.749	295203	29.496	307174
	32	29.879	231094	26.125	313014	28.229	242413	26.952	342311	27.321	351487
Average	4	*32.17*	51434	32.025	84202	29.424	129482	30.042	**179578**	30.848	164015
	8	*31.909*	110262	31.283	159414	29.147	164034	29.784	220508	30.395	**224807**
	16	*30.969*	183772	29.220	249308	28.829	211131	28.917	285890	29.232	**300040**
	32	*29.500*	230192	25.889	312112	27.902	241960	26.85	341114	26.976	**351442**

research," *Neurocomputing*, vol. 335, pp. 299–326, 2019.

[12] M. Khodaei and K. Faez, "A data hiding scheme using the varieties of pixel-value differencing in multimedia images," *IET Image Processing*, vol. 6, pp. 677–686, 8 2012.

[13] R. Kumar, D.-S. Kim, and K.-H. Jung, "Enhanced ambtc based data hiding method using hamming distance and pixel value differencing," *Journal of Information Security and Applications*, vol. 47, pp. 94–103, 8 2019.

[14] M. Lema and O. Mitchell, "Absolute moment block truncation coding and its application to color images," *IEEE Transactions on Communications*, vol. 32, pp. 1148–1157, 10 1984.

[15] L. Li, M. He, S. Zhang, T. Luo, and C.-C. Chang, "Ambtc based high payload data hiding with modulo-2 operation and hamming code," *Mathematical Biosciences and Engineering*, vol. 16, pp. 7934–7949, 8 2019.

[16] X. Liao, Q.-Y. Wen, and J. Zhang, "A steganographic method for digital images with four-pixel differencing and modified lsb substitution," *Journal of Visual Communication and Image Representation*, vol. 22, pp. 1–8, 1 2011.

[17] C.-C. Lin, X.-L. Liu, W.-L. Tai, and S.-M. Yuan, "A novel reversible data hiding scheme based on ambtc compression technique," *Multimedia Tools and Applications*, vol. 74, pp. 3823–3842, 6 2015.

[18] J. Lin, C.-C. Lin, and C.-C. Chang, "Reversible steganographic scheme for ambtc compressed image based on (7, 4) hamming code," *Symmetry*, vol. 11, no. 10, pp. 1236–1252, 2019.

[19] J. Lin, S. Weng, T. Zhang, B. Ou, and C. C. Chang, "Two-layer reversible data hiding based on ambtc image with (7, 4) hamming code," *IEEE Access*, vol. 8, pp. 21534–21548, 2020.

[20] A. Malik, G. Sikka, and H. K. Verma, "A high payload data hiding scheme based on modified ambtc technique," *Multimedia Tools and Applications*, vol. 76, pp. 14151–14167, 2017.

[21] A. Malik, G. Sikka, and H. K. Verma, "An ambtc compression based data hiding scheme using pixel value adjusting strategy," *Multidimensional Systems and Signal Processing*, vol. 29, pp. 1801–1818, 10 2018.

[22] Z. Ni, Y. Q. Shi, N. Ansari, , and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 354–362, 3 2006.

[23] A. F. Nilizadeh and A. R. Nilchi, "Block texture pattern detection based on smoothness and complexity of neighborhood pixels," *International Journal of Image Graphics and Signal Processing*, vol. 6, pp. 1–9, 4 2014.

[24] University of Southern California, *The USC-SIPI Image Database http://sipi.usc.edu/database/*. USA: Signal and Image Processing Institute, 2022.

[25] D. Ou and W. Sun, "High payload image steganography with minimum distortion based on absolute moment block truncation coding," *Multimedia Tools and Applications*, vol. 74, no. 21, pp. 9117–9139, 2015.

[26] A. K. Sahu and G. Swain, "An optimal information hiding approach based on pixel value differencing and modulus function," *Wireless Personal Communications*, vol. 108, pp. 159–174, 4 2019.

[27] A. K. Shukla, A. Singh, B. Singh, and A. Kumar, "A secure and high-capacity data-hiding method using compression, encryption and optimized pixel value differencing," *IEEE Access*, vol. 6, pp. 51130–51139, 9 2018.

[28] W. Sun, Z.-M. Lu, Y.-C. Wen, F.-X. Yu, , and R.-J. Shen, "High performance reversible data hiding for block truncation coding compressed images," *Signal, Image and Video Processing*, vol. 7, pp. 297–306, 10 2013.

[29] G. Swain, "Digital image steganography using nine-pixel differencing and modified lsb substitution," *Indian Journal of Science and Technology*, vol. 7, pp. 1444–1450, 9 2014.

[30] G. Swain, "A steganographic method combining lsb substitution and pvd in a block," *Procedia Computer Science*, vol. 85, pp. 39–44, 6 2016.

[31] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 890–896, 8 2003.

[32] D.C. Wu and W.H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, pp. 1613–1626, 2003.

[33] H.-C. Wu, N.-I. Wu, C.-S. Tsai, and M.-S. Hwang, "Image steganographic scheme based on pixel-value differencing and lsb replacement methods," *IEE Proceedings on Vision, Image Signal Processing*, vol. 152, pp. 611–615, 10 2005.

[34] C.-H. Yang, C.-Y. Weng, H.-K. Tso, and S.-J. Wang, "A data hiding scheme using the varieties of pixel-value differencing in multimedia images," *Journal of System and Software*, vol. 84, pp. 669–678, 4 2011.

[35] W. Zheng, C. Chang, and S. Weng, "A novel adjustable rdh method for ambtc-compressed codes using one-to-many map," *IEEE Access*, vol. 8, pp. 13105–13118, 2020.

Biography

Pei-Chun Lai received his Master degree in Information Management at Chung Hsing University, Taiwan. His research interests include image techniques, and data hiding.

Jau-Ji Shen received his Ph.D. in Computer Science and Information Engineering from National Taiwan University, Taiwan in 1988. He is currently a professor of Information Management at Chung Hsing University, Taiwan.

His research interests include image techniques, data techniques and software engineering.

Yung-Chen Chou received the BS degree in Management Information Systems from National Pingtung University of Science & Technology, Pingtung, Taiwan, Republic of China, in 1998, and the MS degree in Information Management from Chaoyang University of Technology, Taichung, Taiwan, in 2002. He received Ph.D. degree in Computer Science and Information Engineering in 2008 from the National Chung Cheng University, Chiayi, Taiwan. From February 2009 to July 2021, he was an Associate Professor of Asia University, Taichung, Taiwan. Since August 2021 he has been an Associate Professor of iSchool, Feng Chia University, Taiwan. His current research interests include steganography, watermarking, and image processing.

Industrial Internet Security Situation Prediction Based on NDPSO-IAFSA-LSTM

Peng-Shou Xie, Zong-Liang Wang, Nan-Nan Li, Peng-Yun Zhang, Jia-Feng Zhu, and Tao Feng
(Corresponding author: Zong-Liang Wang)

School of Computer and Communications, Lanzhou University of Technology
No. 36 Peng Jia-ping road, Lanzhou, Gansu 730050, China
Email: 1292094887 @qq.com
(Received July 26, 2022; Revised and Accepted Jan. 23, 2023; First Online Feb. 17, 2023)

Abstract

The traditional prediction method takes a long time to train, and the prediction accuracy is not high. Because of the above problems, to further improve the accuracy and efficiency of Industrial Internet security situation prediction, this paper improves the situation prediction method of a Long Short-Term Memory network(LSTM) based on the neural network, which can better process the data of time series. It proposes an Industrial Internet security situation prediction method based on NDPSO-IAFSA-LSTM by integrating nonlinear dynamic particle swarm optimization, improved artificial fish swarm algorithm, and LSTM. Furthermore, NDPSO and IAFSA are used to solve the problem that the two algorithms easily fall into local extremum during optimization and optimize the parameters of the LSTM network. Finally, this paper verifies the designed Industrial Internet security situation prediction method in a simulated network environment. The experimental results show that this method's mean square error and absolute error are 0.005 and 0.0209, respectively, which are less than the error of NDPSO-LSTM, LSTM, and IAFSA-LSTM prediction methods, and improve the accuracy of the Industrial Internet security situation prediction method.

Keywords: Artificial Fish Swarm Algorithm; Industrial Internet; Long Short –Term Memory Network; Nonlinear Dynamic Particle Swarm Optimization Algorithm; Security Situation Prediction

1 Introduction

With the rapid development of information technology and Industrial Internet, many new network infrastructures with low latency and high reliability have been widely used in industrial fields. The high speed and intelligence of the industry bring great convenience to enterprises, but at the same time, it also makes the security of industrial control systems isolated from the internet more complicated. Network security issues such as worms, hackers dragging libraries, 0-day exposure, and privacy data leakage are constantly emerging. Network security has become a common concern of the state, enterprises and individuals.

Industrial Internet is a new type of infrastructure, application mode and industrial ecosystem, which will be closely integrated with industrial economy. Through the comprehensive connection of people, machines, things and systems, a brand-new manufacturing and service system covering the whole industrial chain and the whole value chain is constructed, which provides a way to realize the digitalization, networking and intelligent development of industry and is an important cornerstone of the fourth industrial revolution [6]. Industrial Internet is not only a simple application in the industrial field, but its connotation and extension are more abundant. The Internet has made good progress in many industries such as steel, household appliances, electric power, electronic information, etc. However, with the popularization of Industrial Internet devices, security and privacy issues have become increasingly prominent. Predicting the current security situation of Industrial Internet before security incidents can help managers make the right decisions.

Given the randomness, complexity and timing of the environmental factors of the Industrial Internet, this paper improves the long-term and short-term memory network based on the Industrial Internet. Because of the close relationship between security situation and time, the Industrial Internet can be regarded as a time series prediction problem. On this basis, a long-term and short-term memory suitable for time series is selected, and the parameters are optimized by using nonlinear dynamic particle swarm optimization and artificial fish swarm optimization. The simulation and analysis of this method are conducted. The experimental results show that this method can effectively predict the security situation of industrial network,Accurately grasping the security situation of the network system can provide effective information for network managers to make security protection decisions, which provides a useful reference for future in-

formation security work of industrial network.

1.1 Related Works

Situation perception is a concept put forward from the military point of view. It is necessary to analyze the strength of the enemy in order to make a correct judgment in the war. In 1999, Bass put forward the concept of cyberspace situational awareness for the first time [16]. The technology of network security situational awareness [9, 19]can be used to collect the elements of network security threat in the Industrial Internet environment, and make reasonable analysis and evaluation. On this basis, managers analyzes the potential security risks in the network according to the obtained security state values, and then make corresponding security decisions.

Network security situation prediction is to evaluate and analyze the current network security situation and predict the future development trend according to the historical network data and existing data obtained from the known network environment,to make security decisions and improve the network defense ability [8]. In the field of Industrial Internet, a large number of nonlinear and time-varying data are generated every day. Traditional prediction methods,such as hidden Markov chain [3] time series [18]neural network [13] and random forest methods [1], are not suitable for predicting the security situation of Industrial Internet in the current network environment because of their poor fault tolerance and self-learning ability.

Finding a model with strong generalization ability and being good at dealing with nonlinear problems and the time series has become a research hot spot of industrial internet security situation prediction. In view of the long training time and the time correlation between various elements, Li etc [7] used a combination of simple cycle units and attention mechanism to selectively learn the influencing elements, which improved the prediction accuracy and reduced the training time. Dou *et al.* [2] proposed a method of using LSTM to detect anomalies. Zhu *et al.* [5] proposed using intelligent optimization algorithm to improve the convergence speed of the LSTM neural network model. Hu *et al.* [4]combined MapReduce with support vector machine (SVM), and used cuckoo search (CS) to optimize the parameters of SVM. Experiments show that this method has higher accuracy and shorter training time. Wang *et al.* [17]compared several network security situation prediction methods. Experimental results show that RBF is optimized by PSO, but the number of data samples used in the experiment is too small. Zhang *et al.* [20] proposed a network security situation prediction method based on BP neural network. The Seeker Optimization algorithm is used to determine the optimal weight, and the simulated annealing algorithm is brought in enhancing the global search ability of the algorithm, but BP neural network is not appropriate for time sequence data. Zhang *et al.* [21]proposed a situation prediction method based on improved convolutional neural network, combining the advantages of deep separable convolution and decomposition into smaller convolution, an improved convolution neural network security situation model based on composite convolution structure is proposed, and the mapping of situation elements and situation values is realized. Ahmed Abbasi *et al.* [11] predicted the network safety situation by analyzing the time series, thus enhancing the forecast accuracy. The method is to generate the countermeasure network and recursive variational self encoder, which are used in conjunction with the basic predictor to enhance the regularization of the time series forecast model.

1.2 Our Contributions

The main contributions of this paper are as follows:

1) We propose a security situation prediction method based on NDPSO-IAFSA-LSTM to predict the security situation of the Industrial Internet. This method makes use of the advantage that the long-term and short-term memory network can better deal with time series problems, and improves the structure of the long-term and short-term memory network to gradually lessen the effect of gradient vanishing.

2) We propose a hybrid nonlinear dynamic particle swarm optimization and an improved artificial fish swarm algorithm to optimize the parameters of the long and short memory network and improve the accuracy of prediction.

1.3 Organization

The rest of this paper is planned as follows. In the second section, related theories and improvement procedures of short-term and long-term memory networks, particle swarm optimization, and artificial fish swarm algorithm are introduced. The third section introduces our work in detail. The fourth part introduces the experiments and results. Finally, the fifth section concludes this paper and explains the future work direction.

2 Basic Theories and Methods

2.1 Improved LSTM Network Structures

Recurrent neural network (RNN) is an improved multilayer neural network, which solves the problem of long-term dependence. However, when the time series is too long, it is easy to cause gradient explosion and gradient disappearance. Therefore, the LSTM network is put forward. LSTM network is one of the recurrent neural networks. It is improved on the basis of RNN [10, 15, 22] and is different from the traditional RNN structure.LSTM has a stable and powerful ability in solving long-term and short-term dependence problems. Memory cells take the place of the hidden layer of traditional neurons and are the core of LSTM network. By adding input gate, forgetting

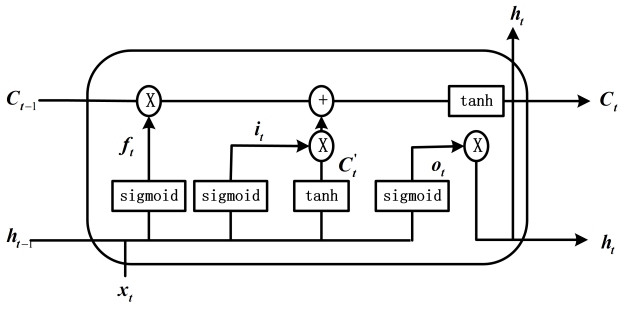


Figure 1: Improved LSTM network unit structure

gate and output gate,the problems of gradient vanishing and gradient explosion in model training is alleviated, and the deficiency of the traditional RNN model is made up.

Since there are three gates, namely the input gate, output gate, and forgetting gate, the LSTM network can add or delete information to the cell state. The prediction performance of the LSTM neural network mainly depends on the activation function. The input vector of the activation function includes the current input and the previous state and then predicts the output according to the results of the hidden layer.

In this paper, the input value is multiplied by the sigmoid function of the input gate, and the input value is multiplied by the tanh function in the candidate vector to reduce the influence of the gradient vanishing problem so that the LSTM has a more complex structure to capture the recursive relationship between the input layer and the hidden layer.

Figure 1 shows the cell structure of the improved LSTM network, which has four layers. In Figure 1, h_t, h_{t-1} is the output of the current unit and the previous unit; x_t is the input of the current unit; Sigmoid and tanh are activation functions; The circles in the figure all represent the arithmetic rules between vectors; C_t is the state of neuron at time t; f_t is the forgetting threshold,the threshold controls how cells should discard information through sigmoid activation function; i_t is the input threshold, which determines the information that needs to be updated by the sigmoid function, and then uses the tanh activation function to generate a new memory C_t, and finally control how much new information is added to the neuron state; o_t is the output threshold, which determines which parts of the neuron state are output by the sigmoid function, and the tanh activation function is used to process the neuron state, and the last consequence is got. The computation formula is as follows

$$\begin{aligned} f_t &= sigmoid(W_f.[h_{t-1}, x_t] + b_f) \\ i_t &= x_t * sigmoid(W_i.[h_{t-1}, x_t] + b_i) \\ C_t^{'} &= x_t * tanh(W_c.[h_{t-1}, x_t] + b_c]) \\ o_t &= sigmoid(W_o.[h_{t-1}, x_t] + b_o). \end{aligned}$$

Where W_f, W_i, W_o, W_c are the weight matrices corresponding to forgetting gate, input gate, output gate and neuron state respectively; b_f, b_i, b_o, b_c represent the corresponding offset constants respectively.

The functions of LSTM cells mainly contain tanh and sigmoid. tanh and sigmoid functions are shown in Formula (1) and Formula (2):

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1)$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

According to the above formulas, the state and output of neurons can be further calculated.

$$\begin{aligned} C_t &= f_t.C_{t-1} + i_t C_t^{'} \\ h_t &= o_t.tanh(C_t). \end{aligned}$$

The internal processing of neurons is completed through three control gate mechanisms, which ensures that the LSTM network can effectively use the input data, form a memory of the past long-term data and learn long-term dependence.

2.2 Improved Particle Swarm Optimization Algorithm

Particle swarm optimization (PSO) algorithm is a biologically inspired swarm intelligence optimization algorithm, which dates back to the research on bird predation [12,14]. This paper presents a nonlinear dynamic particle swarm optimization algorithm (NDPSO). The algorithm adjusts the parameters of particle swarm optimization nonlinearly, so that the search ability of particles changes continuously at different time, so as to balance global and local search ability of particles. At the same time, the cross-border particles are adjusted so that the cross-border particles do not gather at the boundary, which solved the problem that the particle swarm optimization algorithm is easy to fall into local extreme value, thus improved the optimization performance of the algorithm. NDPSO algorithm adjusts the speed and location of particles in each iteration, as shown below:

$$\begin{aligned} V_{i,j}^{t+1} &= \omega^t V_{i,j}^t + c_1^t r_1 (pbest_{i,j}^t + c_2 (gbest_j^t - x_{i,j}^t) \\ X_{i,j}^{t+1} &= X_{i,j}^t + V_{i,j}^{t+1} \\ \omega^t &= \omega_{min} + (\omega_{max} - \omega_{min}) \times (t/t_{max} - 1)^2 \\ c_1^t &= c_{1star} - (c_{1star} - c_{1end}) \times (\omega^t - 1)^2 \\ c_2^t &= c_{2star} + (c_{2end} - c_{2star}) \times (\omega^t - 1)^2 \end{aligned}$$

PSO algorithm sets the inertia factor ω as a constant. The proposed NDPSO algorithm reduces the parabolic shape, which can make the early particle search range larger and the latter particle search granularity finer.

The size of c_1 and c_2 of particle swarm optimization algorithm is 2. In this paper, c_1 and c_2 are set as function of the inertia factors ω. Therefore, the learning factor is related to the alteration law of inertia factor.Let $c_{1star} = c_{2end} = 3, c_{1end} = c_{2star} = 1$,so $c_1 + c_2 = 4$, which is more

in line with the set value of PSO algorithm. Similarly, the variation law is the same as the inertia factor ω. With the increase of iteration time, the self-learning ability of each particle decreases steadily and the social learning ability gradually increases. That is to say, in the early stage, we pay more attention to the free development of individuals, while in the later stage, we tend to look for the best position of the group. t is the current number of iterations, $V_{i,j}^t$ and $x_{i,j}^t$ represent the velocity and position of particle i in the jth dimension; $pbest_{i,j}^t$ is an individual extreme value, which is the best position found by particle i in history. r_1 and r_2 are random numbers between 0 and 1;$gbest_j^t$ is the global extremum, representing the best position that the example can find; V_{max} and X_{max} are the maximum speed and location of particles.

When $X_{i,j}^{t+1} > X_{max}$:

$$X_{i,j}^{t+1} = rand(X_{i,j}^t, X_{max}).$$

When $X_{i,j}^{t+1} < -X_{max}$:

$$X_{i,j}^{t+1} = -rand(X_{i,j}^t, X_{max}).$$

The above operations can make the crossover particles randomly return to the middle area between the previous position and the boundary position to continue optimization, to enhance the overall search capability of the particles.

The specific process of optimizing the LSTM neural network structure by NDPSO algorithm is as follows:

Step 1: Initialize algorithm parameters;

Step 2: Initialize LSTM neural network structure;

Step 3: Set the fitness function of NDPSO as the loss function of the neural network.

Step 4: Calculate the fitness function value of each particle;

Step 5: Renew the local optimal location of each particle and the global optimal location of the particle swarm;

Step 6: Renew the speed and location of each particle;

Step 7: If the maximum number of iterations is not reached. Go to Step 4. The flow chart is shown in Figure 2.

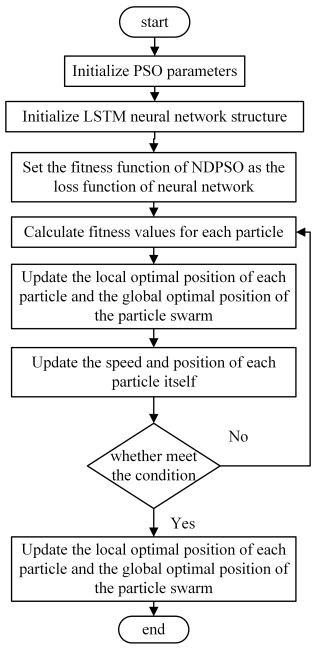


Figure 2: Flow chart of optimizing LSTM neural network structure by NDPSO algorithm

2.3 Improvement of Artificial Fish Swarm Algorithm

The artificial fish swarm algorithm (AFSA) is a new swarm intelligence optimization algorithm based on fish predation behavior.

The idea of artificial fish swarm algorithm is that within the range of fish swarm activities, the place with the most nutrients is generally the place with the largest density of fish swarm. The fish in this water area can discover food by themselves or by following the fish swarm. According to such features, the algorithm adopts the bottom-up design idea, simulates the behavior pattern of individual fish, and constructs artificial fish that abides by a certain behavior pattern, so as to achieve the purpose of optimization.

In the process of fish foraging, there is no need for unifying command. Each fish is foraging by using individual adaptive behavior. Therefore, the artificial fish swarm algorithm does not need centralized control and prior knowledge of related problems. It has no continuous or derivative requirements for the objective function and has strong adaptability. Secondly, the swarm behavior of artificial fish can prevent the local optimal problem, and make the fish swarm search the global extreme value as much as possible.

In this paper, the Metropolis criterion of simulated annealing algorithm (SA) is led into the hunting behavior, and then the Gaussian mutation operator is applied to the best artificial fish.

The idea of a simulated annealing algorithm (SA) originated from the annealing principle of solids. During the process of temperature rise, the particles inside the solid gradually become disordered. During the process of temperature fall, the particle state tends to be orderly, and each temperature has an equilibrium state in the process of temperature fall. According to the Metropolis criterion, the probability of reaching equilibrium at each temperature is $e^{-\triangle E/kT}$, where $\triangle E$ is the change of internal energy. Introduce metropolis criteria into foraging behavior.

The improved foraging behavior even if the food concentration in the next randomly selected state is lower than that in the current state food concentration also ac-

cepts this state with probability $e^{-\triangle E/kT}$ and approaches this state. According to the Metropolis criterion, the artificial fish accept a solution worse than the current state with a certain probability, which is conducive to the fish to jump out of the local extreme value. From the probability formula $e^{-\triangle E/kT}$, it can be found that the probability of accepting the worse solution gradually decreases with the increase of the number of iterations. The improved foraging behavior jumps out of the local extreme value in the early stage of the fish school, and the probability of jumping out decreases when it approaches the optimal domain in the late stage, and approaches the optimal solution.

In addition, after each iteration, the Gaussian mutation operator is introduced into the optimal artificial fish swarm, and the mutation can make it jump out of the local extremum. The Gaussian mutation operator is to add a Gaussian distribution random vector to the original individual, the mutation is shown as Equation (3.

$$X_{Gauss} = X_{ibest} + 0.5 * X_{ibest} * N(0,1). \quad (3)$$

Where X_{ibest} is the optimal fish in the i-th iteration, and N(0,1) is the Gaussian distribution with mean value of 0 and variance of 1.

After using Equation (3) to mutate the optimal artificial fish, if the objective function value after mutation is better than that before mutation, the mutated fish will replace the original optimal artificial fish; If the value of the objective function after mutation is not as good as that before mutation, the mutated fish shall be accepted with a probability of $e^{-\triangle E/kT}$ according to metropolis criteria.

In the artificial fish swarm optimization algorithm, by optimizing the search behavior of the fish swarm in each iteration, the fish swarm will not fall into local extremum, so as to improve the selection accuracy of the optimal solution region.

3 Industrial Internet Security Situation Prediction Method Integrating NDPSO, IAFSA and LSTM

The process of the Industrial Internet security situation prediction method integrating NDPSO, IAFSA and LSTM is shown in Figure 3.

Industrial Internet security situation prediction using the LSTM network is as follows:

Situation sequence X=$\{x_1, x_2, ..., x_k\}$ is used as situation prediction. However, the goal problem is based on $\{x_{k-m}, x_{k-m+1}, x_{k-1}\}$ predicts x_k. The problem can be expressed by Formula (4) as

$$x_k = f(x_{k-m}, x_{k-m+1}, x_{k-2}, x_{k-1}). \quad (4)$$

Where f represents the mapping between $\{x_{k-m}, x_{k-m+1}, x_{k-1}\}$ and x_k.

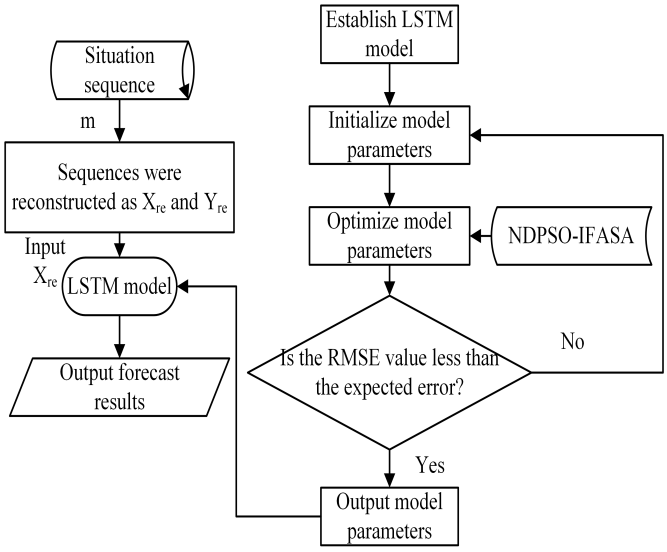


Figure 3: Industrial Internet security situation prediction method integrating NDPSO, IAFSA and LSTM

Firstly, the security situation sequence of Industrial Internet is reconstructed before modeling. The mapping relationship between sliding time window $X_k = \{x_{k-m}, x_{k-m+1}, x_{k-2}, x_{k-1}\}$ and output $\{x_k\}$ is represented by reconstruction, and the results are as follows:

$$X_{re} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \\ x_2 & x_3 & \cdots & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k-m} & x_{k-m+1} & \cdots & x_{k-1} \end{bmatrix}, Y_{re} = \begin{bmatrix} x_{m+1} \\ x_{m+2} \\ \vdots \\ x_k \end{bmatrix}$$

The reconstructed X_{re} is the reconstructed m-dimensional matrix, Y_{re} is the corresponding one-dimensional vector, m is the window length, and the final prediction error (FPS) is used to obtain the optimal window length m.

Secondly, the LSTM network model f of input X_{re} and output Y_{re} is established by using the historical industrial Internet security situation sequence. Three model parameters need to be decided, including the number of hidden layers, the number of hidden layer nodes and the learning rate η. The idea about artificial fish swarm and particle swarm optimization algorithm in the paper is to use the improved artificial fish swarm algorithm to globally optimize the population and determine the optimal solution domain. Then, the particle swarm optimization algorithm is initialized with the optimal solution of the improved artificial fish swarm, and local optimization is started until the termination conditions are met. The combination of the two algorithms not only meets the requirements of global optimization, but also greatly improves the local search speed.

The algorithm steps of improving the fusion of artificial fish swarm and particle swarm optimization are as follows:

1) Initialize the state of each artificial fish and the parameters of the algorithm.

2) The fitness value of each artificial fish is calculated according to the objective function of the particular

problem, and the information is transmitted to the bulletin board.

3) At the start of the artificial fish swarm iteration, the artificial fish will calculate the crowding and tail chasing behavior, and actually perform the behavior with large fitness value, which default to the special foraging behavior combined with the Metropolis criterion.

4) Gaussian mutation operator is used to mutate the optimal artificial fish species, and it is processed according to metropolis criteria.

5) The fitness value of the best artificial fish is compared with the information in the announcement, and the bulletin board is updated with the maximum fitness value.

6) Check the termination conditions (reaching the maximum number of iterations or reaching the minimum error requirement). If the conditions are met, the optimization is ended and the population information of the optimal value, optimal location and maximum number of iterations is recorded; otherwise, go to Step 3.

7) The group information of the maximum iteration algebra is transferred to the particle swarm, which is used as the initial value of the particle swarm, and the parameters of the particle swarm are set.

8) Assign the optimal value of the bulletin board to the global extreme value and the optimal position to the individual extreme value.

9) Adjust the location vector and speed vector of particles to generate new species.

10) Compute the fitness value of the new population, and update the individual extreme value and population extreme value.

11) Check the ending conditions (reaching the maximum number of iterations or the minimum error requirements), and end the optimization if the conditions are met; otherwise, go to Step 8.

Combined with nonlinear dynamic particle swarm optimization and improved artificial fish swarm algorithm, the parameters of LSTM network are optimized. The algorithm uses the adaptive value to assess particles, update the position and speed of particles, and achieve the minimum root mean square error. When RMSE meets the expected error, the iteration is stopped and the optimal solution is output. Otherwise, return to continue the iteration. After determining the model parameters, the LSTM network model f can be obtained.

$$RMSE = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(\hat{x}_i - x_i)^2}$$

Where $x_i \in X, i = 1, 2, ..., k, \hat{x}_i$ is the output value of the model.

Then use f model to predict the future Industrial Internet security situation. We can predict the future j situation values through the situation values at the first k moments of the sequence.

$$\begin{aligned} \hat{x}_{k+1} &= f(x_{k-m+1}, x_k - m + 2, \ldots, x_k) \\ \hat{x}_{k+2} &= f(x_{k-m+2}, \ldots, x_k, \ldots, \hat{x}_{k+1}) \\ &\vdots \\ \hat{x}_{k+j} &= f(\hat{x}_{k+1}, \hat{x}_{k+2}, \ldots, \hat{x}_{k+j-1}) \end{aligned}$$

Where $\hat{x}_{k+j}$ is the predicted value $k + j^{th}$ obtained from the set of m values before the value $k + j^{th}$ in X $j = 1, 2, ..., n$.

4 Experiment and Result Analysis

4.1 Experimental Environments and Used Dataset

Under windows10, cpu2.9G Hz and 4.0GB environments, python in version 3.8 is used as the development tool. Based on tensorflow, sklearn and other open source libraries. The natural gas pipeline dataset is selected as the experimental dataset. SCADA dataset is located on the industrial control system (ICS) cyber attack dataset website. True world raw data is generated using the natural gas pipeline system provided by the internal SCADA Laboratory of Mississippi State University (MSU). The system contains 274628 instances in total. Each row in the dataset contains multiple columns, commonly known as features, with a total of 17 features. Each instance in the dataset contains network traffic information and payload information. Different from the information technology network, the network topology of the SCADA system is fixed, and transactions between nodes are repetitive and regular. This static behavior helps IDSS discover abnormal activity. Payload information provides information about the status, settings, and parameters of the gas pipeline. These values are essential to understand how the system operates and detecting whether the system is out of bounds or critical. There are five different kinds of data in the dataset, including normal data and four other attack data. The four attacks are user-to-root (U2R), probe, denial of service (DOS), and remote-to-local (R2L). The list of features contained in this dataset is shown in Table 1.

4.2 Situation Time Series Data Preprocessing

There is no standard dataset in the Industrial Internet. In order to test the correctness of NDPSO-IAFSA-LSTM prediction, this paper uses the original natural gas

Table 1: Natural gas pipeline dataset feature list

	Features	Type		Features	Type
1	address	Network	11	control scheme	Command Payload
2	function	Command Payload	12	pump	Command Payload
3	length	Network	13	solenoid	Command Payload
4	setpoint	Command Payload	14	pressure measurement	Response payload
5	gain	Command Payload	15	crc rete	Network
6	reset rate	Command Payload	16	command response	Network
7	deadband	Command Payload	17	time	Network
8	cycle time	Command Payload	18	binary result	Label
9	rate	Command Payload	19	categorized result	Label
10	system	Command Payload	20	specific result	Label

pipeline data set for e-xperiments. In part of network data prediction, the first 60% (164776 examples) are selected as the training set, the first 20%(54926 examples) as the verification set and the last 20% (54926 examples) as the test set. Dataset contains default, continuous, and discrete data. Some default values in the data package of natural gas pipeline are shown in Table 2.

For missing values, the KNNimputer interpolation method is based on the KNNimputer interpolation method. KNNimputer searches for the nearest neighbor samples through the Euclidean distance matrix, and uses the mean value of the non empty value at the corresponding position of the nearest neighbor samples to fill in the missing values. For continuous data, to avoid dimensional inconsistency, the obtained sample data set needs to be standardized. The normalization formula is shown in the following equation:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{5}$$

min (x) and max (x) represent the minimum and maximum values of situation values respectively.

At the same time, we obtain the industrial Internet security situation value according to the analytic hierarchy process. Normalize it with Formula (5), and draw the situation value curve. Some situation values are shown in Figure 4.

Figure 4 shows some real values of the normalized industrial Internet security situation, which fluctuates between 0.3-0.8.

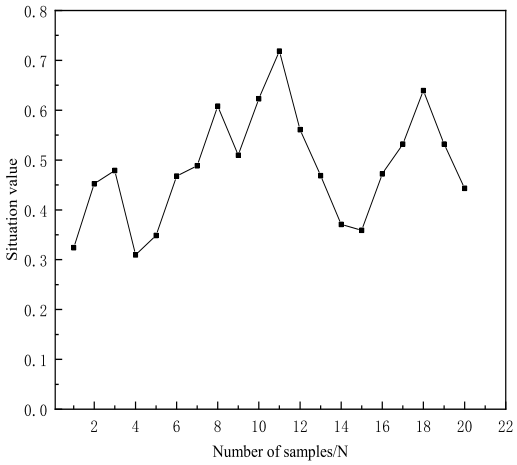


Figure 4: Normalized partial situation values

4.3 Analysis and Comparison of Experimental Results

4.3.1 Error Analysis

In order to evaluate the application of NDPSO-IAFSA-LSTM prediction technology in industrial networks, the average absolute error (MAE) and mean square error (MSE) are used to assess the prediction results:

$$MAE = \frac{\sum_{i=1}^{N} y_i - y_j}{N}$$

$$MSE = \frac{\sum_{i=1}^{N} (y_i - y_j)^2}{N}$$

N stands for the number of samples, represents the actual situation value of the Industrial Internet security situation, stands for the predicted value of Industrial Internet security situation. The larger the MAE and MSE, the smaller the accuracy of the algorithm. On the contrary, the forecast value is more accurate.

4.3.2 Comparison of Prediction Accuracy

The NDPSO-IAFSA-LSTM Industrial Internet security situation prediction method proposed in this paper has been verified by experiments. During the experimental verification, PSO-LSTM, LSTM and IAFSA-LSTM were used for comparison. The comparison data are shown in the Table 3, Table 4 and Figure 5.

The test results show that compared with the prediction results of PSO-LSTM, LSTM and IAFSA-LSTM, the

Table 2: Partial default values in data package of natural gas pipeline

			Gas pipeline Dataset-Packet features			
addr	funct	length	payload	crc	c/r	time stamp
4	3	16	-, -, -, -, -, -, -, -, -, -, -	12869	1	1418682163.170388
4	3	46	-, -, -, -, -, -, -, -, -, -, 0.689655,	12869	0	1418682163.269946
4	16	90	10,115,0.2,0.5,1,0,0,1,0,0, -	17219	1	1418682164.99559

Table 3: Prediction experiment results

Samples	**Real value**	**This paper**	**PSO-LSTM**	**IAFSA-LSTM**	LSTM
1	0.32423	0.30075	0.20125	0.29900	0.31083
2	0.45227	0.44390	0.29900	0.34963	0.47530
3	0.47919	0.46135	0.34788	0.42120	0.44531
4	0.30981	0.28155	0.25187	0.37232	0.28115
5	0.34862	0.31122	0.29726	0.46135	0.31028
6	0.46773	0.44215	0.38978	0.47008	0.43216
7	0.48870	0.46135	0.41247	0.53117	0.47154
8	0.60782	0.59227	0.45262	0.63242	0.58996
9	0.50983	0.47182	0.39152	0.44040	0.46895
10	0.62298	0.62369	0.41072	0.60274	0.61373
11	0.71830	0.69117	0.45262	0.63242	0.69178
12	0.56083	0.53292	0.42294	0.50150	0.52230
13	0.46879	0.40440	0.38987	0.41702	0.48980
14	0.37078	0.34963	0.32170	0.31122	0.39675
15	0.35904	0.33042	0.30250	0.35137	0.38570
16	0.47221	0.44215	0.35486	0.42294	0.44235
17	0.53184	0.51022	0.45087	0.52420	0.55686
18	0.63905	0.59052	0.47182	0.56259	0.58530
19	0.53214	0.54339	0.41945	0.53117	0.54338
20	0.44307	0.45262	0.29900	0.40199	0.46280

Table 4: Error comparison

Method	**Mean square error**	**Mean absolute error**
Method of this paper	0.0005	0.0209
PSO-LSTM	0.0174	0.1201
LSTM	0.0008	0.0210
IAFSA-LSTM	0.0053	0.0478

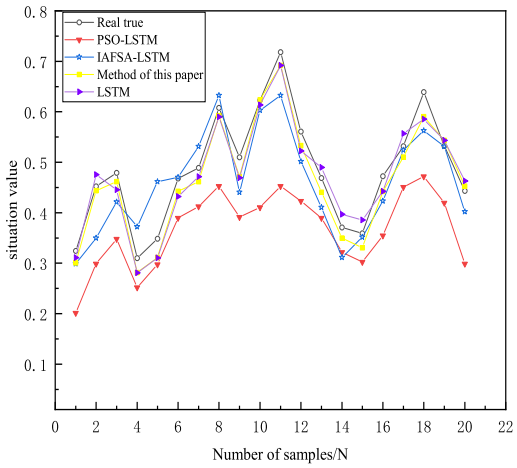


Figure 5: Comparison of prediction results of Industrial Internet security situation

prediction results of this method have fewer changes, the minimum error and higher accuracy.

5 Conclusions

Aiming at solving the problem of the Industrial Internet security situation prediction, this paper put forward an Industrial Internet security situation prediction algorithm based on the NDPSO-IAFSA-LSTM neural network. Two swarm intelligence algorithms are used to optimize the LSTM neural network and establish the corresponding prediction model. The hybrid algorithm combines the advantages of the improved artificial fish swarm algorithm in global optimization capability and high optimization accuracy with the advantages of particle swarm algorithm in local optimization ability and quick convergence speed so that the two algorithms complement each other and fill the defects between each other. To solve the problem of insufficient memory and gradient disappearance of RNN; The LSTM neural network with gating structure is used to control the ratio of input information to current stored information, and the gradient disappears. LSTM is specially designed to solve the long-term problems. All RNNs have a chain form of repetitive neural network modules. In the standard RNN, the repeating structure module has only a very simple structure, such as tanh layer. In order to solve the problem of difficult selection of the LSTM network parameters and easy to fall into local optimization, the particle swarm optimization algorithm is added to the network training. At the same time, in order to solve the problem that the particle swarm optimization algorithm is easy to fall into local extreme value, it is improved, which can effectively realize the optimization of the LSTM network parameters, quickly realize the global optimization, reduce the training time and improve the efficiency. Finally, four prediction algorithms are analysed and compared through experiments. The experimental results show that the prediction method proposed in this paper is basically consistent with the actual trend of the Industrial Internet security situation, and the prediction accuracy is better than the comparison algorithms. It can more quickly, accurately and effectively forecast the variable trend of the Industrial Internet security situation in the future for a period of time, and it has achieved good results in the field of situation prediction, but the disadvantage is that the simulation experiment is only carried out in the virtual environment, It was not tested in the true Industrial Internet environment. The focus of future research work is to conduct experiments in the real Internet environment to analyze and verify the

feasibility and effectiveness of this method.

Acknowledgments

This research is supported by the National Natural Science Foundations of China under Grants No.61862040 and No.61762059. The authors gratefully acknowledge the anonymous reviewers for their helpful comments and suggestions.

References

[1] Y. Chen, W. Zheng, W. Li, and Y. Huang, "Large group activity security risk assessment and risk early warning based on random forest algorithm," *Pattern Recognition Letters*, vol. 144, pp. 1–5, 2021.

[2] S. Dou, G. Zhang, and Z. Xiong, "Anomaly detection of process unit based on lstm time series reconstruction," *CIESC Journal*, vol. 70, no. 2, pp. 481–486, 2019.

[3] X. Hong, "Network security situation prediction based on grey relational analysis and support vector machine algorithm," *min J*, vol. 1, p. 2, 2020.

[4] J. Hu, D. Ma, C. Liu, Z. Shi, H. Yan, and C. Hu, "Network security situation prediction based on mr-svm," *IEEE Access*, vol. 7, no. 1, pp. 130 937–130 945, 2019.

[5] Z. Jiang and C. Sen, "Network security situation prediction method based on nawl-ilstm," *Computer Science*, vol. 46, no. 010, pp. 161–166, 2019.

[6] A. Lc, Z. A. Lei, A. Yy, G. A. Min, A. Sl, A. Yc, and B. Sha, "Iiot talent cultivating mechanism in line with industrial internet," *Procedia Computer Science*, vol. 199, pp. 377–383, 2022.

[7] Y. Li, B. Liu, L. Zhang, S. Yang, C. Shao, and D. Son, "Fast trajectory prediction method with attention enhanced sru," *IEEE Access*, vol. 8, pp. 206 614–206 621, 2020.

[8] Z. Li, T. Ma, Y. Zhou, and X. Wang, "Research and simulation of network security situation prediction algorithm," *Journal of Physics: Conference Series*, vol. 1941, no. 1, p. 012051 (7pp), 2021.

[9] X. Liu, J. Yu, W. Lv, D. Yu, Y. Wang, and Y. Wu, "Network security situation: From awareness to awareness-control," *Journal of Network and Computer Applications*, vol. 139, pp. 15–30, 2019.

[10] Y. Liu, D. Li, S. Wan, F. Wang, W. Dou, X. Xu, S. Li, R. Ma, and L. Qi, "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 135–151, 2022.

[11] S. H. A. Mahmood and A. Abbasi, "Using deep generative models to boost forecasting: A phishing prediction case study," in *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2020, pp. 496–505.

[12] S. Mubeen, M. S. Priya, and M. Vijayaraj, "A novel approach for predicting the tc center of remotely sensed images using pso based density matrix," *Earth Science Informatics*, vol. 15, no. 1, pp. 197–209, 2022.

[13] J. Sembiring, A. Amanov, and Y. Pyun, "Artificial neural network-based prediction model of residual stress and hardness of nickel-based alloys for unsm parameters optimization," *Materials Today Communications*, vol. 25, p. 101391, 2020.

[14] Z. Shafiei Chafi and H. Afrakhte, "Short-term load forecasting using neural network and particle swarm optimization (pso) algorithm," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–10, 2021.

[15] L. Shang, W. Zhao, J. Zhang, Q. Fu, Q. Zhao, and Y. Yang, "Network security situation prediction based on long short-term memory network," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2019, pp. 1–4.

[16] O. N. Toxirjonovich and X. G. Tulanovna, "Situational awareness gaps and opportunities for cyber security," *ACADEMICIA: An International Multidisciplinary Research Journal*, vol. 12, no. 1, pp. 512–518, 2022.

[17] G. Wang, "Comparative study on different neural networks for network security situation prediction," *Security and Privacy*, vol. 4, no. 1, p. 138, 2020.

[18] X. Wei, L. Zhang, H.-Q. Yang, L. Zhang, and Y.-P. Yao, "Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks," *Geoscience Frontiers*, vol. 12, no. 1, pp. 453–467, 2021.

[19] Y. Xue, "Research on network security intrusion detection with an extreme learning machine algorithm," *International Journal of Network Security*, vol. 24, no. 1, pp. 29–35, 2022.

[20] R. Zhang, M. Liu, Y. Yin, Q. Zhang, and Z. Cai, "Prediction algorithm for network security situation based on bp neural network optimized by sa-soa." *International Journal of Performability Engineering*, vol. 16, no. 8, p. 1171, 2020.

[21] R. Zhang, Y. Zhang, J. Liu, Y. Fan, and I. E. University, "Network security situation prediction method using improved convolution neural network," *Computer Engineering and Applications*, vol. 55, no. 6, pp. 86–93, 2019.

[22] X. Zhang, X. Wang, and S. Yin, "Multi-modal data transfer learning-based lstm method for speech emotion recognition," *International Journal of Electronics and Information Engineering*, vol. 13, no. 2, pp. 54–65, 2021.

Biography

Peng-shou 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 Security on Internet of Things. E-mail: xiepsh lut@163. com

Zong-liang Wang was born in Mar.1997. He is a master

student at Lanzhou University of Technology. His major research field is network and information security.E-mail: 1292094887@qq.com

Nan-nan Li was born in Feb.1997. She is a master student at Lanzhou University of Technology. Her major research field is network and information security. E-mail: magq1514@163.com

Peng-Yun 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

Jia-Feng Zhu was born in Jan. 1997. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: zhujiafeng688@163.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

Research on the Application of the Machine Learning Algorithm Based on Parameter Optimization in Network Security Situation Prediction

Xiaoyan Wang[1] and Jiangli Wang[2]
(Corresponding author: Jiangli Wang)

Puyang Petrochemical Vocational and Technical College, Puyang, Henan 457000, China[1]
Qinhuangdao Open University, Qinhuangdao, Hebei 066000, China[2]
Email: wangmian403@126.com
(Received Nov. 24, 2021; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)

Abstract

Situation prediction plays an important role in achieving network security. This paper mainly studied machine learning algorithms and selected the long short-term LSTM method for parameter optimization. The optimization effects of the improved particle swarm optimization (IPSO), adaptive chaos particle swarm optimization (APSO), and quantum particle swarm optimization (QPSO) were compared. The IPSO-LSTM, APSO-LSTM, and QPSO-LSTM methods were established and applied to situational predictions. It was found that the LSTM algorithm had a better prediction effect compared with the other machine learning algorithms, such as the support vector machine (SVM). Furthermore, the training time of the QPSO-LSTM algorithm was the shortest, 26.77 s, 27.24% shorter than the LSTM algorithm. Moreover, only one out of 28 samples was wrongly predicted by the QPSO-LSTM algorithm, and its RMSE, MAPE, and MAE were 0.13, 0.01, and 0.05, respectively, all of which were smaller than those of the other PSO methods. The experimental results demonstrate the advantages of the QPSO-LSTM algorithm in situation prediction. The QPSO-LSTM algorithm can be further applied in real networks.

Keywords: Machine Learning; Network Security; Parameter Optimization; Situation Prediction

1 Introduction

With the continuous development of the Internet, the scope of network coverage has been expanded, and the emergence of online payment and online shopping has brought convenience and efficiency to all walks of life. However, network security issues have also been seriously challenged [9] as the number of malicious programs and security vulnerabilities continues to grow. Some traditional network security methods, such as firewalls and intrusion detection [19], are effective and have limitations when dealing with large-scale, unknown attacks [12]. Therefore, Network Security Situational Awareness (NSSA) technology has emerged, a method that can predict future trends by assessing the current state [3], so that defenses can be deployed in advance to mitigate the harm of network attacks. Network security situational prediction (NSSP) is an important element in NSSA [8] and the key to solving network security problems [6], which has received extensive attention from researchers [16].

Tang *et al.* [13] designed an adaptive cloud improved genetic algorithm optimization extreme learning machine (CGA-ELM) method and found through simulation experiments that the prediction accuracy and convergence speed of the CGA-ELM method was improved by 4.9% and 64.28%, respectively, compared to the traditional GA-ELM method. Li *et al.* [5] optimized the radial basis function (RBF) neural network with an improved comprehensive learning particle swarm optimization (PSO) algorithm and found through experiments that the method had better accuracy and efficiency, suggesting more excellent prediction performance.

Wei *et al.* [18] extracted features from original sequential networks and then trained the recurrent neural network (RNN) with these features. The method performed well in prediction although it required more training time. Lin *et al.* [7] used a vulnerability prediction algorithm to predict the number of future vulnerabilities and combined it with a Bayesian attack graph to predict the subsequent attacks. They found through experiments that the method could accurate predictions of attacks. This paper focused on the long short-term memory (LSTM) method in machine learning, optimized its parameters using PSO,

and compared the performance of different PSO improvement methods. This work provides theoretical support for achieving accurate situation prediction.

2 Machine Learning-Based Situation Prediction Algorithm

Situation prediction is the ultimate goal of NSSA, which is a complex problem. Predicting the future network state can help network managers to prepare more fully for possible threats and take corresponding defensive measures in time [4]. Situation prediction means predicting future trends by analyzing historical data. Traditional methods for situation prediction include autoregressive models, etc., but these methods are poor in processing nonlinear data, resulting in low prediction accuracy. With the development of machine learning algorithms, algorithms such as SVM and neural networks have been applied in NSSP [20]. Neural networks have good self-learning ability so it can deal with nonlinear and time-varying data better. The commonly used networks include LSTM network and RBF neural network [14]. The LSTM algorithm has a wide range of applications in data prediction [15], and it can effectively process large-scale data; therefore, this paper focuses on the study of LSTM applications in NSSP.

LSTM is a special RNN structure that enables the network to have memory function through the gate structure. Suppose the input layer of the LSTM is $(x_1, x_2, \cdots, x_T)$ and the hidden layer is $(h_1, h_2, \cdots, h_T)$. Then, the LSTM at time t is written as: $f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$, where σ is the sigmoid activation function, h_{t-1} is the output of the previous layer, and w_f and b_f are the weights and biases of the forgetting gate. The result is $[0, 1]$, where 0 stands for completely discarded and 1 stands for completely retained.

Then, the input gates are used to determine which information needs to be stored. First, input gate i_t is updated: $i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$. Then, new cell state C_t is calculated:

$$\begin{aligned} \tilde{C}_t &= \tanh(w_c \cdot [h_{t-1}, x_t] + b_c), \\ C_t &= f_t \star C_{t-1} + i_t \star \widetilde{C}_t, \end{aligned}$$

where C_{t-1} is the memory of the previous moment. Finally, the final result of the LSTM is obtained by the output gate:

$$\begin{aligned} o_t &= \sigma(w_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t &= o_t \star \tanh(C_t). \end{aligned}$$

3 Improvement of LSTM by Parameter Optimization Method

In the training process, the LSTM involves many parameters. In order to further improve the prediction performance, the LSTM can be improved by means of parameter optimization, and the commonly used algorithms include genetic algorithm (GA), ant colony algorithm (ACO), and so on [11]. The PSO algorithm is a simple and stable optimization algorithm that has shown good applications in different fields [1] and is also a common method for parameter optimization of neural networks. Therefore, the LSTM is improved using the PSO algorithm.

In a J-dimensional search space, the population consists of particles. Let the number of iterations be k and the total number of particles be I. The velocity of every particle is: $V_{i,J}^k = [V_{i,1}^k, V_{i,2}^k, \cdots, V_{i,J}^k]$, and the position is: $X_{i,J}^k = [X_{i,1}^k, X_{i,2}^k, \cdots, X_{i,J}^k]$. During the motion of the particles, let its individual optimally position be $p_{i,J}^k$ and the best position of the population be g_J^k. The particle's update equation is:

$$\begin{aligned} V_{i,j}^{k+1} &= WV_{i,j}^k + c_1 r_1 (p_{i,j}^k - X_{i,j}^k) + c_2 r_2 (g_j^k - X_{i,j}^k), \\ X_{i,j}^{k+1} &= X_{i,j}^k + V_{i,j}^{k+1}, \end{aligned}$$

where W is the inertia weight, c_1 and c_2 are learning factors, and r_1 and r_2 are the random numbers in (0,1).

The PSO algorithm is prone to deviate from the optimal solution and fall into local optimum in the iterative process; therefore, in order to further improve the effectiveness of the PSO algorithm for LSTM parameter optimization, several optimized PSO methods are selected.

1) Improved PSO (IPSO): the values of c_1, c_2, and W are improved. Based on the trigonometric function, the values of c_1 and c_2 are improved:

$$\begin{aligned} c_1 &= c_{10} \cos(\pi \frac{k}{k_{max}}) + c_{11}, \\ c_2 &= -c_{20} \cos(\pi \frac{k}{k_{max}}) + c_{21}, \end{aligned}$$

where k_{max} is the maximum number of iterations. $c_{10} = c_{20} = 1$, and $c_{11} = c_{21} = 1.5$. Based on linear differential decrement strategy, the value of W improved:

$$W_k = W_{max} - \frac{W_{max} - W_{min}}{(k_{max}^2} \cdot k^2,$$

where W_{max} and W_{min} are the maximum and minimum values of W.

2) Adaptive PSO (APSO): Let c_1 and c_2 adjust adaptively according to the distance between $p_{i,J}^k$ and g_J^k. The corresponding equations are:

$$\begin{aligned} c_1 &= \frac{1}{(1 + \exp(-(p_{i,j}^k - X_{i,j}^k))}, \\ c_2 &= \frac{1}{(1 + exp(-(g_j^k - X_{i,j}^k))}. \end{aligned}$$

The adaptive adjustment of W is realized based on the cosine function. The equation is:

$$W_k = W_{min} + (W_{max} - W_{min}) \cdot \cos(\frac{\pi}{2} \cdot \frac{k}{k_{max}})$$

where W_{max} and W_{min} are the maximum and minimum values of W.

3) Quantum PSO (QPSO): The particle velocity is removed from the PSO. It is considered that every particle has a quantum state. For population I, the update formula for its particle position is:

$$\begin{aligned} X_{i,j}^{k+1} &= p_i \pm \gamma |I_{best} - X_{i,j}^k| \ln(\frac{1}{r}), \\ p_i &= \varphi p_{i,j}^k + (1-\varphi) g_j^k, \\ I_{best} &= \frac{1}{N} \sum_{i=1}^{i} p_{i,j}^k, \end{aligned}$$

where I_{best} refers to the average value of the historical best positions of the particles, φ and r are uniformly distributed values in (0,1), and γ is the innovation parameter, which is generally not greater than 1. In the QPSO, the particle velocity is no longer calculated. The particle position is updated according to the above formula during particle update until the maximum number of iterations or the global optimum is reached.

4 Results and Analysis

The experiment was conducted on a Windows 10 operating system with Intel(R) Core(TM) i7-10510U as the core processor. Python 3.7 was used for testing. The experimental data were obtained from the National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC), in which the Weekly Report of Network Security Information and Dynamics classifies the situation into five categories: excellent, good, medium, poor, and critical. They were converted into 5-1, i.e., excellent = 5, good = 4, medium = 3, poor = 2, and critical = 1. The data from the first issue of 2019 to the 52nd issue of 2021 were selected as the training samples (52 issues every year). The data of issue 1-28 in 2022 were used for testing. The experimental data are shown in Table 1.

The data in Table 1 were normalized according to $y_i = 2 \times \frac{x_i - x_{min}}{x_{max} - x_{min}} - 1$ and mapped to the interval of [-1,1] to accelerate the convergence of the neural network.

According to different parameter optimization methods, the parameters of the LSTM were optimized to obtain three NSSP methods, PSO-LSTM, IPSO-LSTM, APSO-LSTM, and QPSO-LSTM. The indexes for evaluating the prediction performance of the methods are shown below.

1) Root mean square error (RMSE): $RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i' - y_i)^2}$;

2) Mean absolute percentage error (MAPE): $MAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{|y_i' - y_i|}{y_i}$;

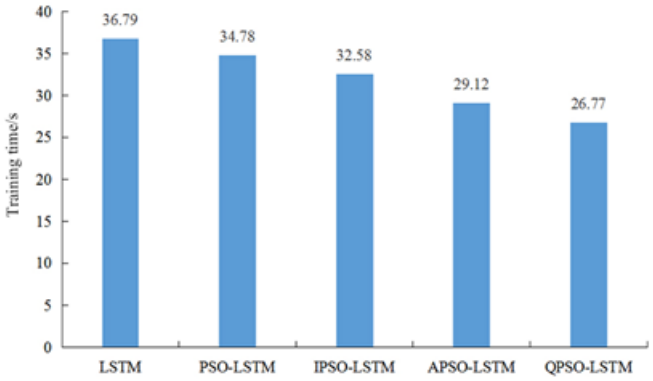


Figure 1: Training time of different NSSP methods

3) Mean absolute error (MAE): $MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i' - y_i|$.

In the equations, N is the number of samples, y_i' is the true value, and y_i is the predicted value. First, the prediction results of the LSTM method were compared with several other machine learning methods: the support vector machine (SVM) [10], the radial basis function (RBF) neural network [2], and the recurrent neural network (RNN) [17]. The results are demonstrated in Table 2.

It was seen from Table 1 that the RMSE of the LSTM algorithm was 4.27, which was 83.22% smaller than the SVM algorithm, 77.96% smaller than the RBF neural network, and 66.8% smaller than the RNN algorithm. It indicated that the prediction result of the SVM algorithm had the largest error with the actual value in the situation prediction. Moreover, the MAPE and MAE values were high in the SVM method, while the RMSE, MAPE, and MAE of the LSTM algorithm were the lowest among the four methods, which indicated that LSTM algorithm performed best in situation prediction among these machine learning methods. The results proved the reliability of the LSTM algorithm in solving the NSSP problem.

Then, the training speed of several LSTM methods was compared, as shown in Figure 1.

It was observed from Figure 1 that the training time of the LSTM algorithm was 36.79 s. After parameter optimization, the training time of the LSTM methods were reduced, which indicated that the parameter optimization did not increase the training time of the LSTM algorithm but improved the efficiency of computation. After PSO optimization, the training time of the PSO-LSTM algorithm was 5.46% shorter than that of the LSTM algorithm. The comparison of these LSTM methods demonstrated that the QPSO-LSTM algorithm had the shortest training time, 26.77s, which was 27.24% shorter than the LSTM algorithm, 23.03% shorter than the PSO-LSTM algorithm, and smaller than IPSO- and APSO-optimized LSTM algorithms. These results suggested the that QPSO-LSTM algorithm had the greatest advantage in training speed. The situation prediction results of the LSTM algorithm and the parameter-

Table 1: Experimental data

Sample number	2019.1	2019.2	2019.3	2019.4	2019.5	···	2022.26	2022.27	2022.28
Number of hosts infected with network viruses in the territory/million	20.1	19.2	21.1	23.6	23.3	···	6266.5	3319.7	2681.0
Total number of websites tampered with in the territory/n	669	690	638	701	318	···	1835	693	2381
Total number of websites implanted with back doors in the territory/n	553	805	561	600	424	···	670	662	601
Number of counterfeit pages against domestic websites/n	2940	3634	1718	1260	320	···	13001	9711	746
Number of new information security vulnerabilities/n	138	225	375	177	205	···	383	439	385
Security situation	4	4	4	4	5	···	4	4	4

Table 2: Comparison of prediction results between different machine learning methods

	RMSE	MAPE	MAE
SVM	25.44	0.42	24.33
RBF Neural Network	19.37	0.35	18.42
RNN	12.86	0.21	12.45
LSTM	4.27	0.05	3.53

optimized LSTM methods were compared, and the results are demonstrated in Table 3.

It was seen from Table 3 that the LSTM algorithm wrongly predicted six out of the 28 samples. After parameter optimization, the prediction performance of the PSO-LSTM algorithm was improved, and it wrongly predicted five samples. IPSO-LSTM, APSO-LSTM, and QPSO-LSTM algorithms wrongly predicted four samples, two samples, and one sample, respectively, indicating that the QPSO-LSTM algorithm performed best in situation prediction.

A comparison of the RMSEs between these LSTM methods is shown in Figure 2.

It was found from Figure 2 that after parameter optimization, the RMSE of the PSO-LSTM algorithm was 3.83, which was 10.3% less than the LSTM algorithm, indicating that the optimization of PSO effectively reduced the error between the predicted and actual values. After further improvement, the RMSEs of IPSO-LSTM, APSO-LSTM and QPSO-LSTM algorithms were all further reduced, among which, the RMSE of the QPSO-LSTM algorithm was the lowest, 0.13, which was 96.96% less than the LSTM algorithm. These results verified the reliability of parameter optimization for QPSO.

The comparison between these LSTM methods in MAPE is shown in Figure 3.

According to Figure 3, the MAPE of the LSTM algorithm was 0.05. After parameter optimization, the MAPE of PSO-LSTM was reduced to 0.04, which proved

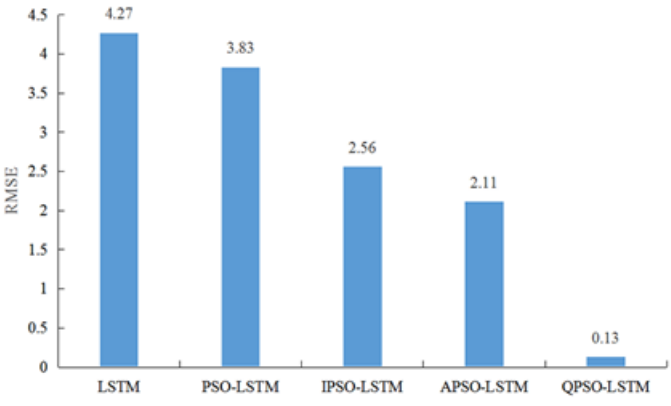


Figure 2: Comparison of RMSE between different NSSP methods

Table 3: Comparison of the situation prediction results between different NSSP methods

	Actual results	LSTM	PSO-LSTM	IPSO-LSTM	APSO-LSTM	QPSO-LSTM
1	Good	Excellent	Good	Good	Good	Good
2	Good	Good	Good	Good	Good	Good
3	Good	Good	Good	Good	Good	Good
4	Good	Good	Excellent	Excellent	Good	Good
5	Good	Good	Good	Good	Good	Good
6	Good	Good	Good	Good	Good	Good
7	Good	Difference	Good	Good	Good	Good
8	Good	Good	Good	Good	Good	Good
9	Good	Good	Medium	Good	Good	Good
10	Good	Good	Good	Excellent	Good	Good
11	Good	Good	Good	Good	Good	Good
12	Good	Medium	Good	Good	Excellent	Excellent
13	Good	Good	Good	Good	Good	Good
14	Good	Good	Medium	Good	Good	Good
15	Good	Good	Good	Good	Good	Good
16	Good	Excellent	Good	Good	Good	Good
17	Good	Good	Good	Good	Good	Good
18	Good	Good	Excellent	Good	Good	Good
19	Good	Good	Good	Good	Good	Good
20	Good	Medium	Good	Medium	Good	Good
21	Good	Good	Good	Good	Good	Good
22	Good	Good	Good	Good	Medium	Good
23	Good	Good	Medium	Good	Good	Good
24	Good	Good	Good	Good	Good	Good
25	Good	Good	Good	Good	Good	Good
26	Good	Medium	Good	Good	Good	Good
27	Good	Good	Good	Good	Good	Good
28	Good	Good	Good	Medium	Good	Good

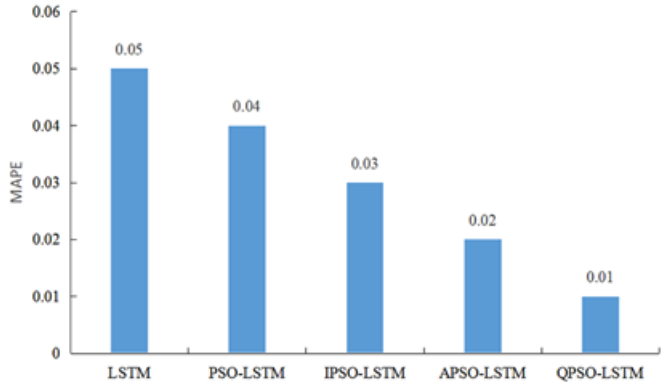


Figure 3: Comparison of MAPE between different NSSP methods

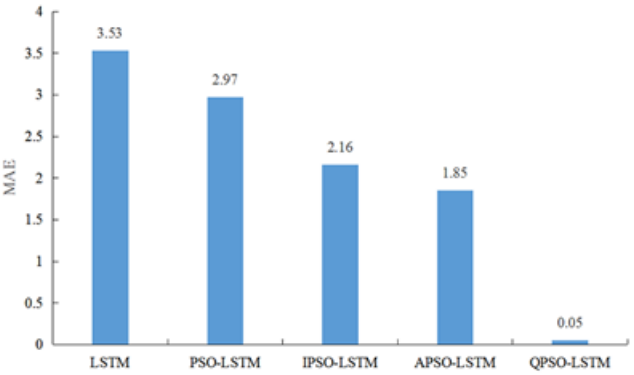


Figure 4: Comparison of MAE between different NSSP methods

the reliability of PSO optimization. Among these optimized methods, the QPSO-LSTM algorithm had the lowest MAPE, 0.01, which was 80% less than the LSTM algorithm and was also significantly lower than PSO-LSTM, IPSO-LSTM, and APSO-LSTM algorithms. The above results suggested that optimizing parameters with QPSO was the most effective and performed best in the situation prediction.

A comparison of the MAE between these LSTM methods is shown in Figure 4.

It was observed from Figure 4 that the MAE of the LSTM algorithm was the highest, reaching 3.53, and after PSO optimization, the MAE of the PSO-LSTM algorithm decreased to 2.97, showing a reduction of 15.86%. In comparison, the MAE of the first four methods was greater than 1, and only the MAE of the QPSO-LSTM algorithm was below 1, only 0.05, which was 98.58% less than the LSTM algorithm and 98.32% less than the PSO-LSTM algorithm. These results verified that the QPSO-LSTM algorithm was excellent in solving the NSSP problem and could accurately predict the future situation.

5 Conclusion

This paper mainly studied the LSTM method and optimized the parameters of the LSTM algorithm using PSO. Several improved PSO methods were proposed in order to further improve the effect of parameter optimization. It was found through experiments that the LSTM algorithm was more effective in solving the NSSP problem than the other machine learning methods. After parameter optimization, the LSTM algorithm had significantly improved prediction performance. In comparison, the QPSO-LSTM algorithm had a short training time and only wrongly predicted one sample, its RMSE, MAPE, and MAE values were the lowest, and its prediction results had the smallest errors with the actual values. The QPSO-LSTM algorithm can be further applied in the actual situation prediction.

References

[1] S. Aliwi, N. Al-Khafaji, H. Al-Battat, "A single-branch impedance compression network (ICN) optimized by particle swarm optimization algorithm for RF energy harvesting system," *Journal of Physics: Conference Series*, vol. 1973, no. 1, pp. 1-8, 2021.

[2] Y. Deng, X. Zhou, J. Shen, G. Xiao, H. Hong, H. Lin, F. Wu, B. Q. Liao, "New methods based on back propagation (BP) and radial basis function (RBF) artificial neural networks (ANNs) for predicting the occurrence of haloketones in tap water," *Science of The Total Environment*, vol. 772, no. 6, pp. 1-9, 2021.

[3] C. Ding, Y. Chen, A. M. Algarni, G. Zhang, H. Peng, "Application of fractal neural network in network security situation awareness," *Fractals*, vol. 30, no. 2, p. 2240090, 2022.

[4] M. Husák, J. Komárková, E. Bou-Harb, P. Čeleda, "Survey of attack projection, prediction, and forecasting in cyber security," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 640-660, 2019.

[5] R. Li, F. Li, C. Wu, J. Song, "Research on vehicle network security situation prediction based on improved CLPSO-RBF," *Journal of Physics: Conference Series*, vol. 1757, no. 1, pp. 1-8, 2021.

[6] Z. Li, T. Ma, Y. Zhou, X. Wang, "Research and simulation of network security situation prediction algorithm," *Journal of Physics: Conference Series*, vol. 1941, no. 1, pp. 1-7, 2021.

[7] P. Lin, Y. Chen, "Dynamic network security situation prediction based on bayesian attack graph and big data," in *IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC'18)*, pp. 992-998, 2018.

[8] Z. Lin, J. Yu, S. Liu, "The prediction of network security situation based on deep learning method," *International Journal of Information and Computer Security*, vol. 15, no. 4, pp. 386-399, 2021.

[9] B. Liu, Y. Li, "Research on the Predictability of Network Security Situation," in *IEEE 20th International Conference on Communication Technology (ICCT'20)*, pp. 1127-1133, 2020.

[10] Y. W. Liu, H. Feng, H. Y. Li, L. L. Li, "An improved whale algorithm for support vector machine prediction of photovoltaic power generation," *Symmetry*, vol. 13, no. 2, pp. 212, 2021.

[11] L. Qian, W. Wang, G. Chen, M. Yu, "A fetal electrocardiogram signal extraction method based on long short term memory network optimized by genetic algorithm," *Journal of Biomedical Engineering*, vol. 38, no. 2, pp. 257-267, 2021.

[12] L. Shen, Z. Wen, "Network security situation prediction in the cloud environment based on grey neural network," *Journal of Computational Methods in Sciences and Engineering*, vol. 19, no. 1, pp. 153-167, 2019.

[13] Y. Tang, C. Li, "CGA-ELM:A network security situation prediction model," in *International Conference on Computer Technology and Media Convergence Design (CTMCD'21)*, pp. 58-62, 2021.

[14] W. Tao, J. Chen, Y. Gui, P. Kong, "Coking energy consumption radial basis function prediction model improved by differential evolution algorithm," *Journal of the Institute of Measurement and Control*, vol. 52, no. 7/8, pp. 1122-1130, 2019.

[15] K. M. Tsiouris, V. C. Pezoulas, M. Zervakis, S. Konitsiotis, D. D. Koutsouris, D. I. Fotiadis, "A Long Short-Term Memory deep learning network for the prediction of epileptic seizures using EEG signals," *Computers in Biology and Medicine*, vol. 99, pp. 24-37, 2018.

[16] J. Wang, K. Li, G. Zhao, "Security situation prediction optimization model oriented to awareness quality assurance," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 46, no. 1, pp. 22-25 and 57, 2018.

[17] W. Wang, J. Chen, T. Hong, N. Zhu, "Occupancy prediction through Markov based feedback recurrent neural network (M-FRNN) algorithm with WiFi probe technology," *Building & Environment*, vol. 138, pp. 160-170, 2018.

[18] F. Wei, Y. Wu, Y. Fan, "A new method for the prediction of network security situations based on recurrent neural network with gated recurrent unit," *International Journal of Intelligent Computing and Cybernetics*, vol. 13, no. 1, pp. 25-39, 2020.

[19] J. A. Zhang, H. Luo, "A prediction method of network security situation based on QPSO-SVM," *International Journal of Circuits*, vol. 14, pp. 815-820, 2020.

[20] X. Zhang, Z. Ye, L. Yan, C. Wang, R. Wang, "Security situation prediction based on hybrid rice optimization algorithm and back propagation neural network," in *IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS'18)*, pp. 73-77, 2018.

Biography

Xiaoyan Wang received her B.S. degree and M.S. degrees from Zhengzhou University in 2009 and 2012 respectively. She works in Puyang Petrochemical Vocational and Technical College. Her research interests are in areas of computer network and data mining. She has participated in a number of scientific research projects, presided over 3 department level subjects, and presided over 1 provincial-level excellent course.

Jiangli Wang is currently an associate professor at Qinhuangdao Open University. She graduated from the Hebei Normal University with a research interest in mathematics and applied mathematics. She has published five papers in the past two years.

An Image Tamper-proof Encryption Scheme Based on Blockchain and Lorenz Hyperchaotic S-box

Qiu-Yu Zhang, Tian Li, 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: zhangqylz@163.com, kokolt@163.com
(Received June 7, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)

Abstract

In order to solve the problems of the existing image encryption methods, which can't effectively resist differential cryptanalysis, weak robustness, low security, and privacy, and are easy to tamper with in the process of storage and transmission, an image tamper-proof encryption scheme based on blockchain and Lorenz hyperchaotic S-box was proposed. Firstly, the plaintext pixel value of the original image is extracted, and xor with Lorenz hyperchaotic sequence is used to realize a round of diffusion. Secondly, Lorenz hyperchaotic S-box is used to carry out two rounds of diffusion on the diffused pixel values to complete image encryption. Finally, the encrypted image is converted into a 256-bit ciphertext image hash code. Finally, the smart contract is used to complete the Interplanetary File System (IPFS) storage and tamper-proof detection of the ciphertext image hash code. Compared with the existing scheme, the experimental results show that the proposed scheme not only ensures the security of image data transmission through the dual tamper-proof mechanism of on-chain and off-chain so that the attacker can't get the accurate original image through ciphertext image hashes but also realizes the automatic secure storage and tamper detection of ciphertext image information through a smart contract.

Keywords: Blockchain; Image Encryption; Lorenz Hyperchaotic S-box; Smart Contract; Tamper-proof

1 Introduction

With the continuous development of cloud computing and Internet technology, a wide variety of multimedia data and information spread rapidly and widely around the world, which has a huge influence on people's study, work and life. In order to save local storage space, users usually encrypt images and upload them to the cloud. However, the cloud platform itself has hidden risks and is vulnerable to external attacks, which make the accidents of losing user data happened one after another. Therefore, the security risks (malicious attack, tampering, copying or malicious use, etc.) of various image data information in the cloud should not be underestimated.

The secure transmission and privacy protection of digital images in cyberspace are easy to be challenged, while the traditional block encryption algorithm is easily attacked by exhaustive method. At present, digital image encryption algorithms are mainly divided into: encryption methods based on optical transformation [20], DNA encoding [3], neural network and cellular automata [25], cryptography [12], transform domain [18], chaotic system [4,13], etc. The cryptographic system based on chaos can show some excellent characteristics in complexity, security and computing power. Combined with the flexible, effective and secure characteristics of S-box algorithm, image encryption can be realized by dynamically scrambling image pixel values. In addition, the blockchain technology can be applied to the field of image privacy protection to realize the tamper-proof detection of images by taking advantage of its decentralization, non-tampering-proof, traceability and other characteristics. Meanwhile, the blockchain can be used to store ciphertext images and perform tamper-proof detection of ciphertext images, which can fully ensure the confidentiality, security and integrity of image data, and also ensure the security and privacy of communication and transmission process.

In recent years, in the research of image privacy protection combining blockchain and chaotic mapping, most of the image encryption methods used are based on one-dimensional or high-dimensional hyperchaotic systems, which are vulnerable to the attack of spatial reconstruction methods. Moreover, the anti-interference ability of chaotic sequence preprocessing is poor, which easily leads to the problem of low image encryption accuracy. In addition, some algorithms have the problem which diffusion operation has nothing to do with the original image. It

leads to slow encryption and decryption speed and negative impact on security. If the chaotic system based on continuous time is adopted, the integral operation is required when the chaotic sequence is generated, and the computation amount and complexity are high. Moreover, the limited accuracy of the computer may lead to the short period and poor randomness of the chaotic sequence. Therefore, the combination of such encryption algorithm and the blockchain technology can't maximize the advantages of blockchain.

To solve the above problems, an image tamper-proof encryption scheme based on blockchain and Lorenz hyperchaotic S-box is proposed. In this scheme, the Lorenz hyperchaotic system and S-box are used to encrypt and decrypt images, and realizes the tamper-proof of the ciphertext image off-chain. Blockchain is used as the basic framework for information storage and tamper detection of ciphertext images, so as to achieve tamper-proof in the chain of ciphertext images. The main innovations of this paper are:

1) An image encryption scheme combining Lorenz hyperchaotic and S-box is proposed, which can effectively resist various attacks such as differential attack, plaintext attack, statistical analysis attack, etc. And it ensures the security of off-chain image data.

2) Blockchain is used to realize the tamper-proof detection of encrypted images. A smart contract algorithm is designed to realize the automatic storage of ciphertext image hashes (IPFS and blockchain) and the tamper-proof detection on-chain, so that the image data does not have to face the threat of malicious servers, and the storage security of ciphertext images is guaranteed to the maximum extent.

3) A dual tamper-proof detection scheme based on Lorenz hyperchaotic S-box, image encryption algorithm and blockchain is designed. The security of image transmission process is effectively guaranteed through the automatic call of off-chain encryption algorithm and on-chain smart contract.

The rest of the paper is organized as follows: Section 2 summarizes the related research works. Section 3 gives the related technologies, including hyperchaotic mapping, construction of S-box and so on. Section 4 provides the description of the proposed scheme. Section 5 gives the experimental results and analysis, and compares the performance with different existing schemes. At last, there is the conclusion in Section 6.

2 Related Works

In recent years, chaotic image encryption has aroused great interest of researchers. Chaos mapping is usually iterated according to predefined parameters to generate a key matrix for image encryption arrangement and diffusion. For example, Huang *et al.* [4] and Liu *et al.* [13] proposed a scheme of synchronous image encryption by displacement-diffusion operation, which promotes the communication between confusion and diffusion, thus is producing a better security level and higher encryption efficiency, and effectively resisting common attacks. Lu *et al.* [16] proposed an image encryption scheme based on a new compound chaotic map (Logistic-Sine) and S-box, which can effectively resist the chosen plaintext attack and has a good application potential in real-time image encryption. Pourasad *et al.* [19] proposed the method of using two one-dimensional chaotic systems to display chaotic behaviors to encrypt images, which realized the simultaneous encryption of images in spatial domain and frequency domain. It improved the encryption efficiency and enhanced the encryption strength. Wang *et al.* [24] proposed a fast image encryption algorithm based on parallel computing system, which uses coupled mapping lattice to randomly generate a sequence of gray scales cale sequences to change the grayscalescale values and realize real-time fast encryption. Zhang [28] proposed a fast image encryption algorithm based on lifting transform and chaos, which is different from the traditional permutation diffusion structure and has high security, but this algorithm has some defects such as discrete, narrow chaotic range and incomplete output distribution. The new fractional chaotic map proposed by Talhaoui *et al.* [22] realizes fast image encryption, but it also has the problems of low iteration speed and unstable chaotic state.

Image encryption methods based on chaos are divided into four aspects: pixel-level chaos [13], bit-level chaos [14], block-level chaos [8] and S-box chaos [2, 5, 6, 10,16,17,23,26,27]. Because of the complex structure and large key space of the S-box constructed by chaos theory, the research on image privacy protection based on the chaotic S-box construction method is getting hotter and hotter. For example, Alshammari *et al.* [2] proposed a lightweight image encryption system that can be effectively implemented in highly restricted IoT devices. This system not only has good encryption effect, but also respects the limitation of sensor resources, and can meet many standards such as memory consumption, execution time and information entropy. Khan *et al.* [6] In order to improve the security of selective encryption of multimedia (image) data, an S-box selective encryption scheme based on chaotic equation is designed. Wang *et al.* [23] proposed an algorithm to construct S-box based on chaotic mapping and genetic algorithm, which was realized by changing the initial value and control parameters of chaotic mapping. Lin *et al.* [10] proposed a new image encryption algorithm based on Lorenz hyperchaotic mapping and RSA. Because RSA consumes a lot of time and reduces the encryption speed, it can't meet the encryption requirements of a large number of images. Huang *et al.* [5] proposed a two-way propagation arrangement framework for symmetric image encryption using chaos and S-box, which can be used for scrambling color images and grayscale images of any size, and has excellent confusion effect and high efficiency. Lu *et al.* [17] put forward an S-box design algorithm based

on compound chaotic system, which uses continuous-time compound chaotic system to construct S-box, and the S-box has enough elasticity to different attacks. Zhang *et al.* [27]used fractional logic mapping to construct S-box, which has better randomness and security against common attacks. Zahid *et al.* [26] proposed a simple, novel and dynamic linear trigonometric transformation to establish a preliminary S-box. Because the transformation is dynamic in nature, it can produce a powerful S-box when the parameter value changes slightly.

As a distributed storage scheme, blockchain technology has obvious advantages in multimedia privacy protection, such as information tamper-proof, anonymity and network stability. Some centralized services can solve the privacy leakage problem. Blockchain is used to store ciphertext images for tamper-proof detection, which can ensure the confidentiality, security and integrity of image data. And it also can ensure the security and privacy of communication and transmission. For example, Khan *et al.* [7] proposed a sensitive image encryption method based on blockchain in intelligent industry, which can effectively resist brute force attacks, and is very effective in preventing data leakage and protecting the privacy of images. However, it has certain inherent defects in verifying the admission control authority of users. Acharya *et al.* [1] proposed an image encryption scheme based on blockchain and feedback carrying shift register (FCSR). In order to limit the use of resources, this scheme uses IPFS for distributed storage and generates transaction hashes through blockchain network. Blockchain network guarantees that any tampering in the image will be detected, so this method has good security. Zhao *et al.* [29] proposed a color image encryption technology based on the combination of chaotic restricted Boltzmann machine (CRBM) and blockchain. According to Hnon-zigzag, this technology firstly arranges rows, then arranges columns, and then uses CRBM to deploy and replace, finally uses blockchain framework to detect the tampering of encrypted images in the transmission stage, which can effectively deal with various attacks. Li *et al.* [11] proposed a privacy protection method for medical images based on blockchain. The patient's privacy data was stored on different nodes of blockchain by using fragmentation technology, and then the medical images with non-sensitive information, hash return values and text records were stored on blockchain by using IPFS technology, which realized the effectiveness of patient information protection. Li *et al.* [9] proposed a new image encryption algorithm based on blockchain and fingerprint technology, which has good robustness and can well resist the chosen plaintext attack. Shen *et al.* [21] proposed a medical encryption image retrieval scheme based on blockchain, which transmits requests for users through intelligent contracts and feeds back the retrieval results according to the similarity of images. Liu et al [15]proposed a distributed access control system based on blockchain to ensure the security of IoT data, which only uses blockchain to complete the access control function. The data is still stored in the cloud, and there is still a certain security risk.

To sum up, most of the existing image encryption algorithms combine chaotic systems to encrypt images, but the low-dimensional chaotic systems have low encryption accuracy and poor anti-interference ability, Also the high-dimensional hyperchaotic systems are too complex, and there are still risks in the application of single chaotic encryption in image security. However, the existing S-box image encryption algorithm cannot effectively resist the chosen plaintext attack. Its structure is simple and its security is not high. Therefore, a third-order diffusion image encryption and decryption algorithm based on S-box and chaotic mapping is designed, which improves the current chaotic system's low sensitivity and low scrambling degree. In the design, the traditional scrambling mode and innovative third-order diffusion are adopted, which make the encryption method in this paper have better encryption performance and security. Considering the overall encryption performance, the dynamic S-box is generated by exclusive xor of Lorenz sequence by removing the interference term of chaotic sequence, which makes the S-box more uniform and better Strict Avalanche criterion (SAC). In addition, in the practical application of blockchain, the massive storage of data, the burden brought by the generation and consensus of blocks make the application of blockchain limited, and it is easy to be tampered with in the process of uploading to blockchain. Therefore, the encryption algorithm in this paper combines blockchain and IPFS to realize the secure storage and tamper-proof detection of ciphertext images, which ensures the security of images in the interactive process to the greatest extent.

3 Preliminaries

3.1 Lorenz Hyperchaotic Map

Chaotic systems are often used in encryption and secure communication because of their unpredictable, ergodic, pseudo-random behavior and high sensitivity to initial conditions. Among many proposed chaotic image encryption algorithms, Lorenz system [26] acts as a pseudo-random number generator to generate chaotic sequences, and Lorenz system can be expressed as an equation as shown in Equation (1).

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = cx - y - xz \\ \dot{z} = xy - bz \end{cases} \tag{1}$$

where x, y, z are system state variables, and a, b, c are control parameters of the system.

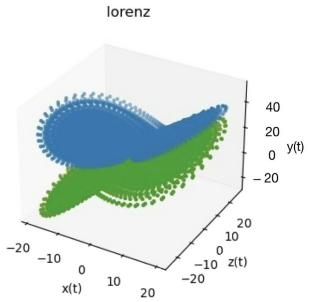
When $a = 10, b = 8/3$ and $c = 28$, Lorenz system is in a chaotic state. However, the Lorenz system has limited dimensions, complexity and ergodicity. In order to achieve better encryption effect, the original Lorenz system needs to be improved. Introducing the nonlinear controller w into the equation of Lorenz chaotic system, so that the

change rate of w is $-yz + rw$, a new chaotic system will be produced. And when only $a = 10, b = 8/3, c = 28$ and $r = -1$, the system is called Lorenz hyperchaotic system, and its equation is shown in Equation (2).

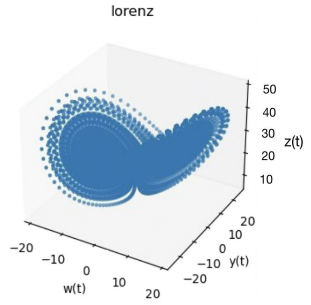
$$\begin{cases} \dot{x} = a(y - x) + w \\ \dot{y} = cx - y - xz \\ \dot{z} = xy - bz \\ \dot{w} = -yz + rw \end{cases} \tag{2}$$

where x, y, z and w are system state variables, and a, b, c and r are control parameters of the system.

Figure 1 shows a 3D model of Lorenz hyperchaotic system. The blue part in Figure 1(a) represents the trajectory of Lorenz hyperchaos in coordinates $x(t), y(t)$ and $z(t)$, and the green part is the projection of its trajectory to$x(t)$. Figure 1(b) represents the trajectory of Lorenz hyperchaos in coordinates $w(t), y(t)$ and $z(t)$.



(a) The trajectories in $x(t)$, $y(t)$ and $z(t)$



(b) The trajectory in $w(t)$, $y(t)$ and $z(t)$

Figure 1: Lorenz 3D model of hyperchaotic system

As can be seen from Figure 1, after long-term operation, Lorenz hyperchaotic system only moves in a limited area of 3D space, and the movement of Lorenz hyperchaotic system in this area is chaotic. Lorenz hyperchaotic system has more complex dynamic characteristics and higher randomness than low-dimensional chaotic system, so it is safer to use Lorenz hyperchaotic system for image encryption.

3.2 Structure of S-box

The S-box is the only nonlinear component in block cipher system, which converts the input plaintext block into ciphertext block. Generally, the mapping of a $m \times n$ S-box is: $\{0,1\}^m \rightarrow \{0,1\}^n$. Most S-box commonly used in cryptography satisfy the condition of $n = m$. At this time, the data in the process of encryption and conversion is neither compressed nor expanded, and S-boxes can realize complete reversible conversion.

Generally, there are two types of S-box construction methods: mathematical method and random generation. Randomly generating constructed dynamic S-boxes can increase the key space and improve security. Chaotic system has good cryptographic characteristics, such as parameter and initial value sensitivity, pseudorandom, etc., which can produce individual more complex dynamic behaviors. Using the sequence generated by chaotic system to construct S-box randomly makes it difficult to predict, so as to effectively improve the security of S-box construction.

Most of the existing image encryption algorithms use chaotic system to encrypt images. Lorenz hyperchaotic system has good pseudo-randomness and can effectively resist plaintext attacks. But low-dimensional chaotic system has low encryption accuracy and poor anti-interference ability; high-dimensional hyperchaotic system is too complex, and the application of single chaotic encryption in image security still has risks. Although the existing S-box image encryption algorithm structure is simple, it can't effectively resist the chosen plaintext attack. Therefore, this paper combines Lorenz hyperchaotic system with S-box to realize a more secure image encryption algorithm. For grayscale or color images with size $M \times N$, the method of constructing Lorenz hyperchaotic S-box is as follows:

According to the definition of Equation (2), set parameters a=10, b=8/3, c=28, r=-1. Set initial values of four variables x_0, y_0, z_0, w_0, through iteration n times, the chaotic sequence x_n, y_n, z_n, w_n is obtained. x_n=[$x(i)$], y_n =[$y(i)$], z_n =[$z(i)$], w_n =[w(i)]. The chaotic sequence x_n, y_n, z_n, w_n is converted into an 8-bit integer sequence $X(i)$, $Y(i)$, $Z(i)$, $W(i)$, by Equation (3), which is denoted as X, Y, Z, W.

$$\begin{aligned} F(i) = \mathrm{mod}(floor((|f(i)| \times 10^6 - \\ floor(|f(i)| \times 10^6)) \times 10^3), 256) \end{aligned} \tag{3}$$

where $floor(a)$ is to round a to the nearest integer less than or equal to a. The value of mod(x, y) is the remainder when x is divided by y, $L = M \times N$, $i = 1, 2, ..., L$.

Use the sequence X and Z for exclusive or operation to obtain the sequence S: S=$X \oplus Z$; 256 different numbers are randomly selected from the elements of S to form an S-box with a size of 16×16, and $S = [s(1), s(2), ..., s(256)]$. The Lorenz hyperchaotic S-box has high security because the S-box generated every time is random, and the S-box generated every time from the same original image is also different.

As shown in Table 1, an example S-box with the size of 16 × 16 is constructed by Lorenz hyperchaotic S-box when the parameters are a=10, b=8/3, c=28, r=-1 and L=262144.

Table 1: The proposed new S-Box

i/j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	91	84	187	21	120	66	156	129	198	236	223	210	136	45	147	67
2	5	228	22	100	58	34	82	105	170	154	199	125	220	20	55	181
3	222	193	254	177	251	139	132	12	231	225	150	28	180	83	89	77
4	238	111	196	207	173	38	143	128	176	88	145	151	148	134	3	123
5	221	54	10	32	244	102	226	114	217	138	188	135	96	234	169	52
6	46	117	73	230	213	195	124	44	142	160	19	18	69	103	233	168
7	90	200	178	133	179	249	64	243	201	56	112	106	155	35	110	95
8	7	157	97	31	161	4	122	70	174	41	91	172	1	204	186	183
9	164	80	205	141	159	17	11	60	92	158	107	40	85	115	229	167
10	192	93	185	86	255	146	0	239	203	162	13	68	6	214	37	2
11	48	182	25	30	74	8	57	53	152	219	39	130	59	119	165	81
12	246	113	197	212	235	208	237	109	9	72	116	36	171	61	189	227
13	184	65	47	209	29	75	78	23	240	242	51	250	98	14	166	71
14	32	121	175	144	202	27	215	153	194	79	118	140	245	101	216	126
15	43	108	87	104	42	16	49	15	76	218	241	63	131	211	26	24
16	224	149	127	252	99	139	62	206	50	253	248	190	247	33	163	94

3.3 SHA-256 Algorithm

SHA-256 hash function [29] is a method to create small digital "fingerprints" from any type of data. For messages with a length less than 2^{64} bits, SHA-256 algorithm will generate a 256-bit hash called message digest. This summary is equivalent to an array with a length of 32 bytes, which is usually represented by a hexadecimal string with a length of 64.

The ciphertext generated from the encrypted image is decomposed into n blocks, so the whole algorithm needs to complete n iterations, and the result of n iterations is the final hash code. Figure 2 shows the process of converting ciphertext into hash code.

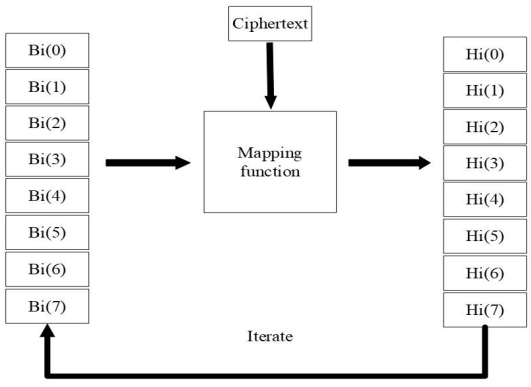


Figure 2: The ciphertext conversion to hash code

The components of SHA-256 include constant initialization, information preprocessing and logical operation.

Constant initialization: the SHA-256 algorithm uses 8 initial hash values and 64 hash constants. These initial hash values are derived from the first 32 bits of the decimal part of the square root of the first 8 prime numbers in the natural number. The 64 hash constants are derived from the first 32 bits of the decimal part of the cube root of the first 64 prime numbers in natural numbers.

Information preprocessing: the preprocessing phase of SHA-256 algorithm includes message filling, message separation and hash initial value setting, so that the whole message meets the specified structure. The preprocessing of information is divided into two steps: additional padding bits and additional length value.

Step 1. Additional padding bits. Fill in the end of the message so that the remainder of the message length after 512 modulo is 448.

Step 2. Additional length value. The additional length value is to add the length information of the original data (the message before **Step 1** filling) to the message that has been filled.

Logical operation: all operations involved in SHA-256 hash function are logical bit operations. In the hash calculation stage, the hash function is used to combine the data blocks to be processed and related constants, and perform multiple iterations to generate a series of hash values until all data blocks are processed.

3.4 IPFS Blockchain Storage

IPFS is the InterPlanetary File System, which is a distributed network transmission protocol for storing and sharing files. Blockchain is a distributed database and shared ledger. Blockchain was born to achieve decentralization, reach a consensus without a central organization, and jointly maintain an account book. Its design motivation is not for high efficiency, low energy consumption or scalability (if high efficiency, low energy consumption and scalability are pursued, centralized program may be a better choice). IPFS works together with blockchain, which can supplement two major defects of blockchain:

1) Blockchain has low storage efficiency and high cost. 2) Cross-chain requires cooperation among all chains, which is difficult to coordinate.

Usually, when uploading data by blockchain, it is widely adopted that only the hash value is stored in blockchain, and the information needed to be stored is stored in centralized database. In this way, storage has become a short board in decentralized applications and a weak link in the network. However, the emergence of IPFS puts forward a solution: using IPFS to store file data, and placing the only permanently available IPFS address in the blockchain transaction, without having to put the data itself in the blockchain. Figure 3 shows the process of data storage under the cooperation of IPFS and blockchain. It can be seen from the figure that with the combination of IPFS and blockchain greatly reduces the storage cost and improves the storage efficiency by using IPFS to process a large number of files, and storing the constant and permanent IPFS index into the blockchain.

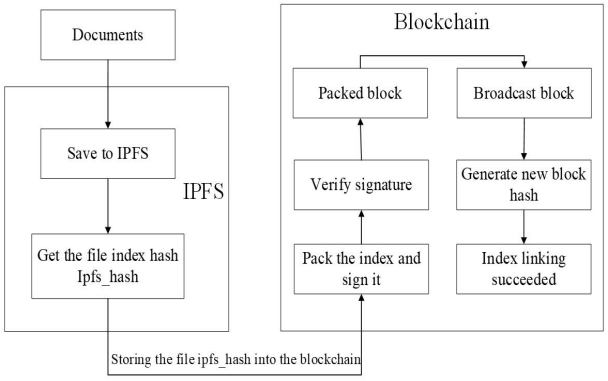


Figure 3: IPFS and data storage process

4 The Proposed Scheme

4.1 System Model

Figure 4 shows the system model of image tamper-proof encryption scheme based on blockchain and Lorenz hyperchaotic S-box. By combining Lorenz hyperchaotic and S-box image encryption scheme, the off-chain tamper-proof protection of images is realized, and the ciphertext image hash is automatically stored and tampered by intelligent contract. Dual tamper-proof detection ensures the security of image data to the greatest extent.

As shown in Figure 4, the proposed scheme firstly encrypts the original image, generates the corresponding ciphertext image hash by SHA-256 algorithm, then uploads the hash value to the blockchain, automatically stores the ciphertext image hash to IPFS by calling smart contract, and returns the storage address of IPFS (ipfs_hash) and stores it in the blockchain. After the storage is completed, the smart contract shows that the transaction is successfully uploaded. When tampering is detected, the smart contract calls ipfs_hash to obtain the ciphertext image hash code stored in IPFS, and compares it with the ciphertext image hash to be detected. If there is no tampering, it can be decrypted according to the corresponding image decryption algorithm.

4.2 Lorenz Hyperchaotic S-box Image Encryption

Figure 5 is the processing flow chart of image encryption algorithm based on Lorenz hyperchaotic S-box.

As shown in Figure 5, the image encryption algorithm based on Lorenz hyperchaotic S-box is mainly divided into three parts:

1) Generation of chaotic sequences:

Step 1. According to the definition of Lorenz hyperchaotic system in Equation (2), set the parameters$a$=10, b=8/3, c=28, r=-1. Set the initial values of four variables x_0, y_0, z_0, w_0, and use the above Lorenz hyperchaotic system to iterate n times to obtain the chaotic sequences x_n, y_n, z_n, w_n.

Step 2. According to Equation (3), the chaotic sequence x_n, y_n, z_n, w_n is converted into an 8-bit integer sequence X, Y, Z, W.

Step 3. Using sequence Y and W to perform XOR operation to obtain chaotic sequence T: T=$Y \oplus W$, where T=$[t(1), t(2), ..., t(256)]$.

2) Generation of S-box:

Step 1. Using sequence X and Z to perform XOR operation to obtain chaotic sequence S: S=$X \oplus Z$.

Step 2. 256 different numbers are randomly selected from the elements of $S$to form an S-box of size 16×16, S=$[s(1), s(2), ..., s(256)]$.

3) Generating an encrypted image according to the image encryption algorithm based on Lorenz hyperchaotic S-box:

Step 1. Input the original image I with size $M \times N$, The plaintext pixel values are extracted, and its two-dimensional matrix is converted into one-dimensional pixel sequence P, P=$[p(1), p(2), ..., p(L)]$.

Step 2. The ciphertext pixel value sequence $P^{'}$=$[p^{'}(i)]$ is generated by a round of diffusion operation between P and chaotic sequence T. When i=1, $p^{'}(1)$=$mod(p(1) + t(1) + skey, 256)$. When $i = 2, 3, ..., L$, calculate $p^{'}(i)$ as shown in Equation (4), where $p(i)$ is the i-th pixel value of the plaintext image, $p^{'}(i)$ is the i-th pixel value of the ciphertext after XOR encryption, $skey$ is a new parameter used as a key, mod is a modular division operation.

$$p'(i) = \mathrm{mod}(p(i) + t(i) + p'(i-1), 256) \quad (4)$$

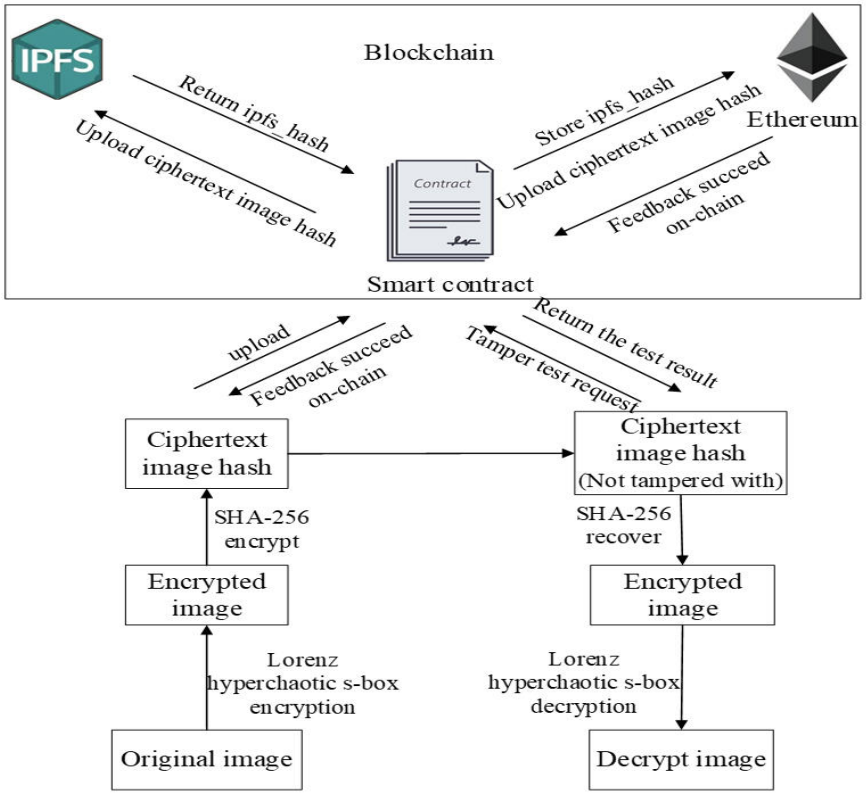


Figure 4: Image tamper-proof encryption system model based on blockchain

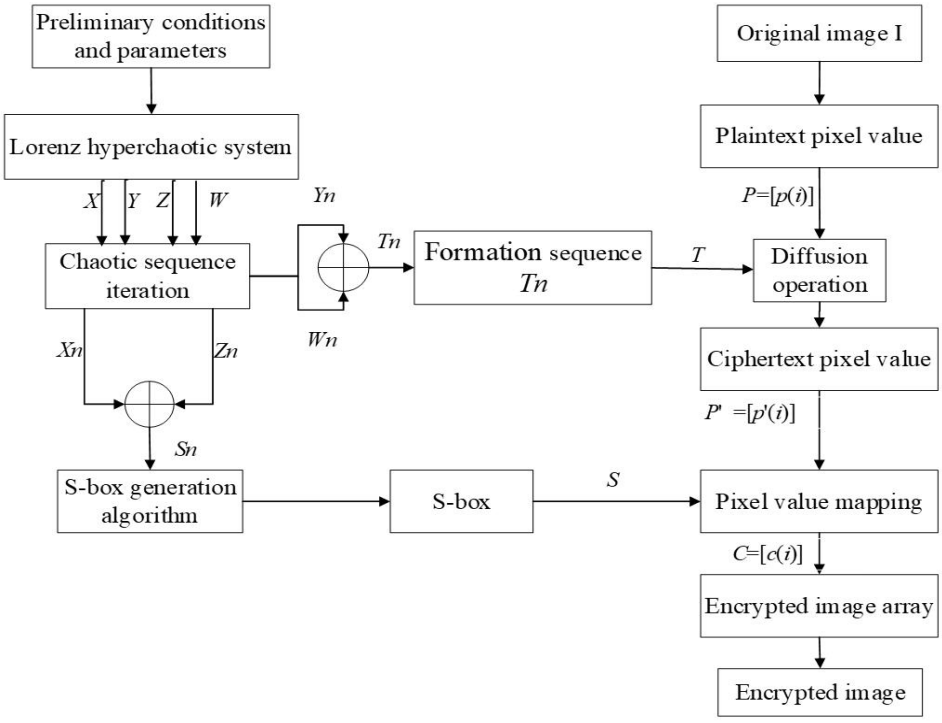


Figure 5: Flow chart of image encryption processing based on Lorenz hyperchaotic S-box

Step 3. The generated ciphertext pixel value sequence P' is mapped by Lorenz hyperchaotic S-box to realize two rounds of diffusion, when i=L, $c(L) = \mathrm{mod}(p'(L) + s(j) + skey, 256)$, where $j = double(p'(L))+1$, when i=$L-1, ..., 1$, the encryption formula is shown in Equation (5) to obtain the ciphertext image pixel sequence

C=[$c(i)$]. Where j=$double(p(i))+1$.

$$c(i) = \mathrm{mod}(p'(i) + s(j) + c(i+1), 256) \quad (5)$$

Step 4. After all password pixel values are determined, convert the one-dimensional ciphertext image pixel value sequence C into a two-dimensional matrix to obtain the encrypted image I'.

The operation step of decryption is the reverse process of the above encryption operation. First, obtain the tampered ciphertext image hash from the smart contract, then use SHA-256 algorithm to restore the ciphertext image to the encrypted image, and finally decrypt the decrypted image by using the reverse process of encryption operation steps.

4.3 Dual Tamper-proof Detection of Image

The image dual tamper-proof based on blockchain and Lorenz hyperchaotic S-box is mainly divided into two parts: on-chain and off-chain. Off-chain tamper-proof ensures that ciphertext images can effectively resist attacks such as differential cryptanalysis, chosen plaintext attack, statistical attack and noise attack during transmission through image encryption algorithm, and ensures that images are not tampered during transmission to the greatest extent. On-chain tamper-proof realizes the secure storage and tamper detection of ciphertext images through the tamper-proof characteristics of blockchain itself and the deployed smart contract.

When sending a transaction to the blockchain, if the trigger conditions of the smart contract are met, the preset logic will be automatically executed. When the verification nodes reach a consensus, the smart contract will be successfully executed. Because the smart contract will be triggered automatically, there is no need for the image service provider to always provide the service online. the user does not need to face the threat of malicious servers directly, and the fairness of the transaction among the image service provider, image users and image owners is guaranteed. In this paper, tamper-proof smart contract is used to store ciphertext image information and detect tampering.

The implementation steps of the tamper-proof smart contract are as follows: firstly, the ciphertext image hash is stored in the array **imghash**, and the smart contract is called to upload it to IPFS and blockchain respectively; Then the Get() function is used to call the hash of the ciphertext image to be detected and the corresponding hash of the ciphertext image stored in IPFS to calculate the similarity. When the calculated Hamming distance is greater than 0, the image data is tampered with; When the Hamming distance is 0, the image data has not been tampered with. The calculation method of Hamming distance is shown in Equation (6).

$$H(d_p, d_q) = \sum_{i=1}^{L} |d_{qi} - d_{pi}|, L \in [0, 255] \quad (6)$$

where $H(dp, dq)$ indicates the Hamming distance between the image data to be queried and the image data stored in IPFS, and dp is the hash value of the ciphertext image to be detected.

The implementation process of tamper-proof smart contract is shown in **Algorithm** 1.

Algorithm 1 Tamper-proof smart contract

Input: *Ciphertext image hash enchash*, *Data to be detected dp*

Output: *Tamper detection result*

```
Initialization Array imghash, Number of images m, Hamming distance sum, IPFS address array ipfsHash
for i→m do
    Upload enchash to imghash[i];
    Save imghash[i] into IPFS;
    Return memory address ipfsHash[i];
    Store ipfsHash[i] to Ethereum;
    ipfsHash[i] was successfully packaged into blocks.

    for j→m do
        Input hash of data to be detected dp;
        Get the corresponding ciphertext image hash imghash[j] on the IPFS according to ipfsHash[j] on the ethereum.
    end for
    if dp!=imghash[i] then
        sum += 1;
    end if
end for
if sum==0 then
    The data has not been tampered with, and return dp;
else
    The data has been tampered with, return False;
end if
RESET sum=0;
```

5 Experimental Results and Analysis

The experimental images are selected from the USC-SIPI test image database. Experimental hardware environment CPU: Intel(R) Core(TM) i7-1165G7 @ 2.80GHz, Display Card: Intel(R) Iris(R) Xe Graphics, Memory: 16GB. Build Ethereum private blockchain network on Ubuntu, the smart contract is written in solidity language, and the image encryption programming language is Python.

5.1 Analysis of Tamper-proof Performance of External Image

5.1.1 Comparison of Encryption and Decryption Effects

In order to demonstrate the universality of the algorithm, Lena_color with a size of 512×512 and Lena_gray with a size of 256×256 are selected as standard test images in the experiment. Figure 6 shows the image encryption result and the decrypted result after extracting the ciphertext image from the blockchain.

As can be seen from Figure 6, the encrypted image is completely confused and unrecognizable, and the decrypted image is no different from the original image. Therefore, the encryption effect achieved by the proposed encryption scheme is feasible.

5.1.2 Encryption and Decryption Efficiency Analysis

Encryption speed is very important for the practicability of image encryption algorithm. Table 2 shows the average encryption and decryption time data of images of different sizes, colors and gray scales.

As can be seen from Table 2, the improved image encryption algorithm in this paper has a fast encryption speed, and it takes an average of 1.216s to encrypt a 256×256 color image and 2.479s to encrypt a 512×512 color image. Figure 7 is a line chart showing the time-consuming encryption and decryption of color Lena images with different sizes.

It can be seen from the chart that the encryption time increases with the increase of image size. With the increase of image size and features, the generated image matrix becomes larger, and the corresponding calculation of diffusion process will also become time-consuming. If the code is further optimized, the speed will be further improved.

5.1.3 Statistical Attack Analysis

1) Histogram analysis
The ideal encrypted image histogram usually has a uniform frequency distribution and does not provide any useful statistics to the attacker [19,22]. For high security image encryption algorithm, the encrypted image must have uniformly distributed histogram. Figure 8 shows the histogram analysis results of Lena_color and Lena_gray images respectively.

   As can be seen from Figure 8, the plaintext image histogram is not uniform, while the encrypted image histogram is almost flat as the distribution of random data. Therefore, the encryption scheme completely hides the pixel distribution information of the original image and can resist statistical attacks.

2) Correlation analysis
High-security encryption algorithms must destroy the correlation between adjacent pixels in the image [22]. The correlation coefficient is an indicator to measure the correlation between adjacent pixels. The smaller the absolute value of the correlation coefficient is, the lower the correlation between adjacent pixels is. The coefficient of correlation adjacent pixels γ_{xy} is calculated as follows:

$$\gamma_{xy} = \frac{Conv(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

where,

$$Conv(x,y) = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)][y_i - E(y)]$$

$$D(x) = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)]^2$$

$$E(x) = \frac{1}{n}\sum_{i=1}^{n}x_i$$

where x, y represents the values of two adjacent pixels randomly selected in horizontal, vertical and diagonal directions, n represents the logarithm of adjacent pixels randomly selected, and γ_{xy} is the correlation coefficient between the original image and its corresponding cryptographic image.

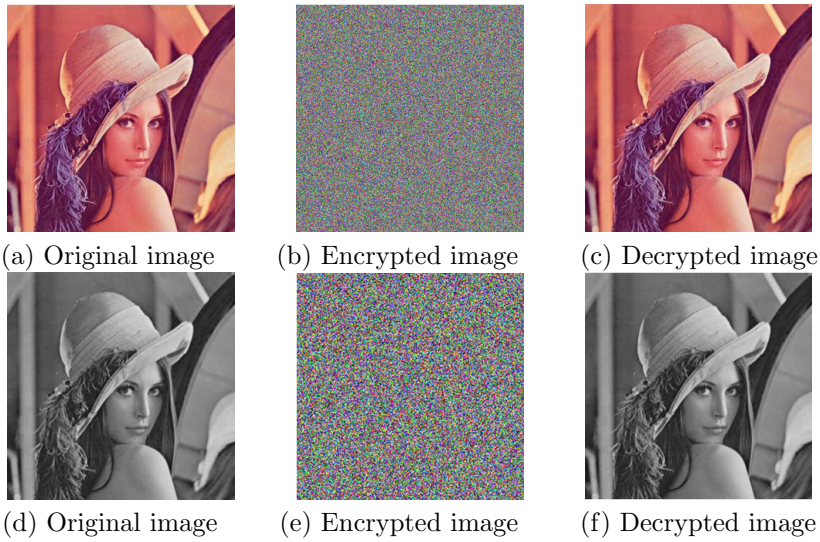
In this paper, 3000 pairs of adjacent pixels were randomly selected to calculate the correlation coefficient of the encrypted image as shown in Figure 9 from the pixel pairs along a certain direction (horizontal, vertical or diagonal).

As can be seen from Figure 9, the pixel points of the encrypted image are evenly distributed, and the correlation between adjacent pixels of the cipher image is much lower than that of the ordinary image. Therefore, the proposed encryption scheme has good performance in resisting statistical analysis attacks.

3) Information entropy analysis
Information entropy [24] is a common indicator to judge the randomness of information sources. The calculation of information entropy is shown in Equation (7):

$$H(s) = -\sum_{i=1}^{L}P(s_i)\log_2 P(s_i) \tag{7}$$

where si is the grayscale value of pixel, $P(si)$ represents the probability of occurrence of si. If $P(si) = 1/2n$, then the information source is completely random. The ideal value of information entropy is 8. The information entropy of an encrypted image should be as close to 8 as possible. Table 3 shows the comparison results of entropy of encrypted images between the proposed scheme and the existing scheme [1,10,16,24,29].



(a) Original image (b) Encrypted image (c) Decrypted image

(d) Original image (e) Encrypted image (f) Decrypted image

Figure 6: Encryption and decryption results of Lena_color and Lena_gray images

Table 2: encryption and decryption time

Image Size	Camrea_gray 225 × 225	Lena_gray 256 × 256	Lena_color 256 × 256	Lena_gray 512 × 512	Lena_color 512 × 512
Encryption time(s)	0.6721	0.7305	1.2159	1.7164	2.4792
Decryption time(s)	0.6982	0.8129	1.2327	1.8026	2.5307

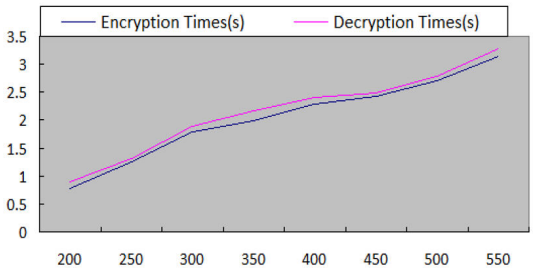


Figure 7: Encryption and decryption time analysis

Table 3: Information entropy analysis

Schemes	Lena_gray	Lena_color		
		R	G	B
Proposed	7.9975	7.9993	7.9994	7.9993
Ref. [24]	7.9027	7.9994	7.9993	7.9993
Ref. [10]	-	7.9993	7.9993	7.9994
Ref. [29]	-	7.9921	7.9917	7.9972
Ref. [16]	7.9971	-	-	-
Ref. [1]	7.9986	-	-	-

As can be seen from Table 3, the entropy value of the proposed scheme is very close to the ideal value 8, with good randomness. Therefore, the proposed scheme has stronger resistance to entropy-based attacks.

4) PSNR analysis

The Peak Signal to Noise Ratio (PSNR) [5] mainly examines the errors between corresponding pixels, it can be used to measure the quality of encryption. Given the size of the encrypted image I and the original image I′ of $M \times N$, the mean square error (MSE) is defined as shown in Equation (8). The PSNR calculation formula is shown in Equation (9).

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(I(i,j) - I'(i,j)\right)^2 \quad (8)$$

$$PSNR = 10\log_{10} \frac{(2^n - 1)^2}{MSE} \quad (9)$$

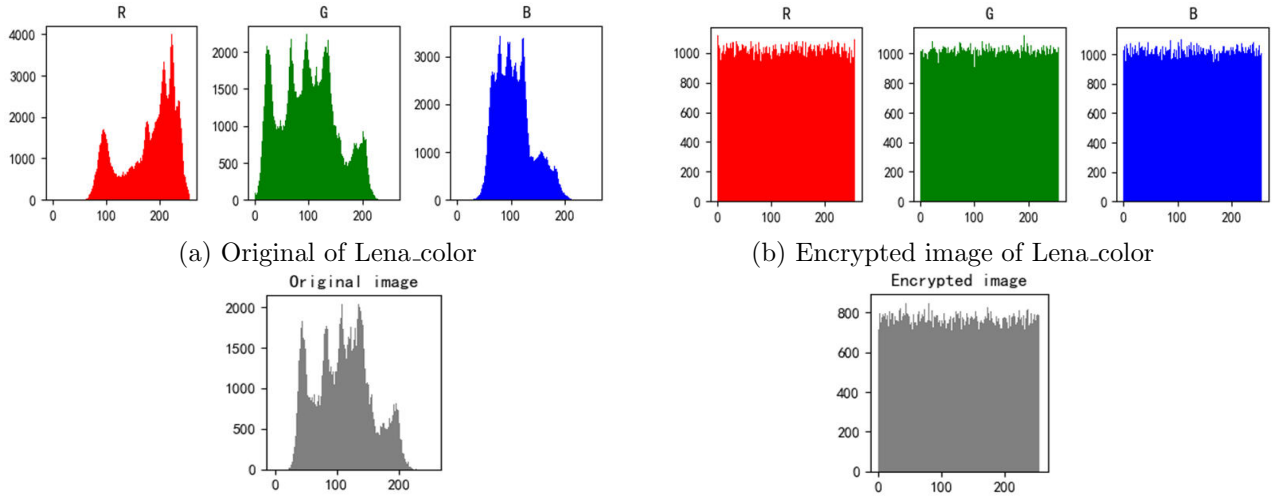
where M and N represent the width and height of the image respectively, and n represents the pixel number. The larger the PSNR value, the smaller the distortion, the smaller the gap between the two images, and the worse the encryption effect.

Table 4 shows the compares the PSNR value of the proposed scheme with that of the existing encryption scheme [5, 28].

Table 4: Comparison of PSNR values

Schemes	Lena_color	Lena_gray
Proposed	7.8892	9.2053
Ref. [5]	8.6234	9.5399
Ref. [28]	-	9.5301

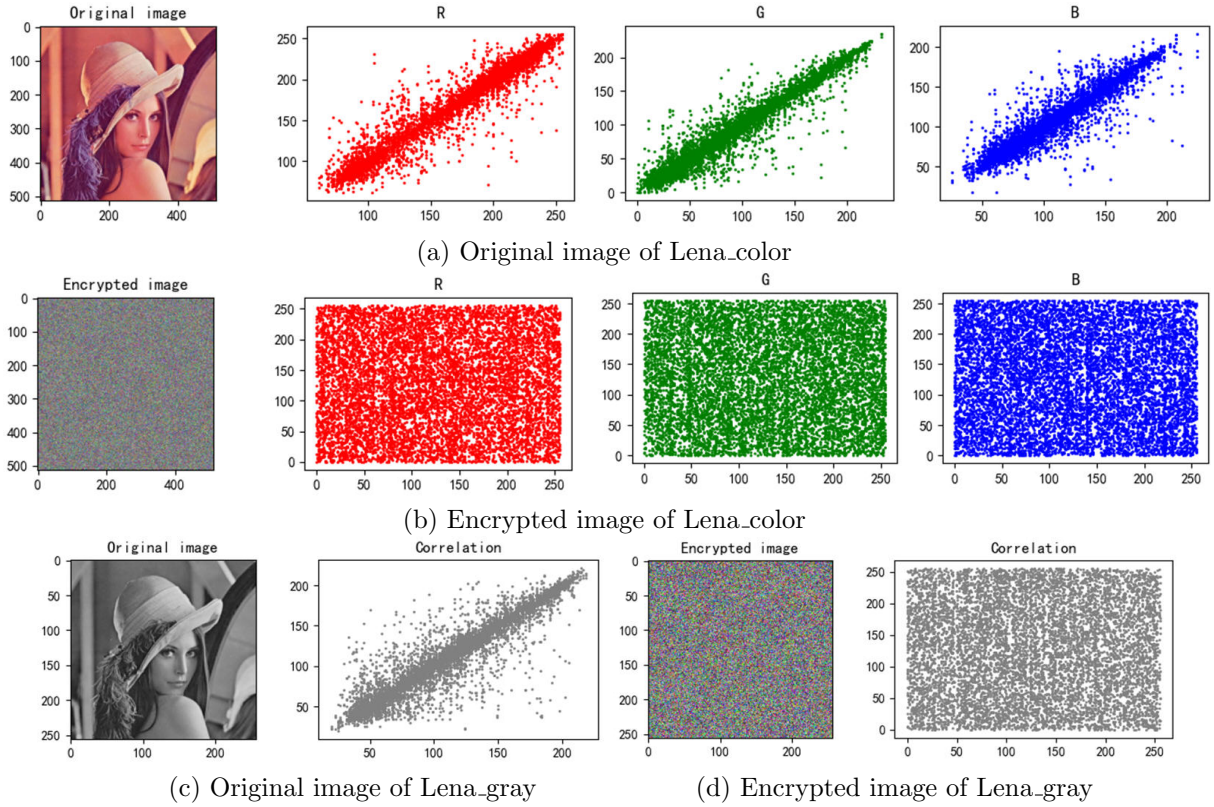
As can be seen from Table 4, the PSNR value of the proposed scheme is smaller than that of other



(a) Original of Lena_color (b) Encrypted image of Lena_color

(c) Original image of Lena_gray (d) Encrypted image of Lena_gray

Figure 8: Histogram analysis of Lena_color and Lena_gray images



(a) Original image of Lena_color

(b) Encrypted image of Lena_color

(c) Original image of Lena_gray (d) Encrypted image of Lena_gray

Figure 9: Pixel correlation analysis of the original image and the encrypted image

schemes, indicating that the error between pixels of the proposed scheme is small and the encrypted image quality is high.

5.1.4 Differential Attack Analysis

In order to resist differential attack [14,26], when a pixel change occurs in the plaintext image, the ciphertext image should have a large change, and the stronger the ability to resist differential attack. The ideal values of NPCR and UACI are 99.6094% and 33.4635%, respectively. The closer the calculation results of NPCR and UACI are to the ideal values, the stronger the encryption algorithm is in resisting differential attacks. NPCR and UACI are defined as Equations (10)-(12):

$$NPCR = \frac{\sum_{i,j} D(i,j)}{N \times M} \times 100\% \tag{10}$$

$$D(i,j) = f(x) = \begin{cases} 1, C_1(i,j) \neq C_2(i,j) \\ 0, otherwise \end{cases} \tag{11}$$

$$UACI = \frac{1}{N \times M} \frac{\sum_{i,j} (C_1(i,j) - C_2(i,j))}{255} \times 100\% \quad (12)$$

where N and M are the width and height of two encrypted images respectively, K' can be obtained by modifying a bit of key K, and $C1$ and $C2$ can be obtained by encrypting the same image with K and K'.

Table 5 shows the differences between the proposed scheme and the existing encryption scheme [1, 14, 22]. NPCR and UACI are used to quantify the differences between the two encrypted images.

Table 5: Sensitivity analysis of different encryption schemes with NPCR and UACI

Schemes	Lena_gray	Lena_color		
		R	G	B
Proposed	99.598	99.596	99.611	99.610
	33.476	33.446	33.475	33.467
Ref. [14]	99.596	99.596	99.612	99.598
	33.454	33.492	33.441	33.479
Ref. [22]	99.631	-	-	-
	33.450	-	-	-
Ref. [1]	99.69	-	-	-
	34.45	-	-	-

As can be seen from Table 5, the NPCR and UACI values of the encryption scheme in this paper are very close to the ideal values for both color and gray images. Compared with the encryption scheme, this scheme adopts two rounds of pixel diffusion encryption and dynamic S-box pixel mapping, which makes the pixel change rate of the encrypted image extremely high when the key changes slightly. This shows that the scheme has good sensitivity to encryption keys and is more effective in resisting differential cryptanalysis.

5.1.5 Noise Attack Analysis

Generally, the image encryption algorithm must be robust enough to resist noise attacks in the actual scene [5, 24]. In this section, 512×512 Lena_color is used to test the robustness of the proposed scheme. Figure 10 shows the analysis results of noise attack after adding gaussian noise and salt-and-pepper noise of different degrees to Lena_color image.

As can be seen from the Figure 10, for adding different noise intensities, as the added noise density increases, the decrypted image becomes more blurred, but after decrypting the noise image, the proposed scheme can still obtain most of the information of the original image from the decrypted image. It shows that the proposed scheme has strong robustness and can resist noise attack.

5.1.6 Performance Analysis of Lorenz Hyperchaotic S-box

In order to be able to use the generated S-boxes in the encryption process, the performance of the S-boxes shown in Table 1 was tested, and the performance of the proposed S-boxes was compared with that of the existing schemes. The results are shown in Table 6.

In order to effectively resist linearity cryptanalysis attacks, the input and output values of s-box must have a high nonlinearity (NL) [17] relationship. By comparing the proposed scheme with the [17,23,26,27], it can be seen from table 6 that the mean nonlinear value of the proposed scheme is 107.3, higher than that of the [17,23,27]. Because [26] uses linear trigonometric transformation to construct S-boxes, which is of high complexity, its NL value is larger than the proposed scheme. The higher the nonlinearity, the better the performance of S-box against linear cryptanalysis attacks. Therefore, the proposed scheme has a good ability to resist linear cryptanalysis attacks.

SAC [27] requires that if a single $j-th$ bit in the input value x changes, the probability of the $i-th$ bit in the output ciphertext value changing should be 0.5. As can be seen from table 6, compared with [17, 23, 26, 27], the SAC value of S-box proposed by the proposed scheme is 0.502, which is very close to the ideal value.

According to the bit independence criterion (BIC) [23], when the k-th bit of the input data block changes, the i-th bit and the j-th bit of the output data block change independently. In order to measure this characteristic of S-box, the strict avalanche criterion (BIC-SAC) and the nonlinear bit independence criterion (BIC-NL) are introduced. The average scores of BIC-SAC and BIC-NL are 0.5 and 103.9 respectively. Comparing the proposed scheme with [17,23,26,27], the BIC-SAC and BIC-NL values of S-box proposed in the proposed scheme are 0.501 and 104.0, which respectively shows that the correlation between the output bits of S-box in the proposed scheme is very weak and has good security.

Linear probability (LP) [23] is to judge whether a cryptographic system has strong confusion and diffusion effects. Differential probability (DP) [23] is a differential cryptanalysis evaluation standard. The lower the LP of S-box, the higher the nonlinear mapping characteristics, and the stronger the resistance to linear cryptanalysis. The smaller the DP, the stronger the ability of S-Box to resist differential cryptanalysis. Comparing the proposed scheme with Ref. [17, 23, 26, 27], it can be seen from Table 6 that the LP and DP values of S-box in the proposed scheme are 0.133 and 0.039. The LP values are smaller than those of S-box in most literatures, and the DP values are basically the same as those of S-box in the comparison literature, which shows that the proposed scheme has obvious advantages in resistance to attacks of differential cryptanalysis and linear cryptanalysis.

To sum up, the S-box generated by this scheme performs well in nonlinearity, strict avalanche criterion, linear probability and differential probability. It can effectively resist linear cryptanalysis attacks, differential cryptanalysis attacks and linear cryptanalysis attacks, and has good security, which shows that the S-box generated by this scheme has good performance.

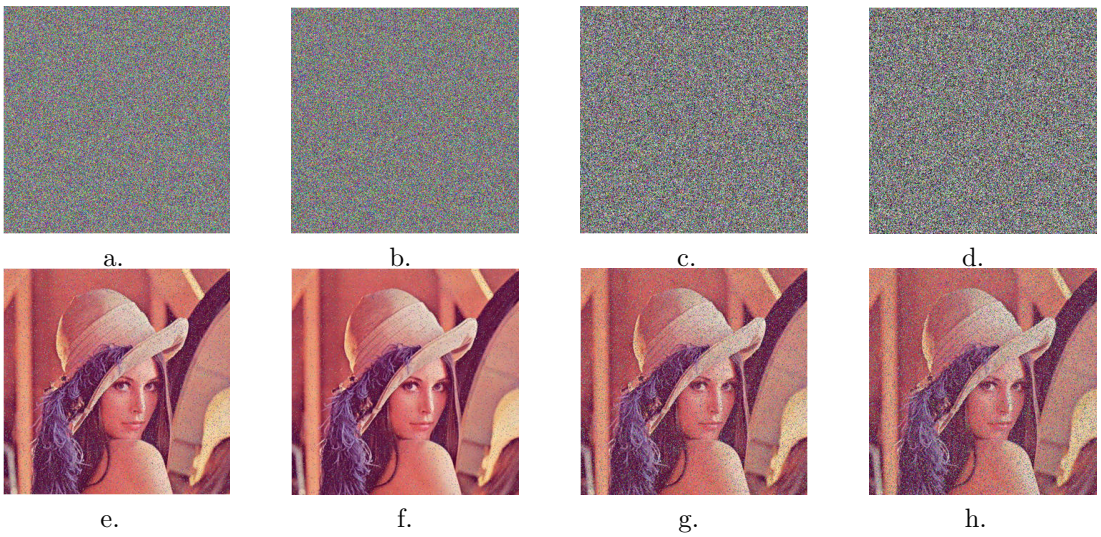


Figure 10: Lena_color image noise attack analysis: (a) add gaussian noise with 0.0005 density, (b)-(d) add 10%, 20% and 30% salt and pepper noise respectively, (e)-(h) is the decryption image corresponding to the noise image

Table 6: S-box performance comparison of different encryption schemes

S-box	NL	SAC	BIC-SAC	BIC-NL	LP	DP
Proposed	107.3	0.502	0.501	104.0	0.133	0.039
Ref. [23]	106.0	0.495	0.498	103.8	0.141	0.039
Ref. [17]	106.3	0.505	0.499	103.8	0.125	0.039
Ref. [27]	105.0	0.503	0.498	102.9	0.148	0.047
Ref. [27]	111.5	0.506	-	104.2	-	0.039

5.2 Tamper-proof Performance Analysis on Chain

On-chain tamper-proof achieves the secure storage and tamper-proof detection of ciphertext images through the tamper-proof feature of blockchain itself and the deployed smart contract. When a transaction is sent to the blockchain, the pre-set logic will be automatically executed if the trigger conditions of the smart contract are met, and the smart contract will be successfully executed when the verification nodes reach a consensus. The on-chain tamper-proof first generates an encrypted image through Lorenz hyperchaotic S-box encryption algorithm, then generates a ciphertext image hash code according to SHA-256, stores it in IPFS by smart contract and returns the storage address ipfs_hash. When the smart contract receives the tamper request, it calls the data stored in IPFS and the data verified by the request for tamper detection. Table 7 takes Lena_color image as the test image, and selects the encrypted image, the encrypted image with 50% clipping, the encrypted image with 30% noise addition and the encrypted image with 50% rotation respectively as the tampering test case, and Let's call each of them I, I_1, I_2, I_3.

The ciphertext image hash is stored in IPFS through the smart contract as the comparison data of the data to be detected, and the corresponding ipfs_hash is returned as: QmNkeepgJwwfay7FAzWfK1kWnXZgYexnVeoB8LtMp8V3Hr. In this experiment, four different kinds of data to be detected are selected for comparative test: the encrypted image that has not been tampered with, the encrypted image that has been clipped by 50%, the encrypted image that has been noisy by 30% and the encrypted image that has been rotated by 50% are selected to generate the ciphertext image hash as the data to be detected through SHA-256 algorithm, and the tamper detection request is sent to the blockchain. The smart contract obtains the corresponding ciphertext image hash through IPfs_hash, calculates the Hamming distance between the data to be detected and the ciphertext image hash, and returns the detection result. As can be seen from Table 7, if the encrypted image is attacked during transmission, the hash code of the generated data to be detected will change, and the encrypted image is tampered through the tamper detection of the smart contract.

6 Conclusions

In this paper, an image tamper-proof encryption scheme based on blockchain and Lorenz hyperchaotic S-box is proposed, which has the characteristics of high security, effectiveness, good response to differential cryptanalysis and image tamper detection, and realizes dual tamper-proof of encrypted image on-chain and off-chain. In the

Table 7: Tamper-proof performance tests

The test case	Encrypted image hash code	detection result
I	3ef71b40f6bbddddc5f4b69c125281a23958923a4b3b642f68d8fa437e948fbd	True
I_1	5511613e7d094337fc5b77aac4c718d2a6b48f4a63df74b0478937bdba4e0699	False
I_2	f0a41d8e8cf379dbbdfc43169f34851ed452b3581e72c6654f2e290caf4e1b20	False
I_3	216a9123ad729d4d418599b05cfcc2327c26aea714fafa2d4d34421aabacec5e	False

proposed scheme, The Lorenz hyperchaotic S-box is used for two rounds of pixel diffusion to ensure the security of the image off-chain. Combining blockchain, IPFS and smart contract technology, distributed storage and automatic tamper detection of image data on-chain are realized. The experimental results show that the proposed scheme has better security than the existing image protection methods, and can effectively resist various attacks such as statistical attacks, differential cryptanalysis attacks, noise attacks, etc. The automatic storage and tamper detection on the smart contract further improve the security of the ciphertext image. However, the proposed scheme is still insufficient in the efficiency of encryption and tamper-proof, and it is not very strong in anti-cropping attack. In future work, we will consider combining homomorphic encryption algorithm to further improve the security and tamper-proof efficiency of image encryption.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61862041).

References

[1] R. S. Acharya, M.and Sharma, "A novel image encryption based on feedback carry shift register and blockchain for secure communication," *International Journal of Applied Engineering Research*, vol. 16, no. 6, pp. 466–477, 2021.

[2] B. M. Alshammari, R. Guesmi, T. Guesmi, *et al.*, "Implementing a symmetric lightweight cryptosystem in highly constrained iot devices by using a chaotic s-box," *Symmetry*, vol. 13, p. 129, 2021.

[3] Z Azimi and S. Ahadpour, "Color image encryption based on dna encoding and pair coupled chaotic maps," *Multimedia Tools and Applications*, vol. 21, pp. 1727–1744, 2020.

[4] L. Q. Huang, S. T. Cai, X. M. Xiong, *et al.*, "On symmetric color image encryption system with permutation-diffusion simultaneous operation," *Optics and Lasers in Engineering*, vol. 115, pp. 7–20, 2019.

[5] L. Q. Huang, W. J. Li, X. M. Xiong, *et al.*, "Designing a double-way spread permutation framework utilizing chaos and s-box for symmetric image encryption," *Optics Communications*, vol. 517, p. 128365, 2022.

[6] N. A. Khan, M. Altaf, and F. A. Khan, "Selective encryption of jpeg images with chaotic based novel s-box," *Multimedia Tools and Applications*, vol. 80, pp. 9639–9656, 2021.

[7] P. W. Khan and Y. Byun, "A blockchain-based secure image encryption scheme for the industrial internet of things," *Entropy*, vol. 22, no. 2, p. 175, 2020.

[8] K. A. Kumar and A. Anjum, "A chaos maps based method using encryption scheme for securing dicom images: A comparative analysis," *International Journal of Electronics and Information Engineering*, vol. 12, pp. 128–135, 2020.

[9] R. P. Li, "Fingerprint-related chaotic image encryption scheme based on blockchain framework," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30583–30603, 2021.

[10] W. Y. Li, Y. G. Zhu, L. Tian, *et al.*, "An image encryption scheme based on lorenz hyperchaotic system and rsa algorithm," *Security and Communication Networks*, vol. 2021, p. 5586959, 2021.

[11] Y. F. Li, Y. W. Wang, J. Wan, *et al.*, "Privacy protection for medical image management based on blockchain," in *International Conference on Database Systems for Advanced Applications*, pp. 414–428, 2021.

[12] Z. Li, C. G. Peng, W. J. Tan, *et al.*, "An efficient plaintext-related chaotic image encryption scheme based on compressive sensing," *Sensors*, vol. 21, no. 3, p. 758, 2021.

[13] L. Liu, Y. H. Lei, and D. Wang, "A fast chaotic image encryption scheme with simultaneous permutation-diffusion operation," *IEEE access*, vol. 8, pp. 27361–27374, 2020.

[14] X. B. Liu, D. Xiao, and Y. P. Xiang, "Quantum image encryption using intra and inter bit permutation based on logistic map," *IEEE Access*, vol. 7, pp. 6937–6946, 2018.

[15] Y. H. Liu, J. B. Zhang, and J Zhan, "Privacy protection for fog computing and the internet of things data based on blockchain," *Cluster Computing*, vol. 24, no. 2, pp. 1331–1345, 2021.

[16] Q. Lu, C. X. Zhu, and X. H. Deng, "An efficient image encryption scheme based on the lss chaotic map and single s-box," *IEEE Access*, vol. 8, pp. 25664–25678, 2020.

[17] Q. Lu, C. X. Zhu, and G. J. Wang, "A novel s-box design algorithm based on a new compound chaotic system," *Entropy*, vol. 21, p. 1004, 2019.

[18] S. K. Mousavi, A. Ghaffari, S. Besharat, *et al.*, "Security of internet of things based on cryptographic algorithms: a survey," *Wireless Networks*, vol. 27, no. 2, pp. 1515–1555, 2021.

[19] Y. Pourasad, R. Ranjbarzadeh, and A. Mardani, "A new algorithm for digital image encryption based on chaos theory," *Entropy*, vol. 23, p. 341, 2021.

[20] G. Qu, X. F. Meng, Y. K. Yin, *et al.*, "Optical color image encryption based on hadamard single-pixel imaging and arnold transformation," *Optics and Lasers in Engineering*, vol. 137, p. 106392, 2021.

[21] M. Shen, Y. W. Deng, L. H. Zhu, *et al.*, "Privacy-preserving image retrieval for medical iot systems: A blockchain-based approach," *IEEE Network*, vol. 33, no. 5, pp. 27–33, 2019.

[22] M. Z. Talhaoui and X. Y. Wang, "A new fractional one dimensional chaotic map and its application in high-speed image encryption," *Information sciences*, vol. 550, pp. 13–26, 2021.

[23] X. Wang, Ü. Çavuşoğlu, S. Kacar, *et al.*, "S-box based image encryption application using a chaotic system without equilibrium," *Applied Sciences*, vol. 9, no. 4, p. 781, 2019.

[24] X. Y. Wang, L. Feng, and H. Y. Zhao, "Fast image encryption algorithm based on parallel computing system," *Information Sciences*, vol. 486, pp. 340–358, 2019.

[25] J. Y. Yu, C. Li, X. Song, *et al.*, "Parallel mixed image encryption and extraction algorithm based on compressed sensing," *Entropy*, vol. 23, no. 3, p. 278, 2021.

[26] A. H. Zahid, M. Ahmad, A. Alkhayyat, *et al.*, "Efficient dynamic s-box generation using linear trigonometric transformation for security applications," *IEEE Access*, vol. 9, pp. 98460–98475, 2021.

[27] J. L. Zhang, Y. Q.and Hao and X. Y. Wang, "An efficient image encryption scheme based on s-boxes and fractional-order differential logistic map," *IEEE Access*, vol. 8, pp. 54175–54188, 2020.

[28] Y. Zhang, "The fast image encryption algorithm based on lifting scheme and chaos," *Information sciences*, vol. 520, pp. 177–194, 2020.

[29] F. X. Zhao, M. Z. Lin, W. Kun, *et al.*, "Color image encryption via hénon-zigzag map and chaotic restricted boltzmann machine over blockchain," *Optics & Laser Technology*, vol. 135, p. 106610, 2021.

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.

Tian Li 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 Minnan Normal University in Zhangzhou, Fujian, China in 2018. Her research interests include network and information security, multimedia security and blockchain technology.

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.

IoT Malware Threat Hunting Method Based on Improved Transformer

Yaping Li and Yuancheng Li
(Corresponding author: Yuancheng Li)

School of Control and Computer Engineering, North China Electric Power University
No. 2 Beinong Road, Changping District, Beijing, China
Email: ncepua@163.com

(Received Aug. 8, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)

Abstract

With the continuous improvement of computer performance and network transmission speed, more and more Internet of Things (IoT) devices are widely used. As a result, they have gradually become the main target of network attacks. This article proposes an IoT malware threat-hunting method based on the improved Transformer. Based on the Transformer model, the method reduces the computational complexity of the attention layer through additive attention. It replaces the residual connection and normalization module through the residual weight parameters to construct an improved Transformer network model, which makes the model can fully extract the features of the input sequences. At the same time, the network converges faster, and the classification accuracy is higher. Finally, the network is trained using the API Call and OpCode datasets. The experimental results show that the proposed method can detect malware types faster and obtain an accuracy rate of 98%.

Keywords: Additive Attention Mechanism; Improved Transformer; Malware; Residual Weight; Threat Hunting

1 Introduction

Malware is a malicious program that interrupts or damages the computer system without the user's permission, thereby affecting the normal use of the user. With the improvement of computer performance and the wide application of smart devices, malware has new communication channels. Attackers take advantage of the vulnerability of IoT devices to illegally enter the system and steal users' crucial information, which has brought great harm to the security of cyberspace and seriously affected users' normal access to the network. According to the survey of relevant researchers, in 2019, more than 1.8 million new malicious samples were discovered on the 360 security mobile terminals, with an average of more than 5,000 new malicious applications every day, and there are many variants of the same type of malware, showing a gradual growth trend [24,25]. They are designed to destroy users' computer systems and steal users' privacy, posing a serious threat to users' information security and property security [27].

Threat hunting refers to the process of actively and continuously searching the network environment for malware or malicious application that can bypass detection or cause harm [11]. Malware threat hunting methods can be divided into static analysis and dynamic analysis. Static analysis is to detect malicious applications without execution. Static features usually include Bytecodes, OpCodes, API Calls, control flow graphs, and strings. In contrast, dynamic analysis refers to executing a malicious application in a controlled environment and observing its set of behaviors. Usually, static analysis is widely used [5]. R. Mirzazadeh [14] builds a similarity graph based on OpCodes of applications to detect malware variants. Igor Santos [18] is characterized by the occurrence frequency of OpCodes to detect malware under different classifiers. Different from the previous literature, Jueun Jeon [10] is a dynamic analysis method. The author executes the malware in a virtual machine for 5 minutes and then extracts various behavior information such as memory, network, and process generated by it into excel. The feature information is integrated according to the frequency of each behavior, and finally, the dynamic detection of malware is completed through the Convolutional Neural Network.

Machine learning techniques have a wide range of applications in malware threat hunting. Jianwen Fu *et al.* [4] proposed to visualize malware as a picture, extract features from the picture, and then implement detection on Random Forest, K-Nearest Neighbor algorithm, and Support Vector Machine respectively, and achieved a detection accuracy of 97.47%. Zhang *et al.* [29] firstly used n-gram technology to process the system call data of the application and used rough set theory to eliminate redundant features to implement detection on Support Vector Machines. Mahdi Rabbani *et al.* [17] applied a particle swarm optimization-based probabilistic neural network for the detection and recognition of malicious be-

haviors. Rafiqul Islam *et al.* [8] presented the first classification method integrating static and dynamic features into a single test, which improved on previous results based on individual features and reduced by half the time needed to test such features separately. Daniel Morato *et al.* [15] presented a ransomware detection algorithm based on the analysis of network traffic and ran experiments using more than 50 samples from 19 different ransomware families, which greatly reduced detection time. Song [23] proposed a malware detection method based on Random Forest, and the detection accuracy was only 92.4%. Reference [26] uses the Android application programming interface call sequence and permission as features and uses the ensemble learning method to detect the malware.

In recent years, deep learning algorithms have also been widely used in the field of malware detection. Convolutional Neural Network (CNN) has certain advantages in judging image similarity due to the existence of local receptive fields. Yang Ming *et al.* [13] proposed a SIC (Sim HashImage-CNN) model, which firstly converted the malware program into a disassembler, generated a grayscale image, and then encoded it with the Sim Hash algorithm to obtain the feature vector of the malware, and finally used a convolutional neural network to classify malware with an accuracy rate of 96.7%. Kolosnjaji B [12] added a recurrent layer to CNN to identify the malicious application based on the system call sequence. Due to the limitation of a fixed number of input neurons in the feed-forward neural network of CNN, the accuracy is not high in detecting variable-length sequences of [9]. Recurrent Neural Networks (RNN) have great advantages in classifying time series data. Tobiyama *et al.* [20] proposed to use RNN to extract the API Call log sequences within 5 minutes, and then input them into CNN, achieving 96% classification accuracy. Pascanu *et al.* [16] built an API call language model through RNN, then generated a fixed-length feature vector, and then used a multi-layer perceptron to classify the feature vectors. Reference [6] proposed using Long Short-Term Memory network (LSTM) to detect Android malware. The author used ByteCodes extracted from Android applications as features, encoded them into feature vectors through the one-hot method, and then input them into the LSTM network. Finally, 95.3% accuracy is obtained on large datasets. Considering the limitations of a single feature, the reference [3] used API calls, internal function calls, and other attack traces as features, and then conducted experiments on LSTM network, which successfully improved the detection accuracy, with an accuracy rate of 98.5%. Reference [1] extracted features through CNN, and then input this primary feature into LSTM for high-level feature extraction, but the detection accuracy was not high. Burnap *et al.* [28] combined RNN and LSTM for a detection accuracy of 98% and an error rate of 1.41%. Deep Belief Network (DBN) is a deep neural network that has been gradually applied in malware detection tasks in recent years [7, 19, 30]. Hou S25 proposed using DBN for malware detection. The author extracted the dynamic and static features of malware from different files, and then trained and fine-tuned the network to optimize the detection accuracy of the model. Although many deep learning techniques have achieved good results in the field of malware detection, they also have certain limitations. For example, when the depth of the network increases, the deep belief network may face the problem of vanishing or exploding gradients, making it difficult to converge. At the same time, although LSTM is suitable for processing variable-length time series data, the calculation at this moment is heavily dependent on the calculation results of the previous moment, which limits the parallel ability of the model and takes a long time.

Transformer [21] is a deep learning model based on complete attention mechanism, which is widely used in tasks such as natural language processing, computer vision, and machine translation. It is different from RNN. RNN is trained in chronological order, while Transformer training is the simultaneous input of all time features into the network, and completely relies on the self-attention mechanism to describe the global dependence of input and output. Compared with RNN, it greatly improves classification efficiency. However, Transformer also has some shortcomings. For example, the computational complexity of its self-attention layer is large, which is the quadratic of the length of the input sequence. Meanwhile, when the network depth increases, the gradient is easy to disappear or explode, and the training time is long. To solve these problems of the Transformer, in this article, we use the additive attention mechanism [22] to replace the calculation of the attention value of the vanilla Transformer, and add residual weight parameters in the attention layer and the feed-forward neural network layer respectively, accelerating the convergence speed of the depth network.

In this article, we propose an Internet of things (IoT) malware threat hunting method based on the improved Transformer. The novelty of this method is that it combines the additive attention and residual weight parameters for the first time to construct an improved Transformer network model, which reduces effectively the computational complexity and detection time. Experiments are carried out on two kinds of datasets to verify the effectiveness of the proposed method. Specifically, the contributions of this paper are as follows.

1) We propose an Internet of things malware threat hunting method based on improved Transformer. Experiments are conducted on API Call and OpCode datasets respectively, and the results suggest that the proposed method outperforms the previous threat hunting model based on CNN and RNN in accuracy, precision, F1-Score, and recall rate.

2) Given the limitations of the Transformer, we use an additive attention mechanism to replace the attention value of vanilla Transformer. At the same time, a residual weight parameter is added in the attention layer and the feedforward neural network layer respectively, which alleviates the gradient disappear-

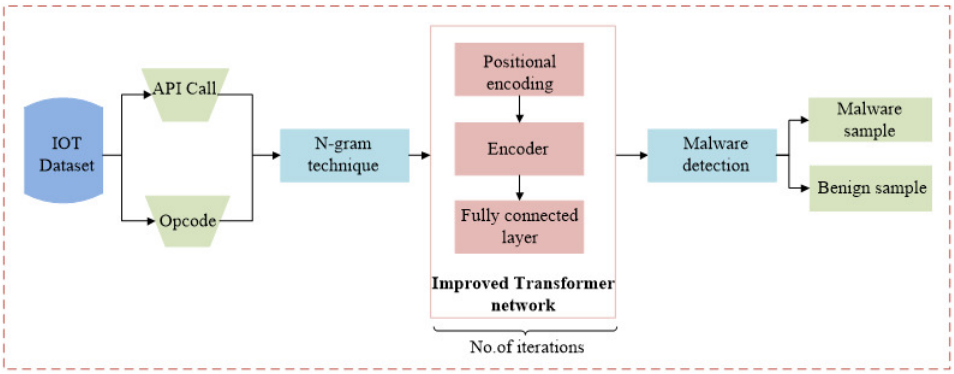


Figure 1: IoT threat hunting model based on improved Transformer

ance or explosion caused by the deep networks. Experiments results on OpCode and API Call files suggest that the proposed method can effectively reduce training cost, accelerate convergence, and improve detection accuracy to a certain extent.

2 Method

The proposed IoT threat hunting model based on improved Transformer consists of three stages, as presented in Figure 1. In the first stage, we collect malware API Call datasets and OpCodes datasets, each of which contains benign samples and malicious samples, then use the n-gram algorithm to convert the features of each sample into vector form, and finally use additive attention mechanism to replace the Transformer's multi-head attention module, and the residual weight is introduced to avoid the disappearance of the gradient, and an improved Transformer network is constructed. The network is trained through the interface call of the application program and OpCode features to build the IoT threat hunting model based on the improved Transformer.

2.1 Data Preparation and Preprocessing

In this article, we selected two kinds of datasets related to threat hunting, API Calls and OpCodes.

API Call: API Call represents the behavior of the malware for a while after installation, these activities are not arbitrary, they come from a limited set, and for the same operating system, these activities have the same meaning in different applications. The API Call dataset used in this article contains a total of 30,000 pieces of data, including 14,302 malicious samples and 15,698 benign samples. Each sample records application program interface call characteristics at different moments, such as CryptProtectData, FindWindow, listen,..., CryptUnprotectMemory. Each feature is firstly converted into a vector and then fed into the classification model.

OpCode: OpCode is an assembly language code that is obtained by decompiling the malware executable files with a decompilation tool. The OpCode dataset used in this article contains 9000 samples, of which 3500 are malicious and 5500 benign. Each sample contains features such as mov, push, add, ..., lea. Similarly, each feature must be vectorized and then input into the classifier.

Usually, a single activity can't infer that the entire sequence is malicious. Sometimes, normal activities combined in different ways can cause a malicious effect, so we need to find a method that can handle multiple activities, we can use word embedding technology to convert the calling behavior at each moment into vector form. One-hot encoding is difficult to achieve good results because it requires a very long vector to represent a feature, which is prone to dimensional disaster and cannot represent the semantic relevance of the context. The n-gram algorithm is a neural network method with high computational performance, that is, given a word, it can predict the occurrence probability of its surrounding words. For example, if n is set to 3, we define the three words before the center word and the three words after the center word as surrounding words, and the probability of these surrounding words appearing is the output of n-gram.

Taking the API Call dataset as an example, we select the application program interface call sequence from to for the first sample in the dataset, "CreateRemoteThreadEx, LoadStringW, recv, NtOpenKeyEx, CryptExportKey, CreateServiceA, CopyFileW, GetAdaptersInfo, GetNativeSystemInfo, WSAAccept". When we set the window size n=3, the vicinal features of the feature " NtOpenKeyEx " are "CreateRemoteThreadEx ", "LoadStringW", "recv", "CryptExportKey", " CreateServiceA", " CopyFileW ". The training goal of the n-gram model is to maximize the output probability of vicinal words. The input layer is the one-hot encoding of "NtOpenKeyEx" and the output layer is the prediction probability of the adjacent words of the central word, and the weight matrix from the input layer to the hidden layer is the embedding matrix we want. The principle of vectorizing API Calls through the n-gram algorithm is shown in Figure 2, in which "1" represents the position of the feature.

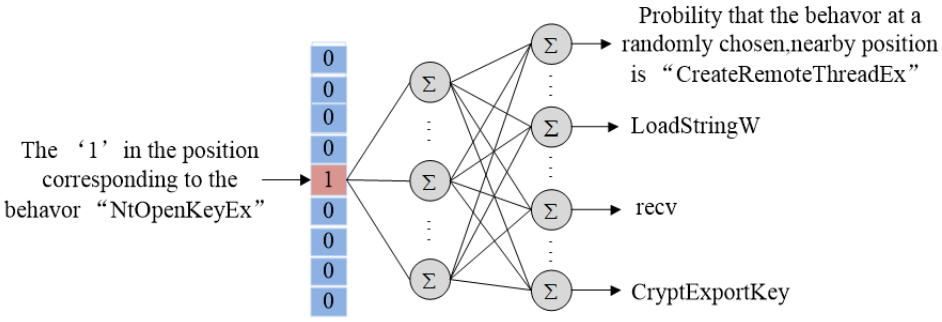


Figure 2: IoT threat hunting model based on improved Transformer

2.2 The IoT Threat Hunting Model Based on Improved Transformer

Transformer is a deep learning model following CNN and RNN. It has powerful sequence feature extraction capability and is suitable for malware detection. However, the computational complexity of its self-attention layer is too large, and when the network is deeper, gradients are prone to vanishing or exploding. To fully extract sequence features and detect malware types quickly, based on the Transformer architecture, we use additive attention and residual weights to construct an improved Transformer network and propose a threat hunting model based on the improved Transformer.

2.2.1 Transformer

Transformer has a good performance in natural language processing tasks. Similar to language models, Transformer can also use its powerful self-attention mechanism to extract the features of the input sequence, and output expected classification results. Transformer consists of encoder and decoder. In classification tasks, the encoder is generally used for feature extraction, and then the classification results are output through the softmax classifier. Because the dataset in this article contains two types of samples, the classification task is binary. So we only introduce Transformer's encoder. Each encoder consists of two modules, a multi-head attention module, and a fully connected feed-forward neural (FFN) network module. Residual connection and layer normalization are also used around the two modules. Specifically, the input matrix and the weight matrices W_i^Q, W_i^K and W_i^V perform linear operations respectively and map results to matrixes Q,K,V ,and then by softmax calculating the single-head attention, h-head attention values can be spliced to obtain the multi-head attention operation result, and the global representation of the input sequence is obtained, and then feed the result into a fully connected feed-forward neural network.The residual connection and normalization are performed near each module through $LayerNorm(x+Sublayer(x))$, where $Sublayer(x)$ refers to the multi-head attention layer or the feed-forward neural network layer. Calculated as follows:

$$\begin{aligned} head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \\ &= softmax(\frac{QW_i^Q \times KW_i^{K^T}}{\sqrt{d_k}})VW_i^V \\ MultiHead(Q,K,V) &= Concat(head_1, \ldots\ldots head_h)W^0 \\ FFN(Z) &= max(0, ZW_1 + b_1)W_2 + b_2 \end{aligned} \tag{1}$$

Where $W_i^Q \in R^{d_{model} \times d_k}$,$W_i^K \in R^{d_{model} \times d_k}$,$W_i^V \in R^{d_{model} \times d_v}$, and $W^O \in R^{d_{model} \times hdv}$,$d_k$ is the dimension of the matrix K,d_v is the dimension of the matrix$V$,d_{model} is the output dimension of sublayers and embedding layers in the model. In the calculation formula of the fully connected feed-forward neural network, the first layer uses ReLU as the activation function, and the second layer is a linear activation function, $Z$is the output matrix of the attention module.

2.2.2 Threat Hunting Model Based on Improved Transformer

The proposed threat hunting model uses an improved Transformer classifier to detect IoT malicious application and distinguish whether the input is malicious or benign. From the introduction of the Transformer network principle in the previous section, we know that the computational complexity of the Transformer attention layer is the quadratic of the sequence length, and the computational efficiency is low, which is not conducive to the detection of malware. At the same time, when the depth of the network increases, it is more difficult to converge. Thereby, we propose an improved Transformer model for threat hunting task. The network structure is shown in Figure 3.

(1)Input module

Since there is no recurrent or convolution in the model, the sequence relationship needs to be added to the sequence through positional encoding. The position encoding formula is as follows:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{2}$$

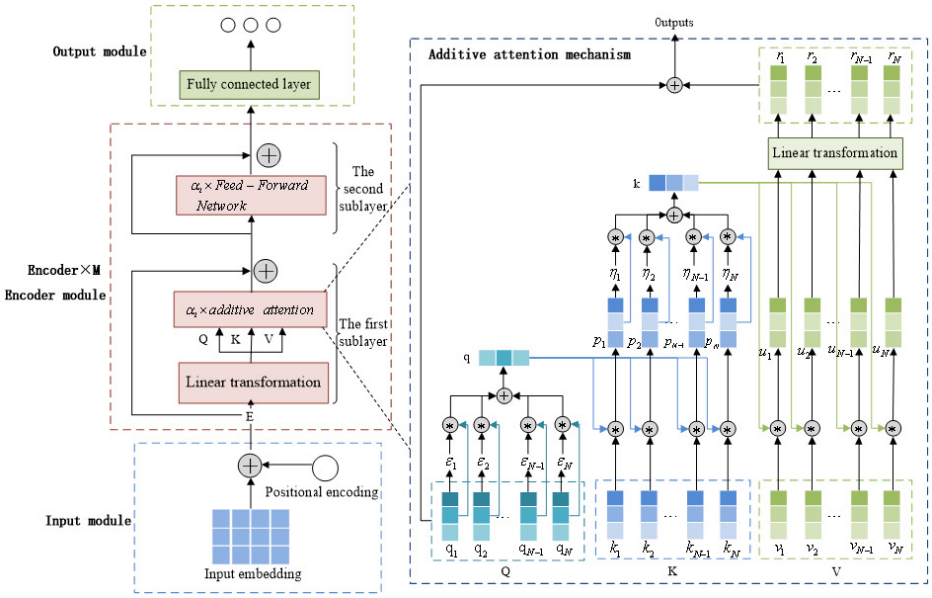


Figure 3: IoT threat hunting model based on improved Transformer

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}). \quad (3)$$

Where pos is the position, i is the dimension, $PE_{(pos,2i)}$and $PE_{(pos,2i)}$ represent position information when the sequence position is even and odd respectively.Then add the word and position embeddings of the sequence to get the representation E of the sequence, that is, the input of the model.

$$E = Po + X. \quad (4)$$

WherePo represents position embedding matrix, X is the vectorized word embedding matrix by n-gram.

(2) Encoder

We use M stacked encoders for feature extraction. This part can be considered to be composed of two sublayers, the first sublayer includes attention mechanism and residual connection, and the second sublayer includes a fully connected feed-forward neural network and residual connection.

Additive attention mechanism: First, the input matrix is linearly transformed into matrices Q,K ,V , and their sub-vectors are written as $Q = [q_1, q_2 \ldots\ldots q_N]$,$K = [k_1, k_2 \ldots\ldots k_N]$,$V = [v_1, v_2 \ldots\ldots v_N]$ respectively. Next, instead of performing dot product calculation among matrices in the vanilla Transformer, additive attention is used to model contextual information of the inputs. Specifically, the context information in Q is first compressed and summarized into a query vector q containing global information according to the following formula.

$$\varepsilon_i = \frac{exp(w_q^T q_i/\sqrt{d})}{\sum_{j=1}^{N} exp(w_q^T q_j/\sqrt{d})} \quad (5)$$

$$q = \sum_{i=1}^{N} \varepsilon_i q_i \quad (6)$$

Where w_q^T is the learnable parameter vector during training.

Then, the interaction between the global query vector q and matrix K is modeled by the formula, and another weight parameter η_i is obtained according to the calculation method of the previous attention weightε_i , then η_i and p_i are combined into the global matrix k with the ability to understand the global context. The calculation formula is as follows.

$$\eta_i = \frac{exp(w_k^T p_i/\sqrt{d}}{\sum_{j=1}^{N} exp(w_k^T p_j/\sqrt{d})} \quad (7)$$

$$k = \sum_{i=1}^{N} \eta_i p_i \quad (8)$$

Finally, to better learn the relationship among application program interface call behaviors at different moments, the global matrix k and each value vector of matrix V are multiplied by the formula $u_i = k * v_i$. Similar to the previous operation, the matrix u_i after interaction is linearly transformed to obtain a matrix R, and the matrix R and the matrix Q are added to form the calculation result of single-head attention. According to formula (1), the h-head attention values are spliced, and the multi-head attention output can be obtained. The computational complexity of the attention module is now reduced from quadratic to linear. It is worth noting that to further reduce the computational complexity, the Q and K matrices of different layers share the same transformation parameters.

Residual connection: After solving the quadratic complexity problem of attention calculation, considering that the signal is prone to gradient disappearance or explosion during deep network propagation, to alleviate this

phenomenon, we add a small architecture to the Transformer architecture with additive attention, that is, discarding the original residual structure and layer normalization. For each multi-head attention module or feed-forward neural network module, add a trainable parameter α that can be used to adjust the output of the current module [2], and then connect the residuals according to formula (9). This part is applied in both sub-layers of the encoder.

$$x_{i+1} = x_i + \alpha_i sublayer(x_i) \tag{9}$$

Where, α_i is the residual weight parameter that can be learned during training. Its initial value is 0. The network is mapped to an identity function. During training, the value of α_i will increase continuously and evolve into a suitable value at a certain moment. $sublayer(x_i)$ represents the calculation results of the above multi-head attention or the calculation results of the feed-forward neural network.

Fully connected feed-forward neural network (FFN): After the computation of the first sublayer is performed, the result is fed to the second sublayer of encoder. The layer includes fully connected feed-forward neural network and residual connection mentioned above. where the feed-forward network consists of two linear transformations with a ReLU activation in between.

$$FNN(x) = max(0, ZW_1 + b_1)W_2 + b_2 \tag{10}$$

Where Z is the output of the first sublayer, which is considered as the input of the feed-forward neural network, and W_1, W_2, b_1 and b_2 are the weight matrices and bias vectors of the two linear transformations, respectively.

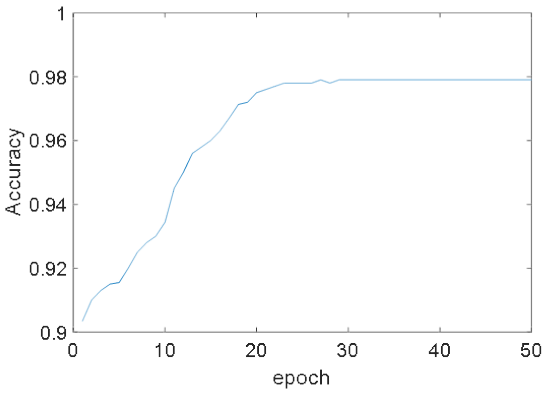
(3) Output module

The output module includes a fully connected layer and a softmax classifier. After the contextual information of the input sequence is extracted by the encoder of the model, the features are sent to the output module and output the application type through softmax.

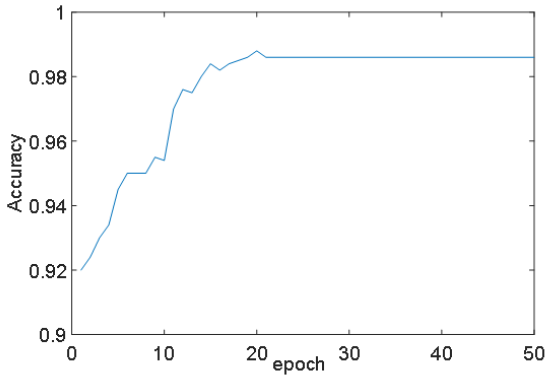
The position-encoded API Call and OpCode sequences are viewed as model input and sent to the improved Transformer network for training, and then the trained network is saved, and the test set is sent to the final model for malware detection, and the detection result of the sample is obtained, that is, benign or malicious.

3 Experiment and Results Analysis

To evaluate the proposed model, we conduct two sets of experiments. On the one hand, we evaluate the performance of our model in detecting malicious and benign IoT applications using common performance metrics, such as accuracy, precision, recall, and F1-Score. On the other hand, we compare the detection time of our model against other IoT threat hunting models based on deep learning, which proves the efficiency of the proposed method.



(a) Accuracy of different epochs on OpCode dataset



(b) Accuracy of different epochs on API Call dataset

Figure 4: Accuracy of different epochs on two kinds of datasets

Our experiments are performed in the environment of Python 3.7, PyTorch 1.6.0, and the device configuration includes NVIDIA TITAN RTX 2080Ti 24GB and CUDA 10.1 version.

3.1 Evaluation Index

We divide the two datasets into training set and test set according to the ratio of 4:1. The performance of the proposed model is evaluated using the 10-fold cross validation method. The following confusion matrix for binary classification task is used for performance quantification, as shown in Table 1.

Table 1: Confusion matrix

Actual redults Detection results	Actually mlicious	Actually benign
Detected as malicious	TP	FP
Detected as benign	FN	TN

1) Accuracy: Accuracy indicates the proportion of be-

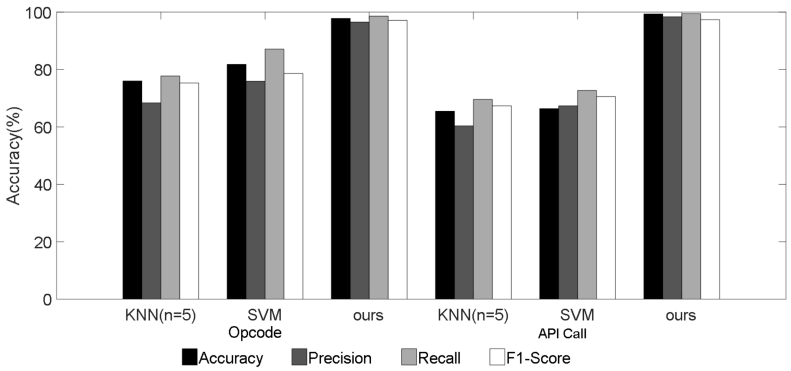


Figure 5: Comparison of experimental results of traditional machine learning methods

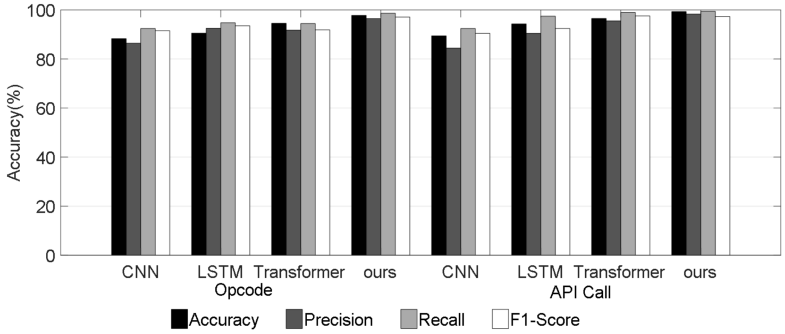


Figure 6: Comparison of experimental results of deep learning methods

nign samples and malicious samples that can be correctly detected by the model.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{11}$$

2) Precision: Among all predicted malicious samples, the proportion of truly malicious samples.

$$Precision = \frac{TP}{(TP + FP)} \tag{12}$$

3) Recall: Recall refers to the proportion of samples that are correctly predicted malicious among all malicious samples.

$$Recall = \frac{TP}{(TP + FN)} \tag{13}$$

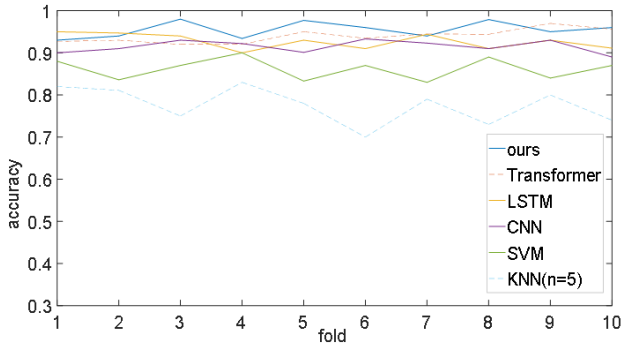
4) F1-Score: F1-Score is the harmonic mean of precision and recall, indicating the comprehensive classification ability.

$$F1 - Score = 2 \times \frac{precisionrecall}{(precision + recall)} \tag{14}$$

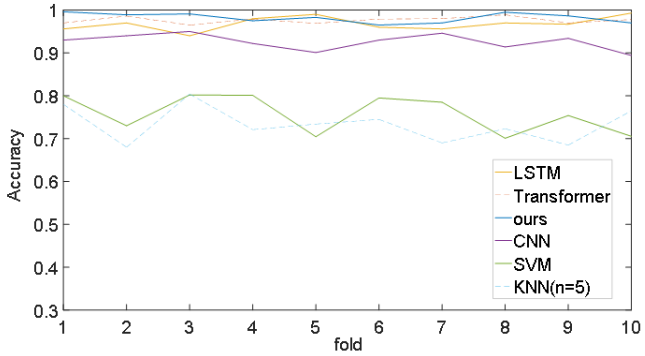
3.2 Experimental Results and Analysis

The super parameters d_k,d_v ,d_{model},h and M of encoder number in the improved Transformer in the experiment are 16, 16, 128, 8, and 6, respectively. During training, the epoch is 50, and the bach_size is 512. Adam is used as the optimizer, and the learning rate is 0.001. The drop is 0.4. Figure 4 shows the accuracy of the proposed model at different epochs. For OpCodes, the accuracy converges to a fixed value at the 30th iteration. For API Calls, our model converges faster, and the accuracy converges to a fixed higher value at the 21st iteration with an accuracy of 98.6%.

To compare the performance of different models under different datasets, we use traditional machine learning algorithms and several deep learning algorithms to conduct comparative experiments on the four indexes of accuracy, precision, recall, and F1-Score. The results are as follows Figures 5, 6, and 7. In terms of machine learning methods, we select the K-Nearest Neighbor (KNN) and the Support Vector Machine algorithm (SVM) for comparison. The experimental results show that the proposed method is significantly better than the two traditional algorithms in four indexes. In terms of deep learning methods, we select CNN, LSTM, and the recent Transformer model for training. The experimental results suggest that our model achieves the optimal detection results on the four indexes and outperforms other models on both OpCodes and API Calls, indicating that the proposed model has better generalization performance. In addition, as shown in the figure, the detection accuracy of the Transformer model is better than that of CNN and LSTM, indicating that the attention mechanism has an advantage in capturing correlation between sequences, helping to better



(a) 10-fold cross validation results comparison under different classifiers on OpCodes



(b) 10-fold cross validation results comparison under different classifiers on API Calls

Figure 7: 10-fold cross validation results comparison under different classifiers

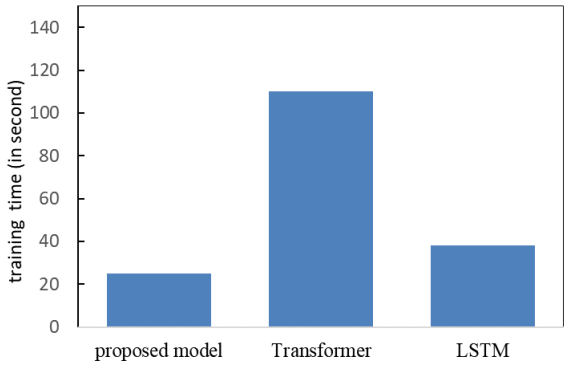


Figure 8: The training time of different models on API Calls

extract features. At the same time, compared with the Transformer, the detection results of our proposed model are better than the latter, which indicates that the residual weights play a role, which makes the model improve the accuracy to a certain extent while maintaining the network depth.

In terms of training time, we conduct a comparative analysis on the API Call dataset with two deep learning models, one of which is the LSTM model often used for threat hunting, and the other is the fundamental model Transformer of the proposed model. Fig.8 shows the comparison of the training time of the proposed model and those two models. It can be seen from the figure that the training time of our proposed model is significantly less than that of LSTM and Transformer, indicating that the additive attention mechanism and residual weights greatly reduce training time, which further demonstrates the effectiveness of the proposed approach.

4 Conclusions

In this paper, we propose a threat hunting method based on improved Transformer for detecting IoT malware. We first process the dataset through n-gram technique to convert the features of each sample into a vector representation. Then train the improved Transformer network to obtain a threat hunting model based on the improved Transformer. Finally, our proposed method is evaluated using the test set OpCode and API Call sequences and obtains over an accuracy rate of 98%. Meanwhile, Finding from our evaluations demonstrates that the proposed threat hunting method outperforms traditional KNN, SVM, and mainstream CNN, LSTM, and Transformer models in four metrics of accuracy, precision, recall, and F1-Score. Mainly because our proposed model can process the features of input samples in parallel, and the complete multi-head attention mechanism can help capture the correlation between features, which effectively improves the accuracy of malware detection. In

addition, we use additive attention mechanism and residual weight parameters in the encoder of each layer, which effectively reduces the training time while improving the accuracy, which is verified by experiments.

In the future, we will evaluate the proposed model on other malware datasets, such as ByteCode, permission, etc. At the same time, we will also explore new deep learning models to increase malware detection accuracy and reduce training time, especially for malware variants.

Acknowledgments

This work was supported by the State Grid Corporation Science and Technology Project "Research on the Identification and Active Defense of Advanced Persistent Threat for New Power System" under Grant 5700-202199539A-0-5-ZN.

References

[1] B. Alsulami and S. Mancoridis, "Behavioral malware classification using convolutional recurrent neural networks," in *13th international conference on malicious and unwanted software (MALWARE)*. IEEE, pp. 103–111, 2018.

[2] T. Bachlechner, B. P. Majumder, H. Mao, G. Cottrell, and J. McAuley, "Rezero is all you need: Fast convergence at large depth," in *Uncertainty in Artificial Intelligence*. PMLR, pp. 1352–1361, 2021.

[3] Z. Chen, Q. Yan, H. Han, S. Wang, L. Peng, L. Wang, and B. Yang, "Machine learning based mobile malware detection using highly imbalanced network traffic," *Information Sciences*, vol. 433, pp. 346–364, 2018.

[4] X. Fu, "Shan, 2018 fu j., xue j., wang y., liu z., shan c," *Malware visualization for fine-grained classification, IEEE Access*, vol. 6, pp. 14 510–14 523, 2018.

[5] H. Haddadpajouh, A. Mohtadi, A. Dehghantanaha, H. Karimipour, X. Lin, and K.-K. R. Choo, "A multi-kernel and metaheuristic feature selection approach for iot malware threat hunting in the edge layer," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4540–4547, 2020.

[6] A. Hota and P. Irolla, "Deep neural networks for android malware detection." in *ICISSP*, pp. 657–663, 2019.

[7] S. Hou, A. Saas, Y. Ye, and L. Chen, "Droiddelver: An android malware detection system using deep belief network based on api call blocks," in *International conference on web-age information management*. Springer, pp. 54–66, 2016.

[8] R. Islam, R. Tian, L. M. Batten, and S. Versteeg, "Classification of malware based on integrated static and dynamic features," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 646–656, 2013.

[9] 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.

[10] J. Jeon, J. H. Park, and Y.-S. Jeong, "Dynamic analysis for iot malware detection with convolution neural network model," *IEEE Access*, vol. 8, pp. 96 899–96 911, 2020.

[11] X. Jiachen, W. Yijun, and X. Zhi, "A survey of threat hunting in cyberspace," *Communication Technology*, vol. 53, no. 01, pp. 1–8, 2020.

[12] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Australasian joint conference on artificial intelligence*. Springer, pp. 137–149, 2016.

[13] Y. Ming and Z. Jian, "Malicious software static detection model based on image recognition," *Information Network Security*, vol. 21, pp. 25–32, 2021.

[14] R. Mirzazadeh, M. H. Moattar, and M. V. Jahan, "Metamorphic malware detection using linear discriminant analysis and graph similarity," in *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, pp. 61–66, 2015.

[15] D. Morató Osés, E. Berrueta Irigoyen, E. Magaña Lizarrondo, and M. Izal Azcárate, "Ransomware early detection by the analysis of file sharing traffic," *Journal of Network and Computer Applications, 124 (2018) 14–32*, 2018.

[16] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, "Malware classification with recurrent networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 1916–1920, 2015.

[17] M. Rabbani, Y. L. Wang, R. Khoshkangini, H. Jelodar, R. Zhao, and P. Hu, "A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing," *Journal of Network and Computer Applications*, vol. 151, p. 102507, 2020.

[18] I. Santos, F. Brezo, X. Ugarte-Pedrero, and P. G. Bringas, "Opcode sequences as representation of executables for data-mining-based unknown malware detection," *Information Sciences*, vol. 231, pp. 64–82, 2013.

[19] X. Su, D. Zhang, W. Li, and K. Zhao, "A deep learning approach to android malware feature learning and detection," in *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE, pp. 244–251, 2016.

[20] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware detection with deep neural network using process behavior," in *2016 IEEE 40th annual computer software and applications conference (COMPSAC)*, vol. 2. IEEE, pp. 577–582, 2016.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin,

"Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[22] C. Wu, F. Wu, T. Qi, Y. Huang, and X. Xie, "Fastformer: Additive attention can be all you need," *arXiv preprint arXiv:2108.09084*, 2021.

[23] S. Xin, Z. L. Zhao Kai, and F. Wenbo, "Research on android malware detection method based on random forest," *Netinfo Security*, no. 9, pp. 1–5, 2019.

[24] G. Yangchen, F. Yong, L. Liang, and Z. Lei, "Research on android malware detection technology based on convolutional neural network," *Journal of Sichuan University: Natural Science Edition*, vol. 57, no. 4, pp. 673–680, 2020.

[25] Y. Ye, Q. Liang, Z. Yi'an, Z. Lixiang, J. Yao, D. Jiawei, and N. Juntao, "A hybrid feature based malware detection method for mobile terminals," *Journal of Information Security*, vol. 7, no. 2, pp. 120–138, 2022.

[26] S. Y. Yerima and S. Sezer, "Droidfusion: A novel multilevel classifier fusion approach for android malware detection," *IEEE transactions on cybernetics*, vol. 49, no. 2, pp. 453–466, 2018.

[27] C. Yi, T. Di, and Z. Wei, "Android malware detection based on deep learning: achievements and challenges," *Journal of Electronics and Information*, vol. 42, no. 9, pp. 2082–2094, 2020.

[28] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: deep learning in android malware detection," in *Proceedings of the 2014 ACM conference on SIGCOMM*, pp. 371–372, 2014.

[29] B. Zhang, J. Yin, W. Tang, J. Hao, and D. Zhang, "Unknown malicious codes detection based on rough set theory and support vector machine," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, pp. 2583–2587, 2006.

[30] D. Zhu, H. Jin, Y. Yang, D. Wu, and W. Chen, "Deepflow: Deep learning-based malware detection by mining android application for abnormal usage of sensitive data," in *2017 IEEE symposium on computers and communications (ISCC)*. IEEE, pp. 438–443, 2017.

Biography

Yaping Li biography.Yaping Li is a master student of North China Electric Power University. Her research direction is power information security. Email:lyp2511630799@163.com.

Yuancheng Li biography.Yuancheng Li was a postdoctoral research fellow in the Digital Media Lab, Beihang University. He has been with the North China Electric Power University, where he is a professor and the Dean of the Institute of Smart Grid and Information Security. He was a postdoctoral research fellow in the Cyber Security Lab, college of information science and technology of Pennsylvania State University.

Intelligent Algorithms for Identification and Defense of Telecommunication Network Fraudulent Call Information under Legal System

Jianbing Yan

(Corresponding author: Zong-Liang Wang)

Department of Ideological and Political Education, Zhengzhou Tourism College, China
No. 188, Jinlong Road, Zhengdong New District, Zhengzhou, Henan 451464, China
Email: gyanzx880137@yeah.net

(Received Dec. 20, 2021; Revised and Accepted Jan. 23, 2023; First Online Feb. 17, 2023)

Abstract

With the development of technology, telecommunication network frauds are also appearing more frequently. For the identification and defense of telecommunication network fraud information, this paper first analyzed it from the legal system's perspective and designed an intelligent algorithm based on text classification. First, a convolutional neural network (CNN) extracted text features after text preprocessing. Then, a bi-directional long short-term memory (BiLSTM) algorithm was used to extract temporal information. Finally, the BiLSTM algorithm was optimized by combining attention to obtain the C-BiLSTM-attention intelligent algorithm. The experiment on the dataset found that compared with LSTM, BiLSTM, and C-BiLSTM algorithms, the C-BiLSTM-attention algorithm performed better on the text classification of call information. Furthermore, the precision, accuracy, recall rate, and F1 value were all the highest, 94.36%, 93.37%, 93.02%, and 93.69%, respectively. This proves that the C-BiLSTM-attention algorithm is reliable enough to be applied in actual telecommunication network fraud to achieve identification and defense.

Keywords: Call Information; Intelligent Algorithm; Legal System; Telecommunication Network Fraud

1 Introduction

While the ever-advancing information technology facilitates people's life and work, telecommunication network fraud has emerged more and more frequently [9]. Telecommunication network fraud is a remote and contactless fraud to induce victims to transfer money through calls and text messages [2, 11], bringing a double blow to people's spirit and property [16].

In parallel with the development of science and technology, the means of telecommunication network fraud have been updated, becoming more accurate and effective and spreading more widely [15]. With the help of technology, massive user information is stolen and victims, and fraud becomes more targeted. Monetary loss due to telecommunication network fraud continues to rise, the number of fraud cases increases constantly, and the fraudulent behavior becomes more hidden and difficult to detect, bringing greater difficulty to the identification and defense of telecommunication network fraud. How to effectively identify telecommunication network fraud has become an important issue at present.

Zhong *et al.* [17] processed part of a call, established a term frequency-inverse document frequency (TF-IDF) model, identified fraudulent calls through a generalized learning system, and found through experiments that the method had a high accuracy and fast training.

Meijaard *et al.* [7] designed an unsupervised machine learning method to detect international revenue sharing fraud (IRSF) and compared it with existing post-mortem anti-fraud schemes to demonstrate the effectiveness of the method.

Ying *et al.* [14] proposed an incremental graph mining-based fraudulent call detection method that can automatically label fraudulent phone numbers, performed experiments on an anti-fraud data set called whoscall, and found that the efficiency of the method significantly improved.

Min *et al.* [8] extracted the behavioral features of fraudulent calls, identified fraudulent phone numbers by the K-means algorithm, and proved the feasibility of the method by an actual sample set. After analyzing the legal system related to telecommunication network fraud, this paper designed a method that can classify the text of call information and proved the effectiveness of the method through experiments. This work provides theoretical support for better implementation of identification and defense in actual telecommunication network fraud.

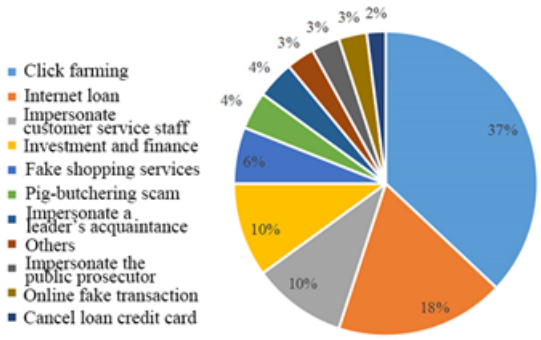


Figure 1: The top ten most prevalent types of fraud in 2021

2 Telecommunication Network Fraud Under the Legal System

Telecommunication network fraud means making up factors to infringe on other people's property with the assistance of computers and cell phones. In 2021, the National Anti-Fraud Center has conducted statistics on the top ten most prevalent types of fraud, as shown in Figure 1.

Currently, common scams can be divided into the following four categories.

1) Impersonation type: the cheater cooks up stories according to the characteristics of the victim, such as impersonating the public prosecutor and friends, to carry out money fraud.

2) Threatening type: the cheater impersonates gangdom, kidnappers, etc., to demand high ransom and extort money.

3) Lure type: the cheater fabricates scams such as winning prizes and tax refunds to defraud the victim.

4) Technical type: the cheater creates phishing websites, sends them to lure victims to click, and steals users' bank information.

Telecommunication network fraud has a high degree of concealment. With the help of the network, telecommunication network fraud tends to be more specialized and industrialized, and the updating speed of fraudulent means is also fast. Therefore, for telecommunication network fraud, although severe crackdowns have been carried out, such cases are still popping up. In the current legal system, the content related to telecommunications network fraud is as follows.

1) In the Criminal Law, there are detailed provisions on the types of financial fraud. When a fraud involves illegal fund-raising or the use of financial instruments, credit cards, and insurance and the fraud amount is large, the corresponding units and individuals will be punished; however, there are no independent and obvious provisions on telecommunication network fraud.

2) The Network Security Law clearly stipulates that network operators need to protect the security of user information and timely takes measures when user information is leaked.

3) The E-commerce Law gives clear industry regulations for e-commerce and regulates the industry order.

4) The Personal Information Protection Law establishes a sound system for the protection of personal information to prevent and punish infringement of personal information.

However, there are some imperfections in the current legal system, such as the identification of the crime of telecommunication network fraud. It may result in different sentences for the same case due to the fact that most of the crimes are committed by multiple people. Moreover, the types of punishment for fraud are also limited. Punishment against liberty and fine punishment cannot eliminate recidivism. Long-term preventive measures are lack of.

After telecommunication network fraud, telecommunication operators also accumulate a large amount of call information. If these call information can be effectively analyzed to classify telecommunication fraud information, it has a great effect on improving the identification and defense of telecommunication network fraud. Therefore, this paper designed an intelligent algorithm to analyze the communication information in telecommunication network fraud.

3 Intelligent Algorithm-Based Call Information Classification

3.1 Call Information Pre-Processing

In this paper, we realize the classification of call information from the perspective of text classification. After obtaining the original data set of call information text from operators and networks, it needs to be pre-processed first, and then by intelligent algorithms, text classification is performed.

Chinese text first needs to be processed by word separation, and common tools include Jieba word separation, NLPIR word separation, etc. [13]. This paper used Jieba word separation, which is based on string matching and easy to use. It has a high accuracy rate for word separation and has been widely used in natural language processing (NLP) [3]. Its word separation performance is as follows. Before word separation: Your credit card has high charges abroad.

After word separation: Your/credit card/has/ high charges/abroad.

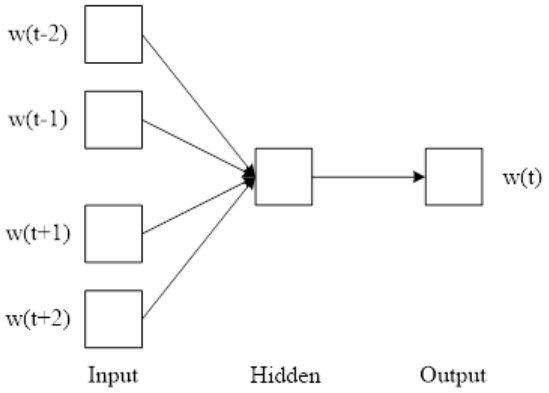


Figure 2: The CBOW model

After word separation: A table of stop words is established to eliminate stop words in the text, to improve the classification performance. Then, the Continuous Bag-of-Words (CBOW) model [10] in Wodr2Vec is used for text vectorization, as shown in Figure 2.

In the CBOW model, the input layer is the oneshot form of words. Then, in the hidden layer, all the oneshot vectors are multiplied by input weight matrix W, and the results are added up and averaged as the vector of the hidden layer. After that, it is multiplied by output weight matrix W' to obtain the V-dimensional vector. The intermediate word is predicted after the probability distribution of word at every dimension is obtained by softmax. Finally, the word vector of every word is obtained by updating W and W' through gradient descent. Word vectors at 300 dimensions are obtained by the CBOW model for intelligent algorithm classification after text classification.

3.2 Long Short-Term Memory-Based Intelligent Classification Algorithm

Long short-term memory (LSTM) is a special kind of recurrent neural network (RNN) [12]. Compared with RNN, LSTM has better performance in long time series and has wide applications in data prediction [5] and classification [1]. LSTM controls information through three gate structures. Let the input vector of LSTM be x and the output vector be h. The computational steps are shown below.

1) Which information needs to be discarded from previous cell state C_{t-1} is determined through the forgetting gate, and the corresponding equation is:

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

where t stands for the time, w_f and b_f are the weight and bias, and σ is the sigmoid activation function.

2) Which information needs to be added to new cell state C_t is decided through the input gate, and the corresponding equations are:

$$\begin{aligned} i_t &= \sigma(w_i[h_{t-1}, x_t] + b_i), \\ \widetilde{C_t} &= \tanh(w_c[h_{t-1}, x_t] + b_c), \\ C_t &= f_t C_{t-1} + i_t \widetilde{C_t}. \end{aligned}$$

3) Output value h_t is obtained through the output gate. The equations are:

$$\begin{aligned} O_t &= \sigma(w_o[h_{t-1}, x_t] + b_o), \\ h_t &= O_t \tanh(C_t). \end{aligned}$$

However, LSTM can only save one-way information. In order to better capture two-way information, bi-directional LSTM (BiLSTM) is proposed to process data in both forward and reverse directions. The output of BiLSTM is jointly determined by forward LSTM output $\overrightarrow{h_t}$ and backward LSTM output $\overleftarrow{h_t}$:

$$\begin{aligned} \overrightarrow{h_t} &= LSTM(x_t, \overrightarrow{h_{t-1}}), \\ \overleftarrow{h_t} &= LSTM(x_t, \overleftarrow{h_{t-1}}), \\ H &= i_t \overrightarrow{h_t} + j_t \overleftarrow{h_t} + b_t, \end{aligned}$$

where H is the vector matrix combining the forward and backward outputs and b_t is the deviation at time t.

BiLSTM lacks the ability to discover features automatically. A convolutional neural network (CNN) [4] algorithm can extract the feature information effectively but ignores the sequence information of texts. In order to further improve the performance of intelligent algorithms on text classification of call information, this paper combines CNN with BiLSTM to obtain the CNN-BiLSTM algorithm. The attention mechanism [6] selects the key information from all the information for prominent attention. Therefore, this paper further optimizes the CNN-BiLSTM algorithm with the attention mechanism to obtain the intelligent C-BiLSTM-attention algorithm.

In the CNN algorithm, for input vector x_i, the output after convolution is:

$$l_i = \tanh(x_i k_i + b_i)$$

where k_i is the convolution kernel weight and b_i is the convolution kernel deviation.

The principle of the attention mechanism is shown in Figure 3.

The attention mechanism essentially consists of multiple "query" and key-value. First, the correlation between "query" and every "key" is computed: $F(Q, K) = Q \cdot K$; then, the weight coefficient of the "value" is calculated: $a_i = softmax(F(Q, K))$; finally, the target attention value is obtained by weighting and summation:

$$Attention(Q, K, V) = \sum_{i=1}^{L_x} a_i \cdot value,$$

where L_x is the length of the key-value.

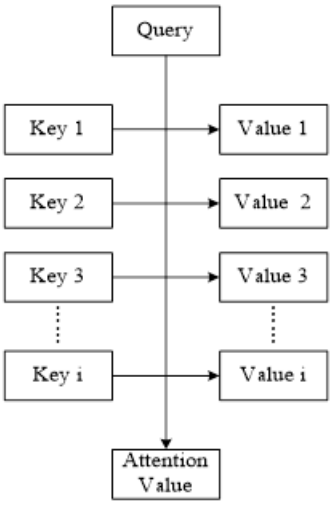


Figure 3: Principle of the attention mechanism

The structure of the intelligent C-BiLSTM-attention algorithm is shown in Figure 4.

The input layer is input with the processed call information text of telecommunication network fraud; then, feature extraction and dimension reduction are performed in the CNN layer; after forward and backward LSTM in the BiLSTM layer, the attention value is calculated in the attention layer; finally, the result of text classification is output through the output layer.

4 Results and Analysis

4.1 Experimental Setup

The experimental data were call information collected from telecommunication operators, which have all been converted into text information, including 10,000 normal call information and 10,000 fraudulent call information. Some examples are shown in Table 1.

Label 0 was used to represent normal call information, and label 1 was used to represent fraudulent call information; 70% of the data in the dataset was used for algorithm training, and 30% was used for algorithm testing. The experimental environment was Windows 10 and 8 G memory. The i7-4720HQ 2.60GHz processor and Python programming language were used. Vectors at 300 dimensions were obtained after the COBW model was trained by the texts. The C-BiLSTM-attention algorithm used the Adam optimization algorithm. The learning rate was 0.0001, and the activation functions used in convolution, LSTM, and attention layers were tanh, sigmoid, and softmax, respectively. The batch size was 128.

4.2 Evaluation Indicators

Based on the confusion matrix (Table 2), the performance of the algorithm was evaluated.

For the binary classification between normal call information and fraudulent call information, the evaluation indicators used are as follows.

1) Precision: $P = \frac{TP}{TP+FP}$;

2) Accuracy: $A = \frac{TP+TN}{TP+FP+TN+FN}$;

3) Recall rate: $R = \frac{P}{TP+FN}$;

4) F1 value: $F1 = \frac{2PR}{P+R}$

4.3 Analysis of Results

The performance of four intelligent algorithms, LSTM, BiLSTM, C-BiLSTM, and C-BiLSTM-attention algorithms, was compared for fraudulent call message identification and defense. After the algorithms are trained, ten experiments were conducted on the test set, and the accuracy of different algorithms are compared, as shown in Figure 5.

It was observed in Figure 5 that the classification precision of these LSTM algorithms was all above 90%. In comparison, the precision of LSTM is the lowest, around 91%, and BiLSTM was between 92%-93%. After combining CNN, the precision of the C-BiLSTM algorithm was around 93%, indicating that the classification performance of the BiLSTM algorithm was improved to some extent after CNN feature extraction. The C-BiLSTM-attention algorithm had the highest precision, between 94% and 95%, indicating that the intelligent algorithm performed best after combining CNN and the attention mechanism.

A comparison of the accuracy between different algorithms is shown in Figure 6.

The comparison results in Figure 6 were similar to Figure 5, LSTM ¡ BiLSTM ¡ C-BiLSTM ¡ C-BiLSTM-attention. When using the LSTM algorithm for classification, its accuracy was below 90%; the accuracy of the BiLSTM algorithm was around 91%; after optimization by CNN and the attention mechanism, the C-BiLSTM-attention algorithm had the highest accuracy, around 93%. These results proved the reliability of the C-BiLSTM-attention algorithm for fraudulent call information identification and defense.

The results of the ten experiments were averaged. The performance of different algorithms was compared, as shown in Figure 7.

It was seen from Figure 7 that among the four algorithms, the C-BiLSTM-attention algorithm had the best performance. Its precision was 94.36%, which was 3.33% higher than that of the LSTM algorithm; its accuracy was 93.37%, which was 3.65% higher than the LSTM algorithm; its recall rate was 93.02%, which was 2.69% higher than the LSTM algorithm; its F1 value was 93.69%, which was 3.01% higher than the LSTM algorithm. These results verified the effectiveness of CNN and the attention mechanism for improving the classification performance of the algorithm. The C-BiLSTM-attention algorithm

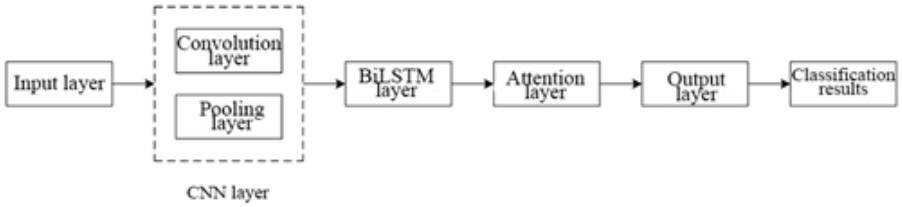


Figure 4: Structure diagram of the intelligent C-BiLSTM-attention algorithm

Table 1: Examples of experimental data

Type	Example
Fraudulent call information	Hello, I am from XX bank. I found that there is a bank card under your name suspected of major criminal cases.
	Uncle! I was arrested for committing a crime outside. Now I have to pay 10,000 yuan. Please give me some money, and I will pay you back after coming back.
	Hello, your flight has been canceled due to a mechanical problem. Do you want to alternate your ticket? Please provide us with your ID number and bank card number.
	Honey! Genuine overseas purchasing! Fake one lose ten. We will give you the lowest price.
	Wang, the last project needs to be paid, so you should hurry up and transfer the money to XXXXXXX.
	Hello, we are from the XX People's Court and have found that you are involved in money laundering.
	The item you purchased on this website is out of stock. Please leave your bank card number information, and we will refund to you.
	Friends, lack of money? Low interest rate, fast arrival, no collateral, no guarantee, legal company, very fast lending.
	I have a high-yield financial product. You can get it if you charge money and become a member.
Normal call information	Have you eaten dinner yet?
	I have to work overtime today and will be back later.
	See you Saturday morning at 10 am at XX Plaza. Let's go for a barbecue dinner.
	I'll take care of this as soon as possible and get it to you by the end of the day.

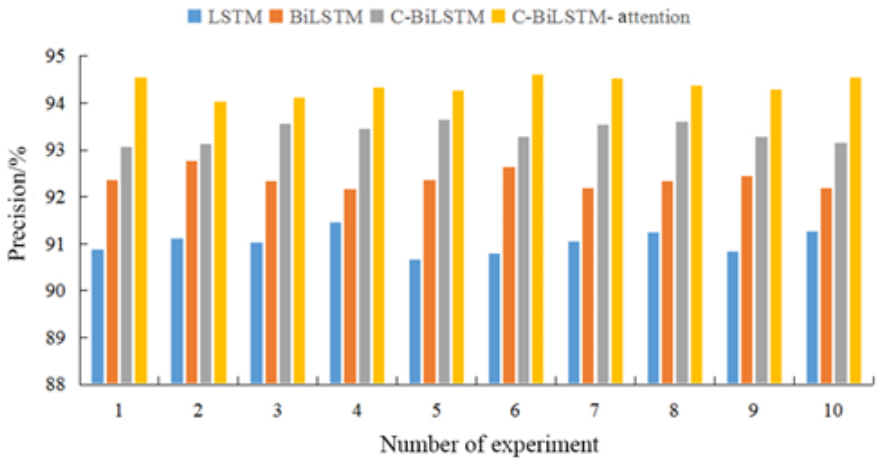


Figure 5: Comparison of precision between different algorithms for fraudulent call information identification and defense

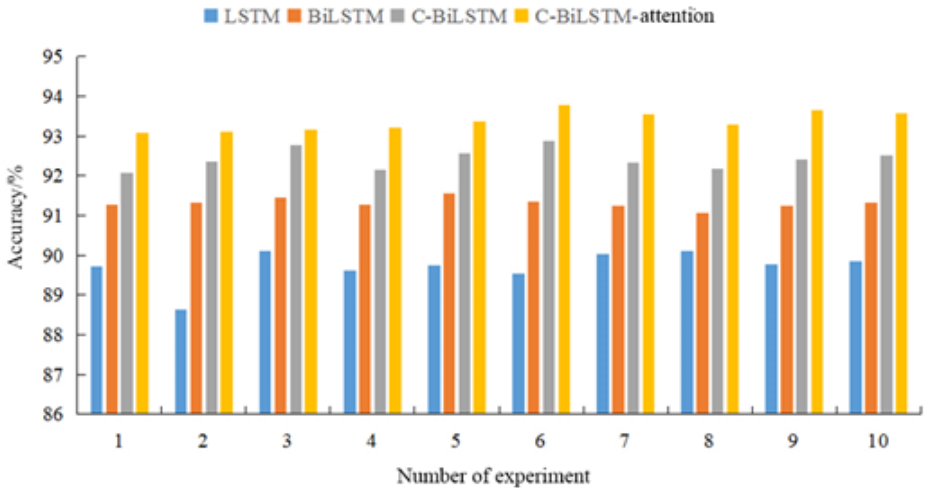
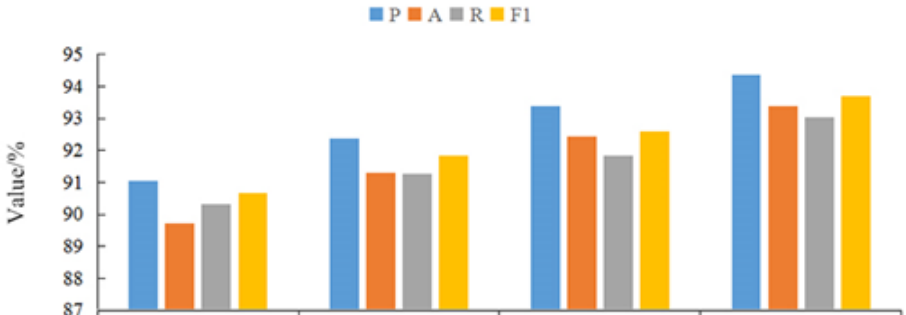


Figure 6: Comparison of accuracy between different algorithms for fraudulent call information identification and defense



	LSTM	BiLSTM	C-BiLSTM	C-BiLSTM-attention
P	91.03	92.38	93.37	94.36
A	89.72	91.31	92.42	93.37
R	90.33	91.27	91.84	93.02
F1	90.68	91.82	92.60	93.69

Figure 7: Comparison of performance between different algorithms for fraudulent call information identification and defense

Table 2: Confusion matrix

Real label	Classified as 1	Classified as 0
1	TP	FN
0	FP	TN

can accurately classify the text of call information for the identification and defense of telecommunication network fraud.

5 Conclusion

This paper designed an improved LSTM intelligent algorithm, the C-BiLSTM-attention algorithm, for the identification and defense of call information in telecommunication network fraud under the legal system. It was found through experiments on the dataset that compared with LSTM, BiLSTM, and C-BiLSTM algorithms, the C-BiLSTM-attention algorithm had better performance, with a precision of 94.36% and an accuracy of 93.37%. The C-BiLSTM-attention algorithm can be applied in the actual telecommunication network fraud identification and defense to effectively distinguish normal call information from fraudulent call information. At the same time, the government should perfect the relevant legal system in practical life to reduce the occurrence of telecommunication network fraud more efficiently.

References

[1] S. N. Chen, F. Liu, C. X. Gao, J. Li, "Gearbox fault diagnosis classification with empirical mode decomposition based on improved long short-term memory," in *IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA'21)*, pp. 568-575, 2021.

[2] O. F. Cossa, N. Sousa, R. Gonçalves, J. Martins, F. Branco, "Prediction of bank frauds by SMS or voice, from cell phone data analysis: A Systematic Literature Review," in *16th Iberian Conference on Information Systems and Technologies (CISTI'21)*, pp. 1-8, 2021.

[3] J. Dong, G. Li, "Hybrid filtering recommendation system for libraries," in *Innovative Computing (IC'20)*, pp. 975-981, 2020.

[4] H. A. Haenssle, C. Fink, R. Schneiderbauer, F. Toberer, T. Buhl, T. Blum, A. Kallo, A. Ben Haji Hassen, L. Thomas, A. Enk, L. Uhlmann, "Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists," *Annals of Oncology*, vol. 29, no. 8, pp. 1836-1842, 2018.

[5] J. Kumar, R. Goomer, A. K. Singh, "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters," *Procedia Computer Science*, vol. 125, pp. 676-682, 2018.

[6] L. Liu, H. Chen, Y. Sun, "A multi-classification sentiment analysis model of Chinese short text based on gated linear units and attention mechanism," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 6, pp. 109.1-109.13, 2021.

[7] Y. J. Meijaard, B. C. M. Cappers, J. G. M. Mengerink, N. Zannone, "Predictive analytics to prevent voice over IP international revenue sharing fraud," in *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 241-260, 2020.

[8] X. Min, R. Lin, "K-Means algorithm: Fraud detection based on signaling data," in *IEEE World Congress on Services (SERVICES'18)*, pp. 21-22, 2018.

[9] L. Peng, R. Lin, "Fraud phone calls analysis based on label propagation community detection algorithm," in *IEEE World Congress on Services (SERVICES'18)*, pp. 23-24, 2018.

[10] D. Qiu, H. Jiang, S. Chen, "Fuzzy information retrieval based on continuous bag-of-words model," *Symmetry*, vol. 12, no. 2, pp. 1-11, 2020.

[11] V. V. Sergeev, I. M. Gorbchenko, V. V. Safronov, "Comparative analysis of fraud detection systems by phone number," *Journal of Physics: Conference Series*, vol. 1679, no. 5, pp. 1-4, 2020.

[12] R. Viadinugroho, D. Rosadi, "Long short-term memory neural network model for time series forecasting: Case study of forecasting IHSG during Covid-19 outbreak," *Journal of Physics: Conference Series*, vol. 1863, no. 1, pp. 1-11, 2021.

[13] Z. Wang, W. Fang, X. Zhao, S. Wei, Y. Feng, Y. Chang, "Transliteration recognition of Tibetan person name based on Tibetan cultural knowledge," *International Journal of Computational Science and Engineering*, vol. 22, no. 2/3, pp. 305-312, 2020.

[14] J. J. C. Ying, J. Zhang, C. W. Huang, K. T. Chen, V. S. Tseng, "FrauDetector +: An incremental graph-mining approach for efficient fraudulent phone call detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 6, pp. 1-35, 2018.

[15] Y. Yuan, K. Ji, R. Sun, L. Ma, Z. Chen, L. Wang, "An Integration method of classifiers for abnormal phone detection," in *6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC'19)*, pp. 1-6, 2019.

[16] J. Zhang, X. Yao, X. Fu, "Identifying unfamiliar callers' professions from privacy-preserving mobile phone data," in *16th International Conference on Mobility, Sensing and Networking (MSN'20)*, pp. 524-530, 2020.

[17] R. Zhong, X. Dong, R. Lin, H. Zou, "An incremental identification method for fraud phone calls based on broad learning system," in *IEEE 19th International Conference on Communication Technology (ICCT'19)*, pp. 1306-1310, 2019.

Biography

Jianbing Yan, born in October 1985, graduated from Macau University of Science and Technology in 2016 and received the doctor's degree of law. He is now a lecturer in Zhengzhou Tourism College. He is interested in financial law.

Security Analysis and Improvement of an Access Control Protocol for WBANs

Parvin Rastegari[1], Mojtaba Khalili[2], and Ali Sakhaei[3]
(Corresponding author: Parvin Rastegari)
Electrical and Computer Engineering Group, Golpayegan College of Engineering, Isfahan University of Technology[1]
Golpayegan, 87717-67498, Iran
Email: p.rastegari@iut.ac.ir
Department of Electrical Engineering, K. N. Toosi University of Technology[2]
Tehran, 1631714191, Iran
Department of Statistics, Payame Noor University (PNU)[3]
P.O.Box 19395-4697, Tehran, Iran
(Received Sept. 13, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)[*]

Abstract

Since the emergence of wireless body area networks (WBANs) as a new technology in telemedicine, the challenges of secure communications in these networks have been noticed extensively; recently, Gao *et al.* have designed an efficient access control protocol for WBANs and claimed that their proposal could authenticate the physician to the patient and satisfy the confidentiality of the request message sent from the physician to the patient concurrently in a certificateless setting. Moreover, at the end of the protocol, the physician and the patient establish a session key for their following secure communications. They first designed a certificateless signcryption (CL-SC) scheme and then implied it to propose their access control protocol. In this paper, we design a key replacement attack against Gao *et al.*'s CL-SC scheme, in which the adversary can obtain the confidential request message sent from the physician to the patient. Moreover, based on our designed attack, the adversary can obtain the session key established by the physician for the following communications to the patient. Afterward, we fix the scheme to be secure against our proposed attacks.

Keywords: Access Control Protocol; Certificateless signcryption; ROM; Signcryption; WBANs

1 Introduction

New technologies for telemedicine have been extensively spread all over the world. The wireless body area network (WBAN) technology which was first proposed in 1996 [21], plays an important role in this field. In a WBAN, the human's vital data from body and environment parameters are collected via a wireless network including some low-power small sensors and actuators. The sensors might be wearable (such as neck, wrist, eye, arm, foot and body wears sensors) or implantable (such as cerebral pressure sensors, blood analyzer chips, heart sensors and so on). Due to the extensive applications of WBANs in various fields (e. g. medical, military, lifestyle, entertainment and so on), the IEEE 802.15.6 standard was presented to provide short-range reliable seamless communications with low-power consumption [13]. It is important to note that the efficiency in the sense of storage, computation and communication costs gets a lot of attention in WBANs because of the source-constrained low-power sensors and the bandwidth-limited communications.

The security aspects of WBANs have been in much attention in recent years. There are a lot of studies in this field and readers can refer to [7, 13] for a comprehensive review. The security requirements of WBANs depend on the applications in which they are used. As mentioned, one of the main applications of these networks is in telemedicine which helps us to replace the face-to-face interaction between the patient and the physician by monitoring patients? health-related parameters remotely, processing and sending them to medical databases. So, the corresponding medical advice can be transferred to the patients according to the received vital data. This remote interaction, can reduce both the medical costs and the risk of infection in infectious disease such as COVID-19 [9]. It is clear that the authentication of the patient and the medical team as well as the confidentiality of the transferred messages, to preserve the privacy of the patients, are very important in telemedicine applications.

A digital signature scheme is a well-known primitive which can be used to satisfy the authentication, the non-repudiation and the integrity of the messages in security protocols. Moreover, an encryption scheme satisfies the confidentiality of the messages in these protocols. In 1997,

[*]This article is an extended version of an ISCISC'21 paper [17]

Zheng proposed the concept of a signcryption scheme which provides the goals of the signature and encryption schemes concurrently in a way much more efficient than encrypting and signing messages separately [20]. A signcryption scheme is a useful primitive for designing access control protocols which manage the security and privacy of the networks by allowing only authorized users to access the network. There are a lot of studies on designing efficient access control protocols based on signcryption schemes in the literature [1, 4, 6, 8, 10–12], which many of them are proposed in the certificateless setting [6,8,10,11]. Certificateless public key cryptography was proposed by Al-Riyami and Paterson in 2003 to eliminate the problem of the management of huge number of certificates in conventional public key infrastructure as well as the key escrow problem in ID-Based public key cryptography [2]. In 2008, the idea of certificateless signcryption (CL-SC) was proposed by Barbosa and Farshim [3]. Since the introduction of CL-SC scheme in 2008, some works have been done to propose CL-SC schemes with provable security in the standard model (i. e. without the assumption of random oracles) [5,15,18,19]. However these schemes are not suitable for designing access control protocols for source-constrained low-power applications such as WBANs because of their heavy computation costs. Furthermore, as some of these schemes are attacked in the literature (such as the proposed attacks in [14,16]), one can see that if the games for the security proofs of schemes are not designed correctly, the security of them are not reliable at all even in the standard model. Based on these descriptions, almost all proposed CL-SC schemes for source-constrained, low-power and bandwidth-limited applications are content with the security proofs in the random oracle model (ROM) [6, 8, 10, 11].

Recently, Gao *et al.* have proposed an efficient CL-SC scheme and designed an access control for WBANs based on their proposal [6]. They proved the confidentiality (IND-CCA2) and unforgeability (EUF-CMA) of their proposal against both the key replacement attacker (which is denoted as $\mathcal{A}_I$ in the literature) and the malicious KGC attacker (which is denoted as $\mathcal{A}_{II}$ in the literature) in the random oracle model (ROM). In this paper:

- We design an attack which shows that the confidentiality of Gao *et al.*'s CL-SC scheme is vulnerable against the key replacement attack, in contrast to their claim. In the designed attack, a key replacement attacker $\mathcal{A}_I$ can obtain the signcrypted messages by replacing the public key of the receiver.
- According to our attack, $\mathcal{A}_I$ can also obtain the session key which is established by the physician for the next communications to the patient in Gao *et al.*'s access control protocol.
- We fix the Gao *et al.*'s scheme to be robust against the proposed attacks.

The remained of the paper is organized as follows. In Section 2, some required preliminaries are provided. In Section 3, an overview of Gao *et al.*'s CL-SC scheme and access control protocol is described. In Section 4, we propose our attacks against the Gao *et al.*'s proposals. In Section 5, we fix Gao *et al.*'s proposals to be robust against our designed attacks. In Section 6, a comparison between Gao *et al.*'s proposals and our improvements is provided. Finally, the paper is concluded in Section 7.

2 Preliminaries

2.1 Related Complexity Assumptions

Definition 1. *Suppose that G is a group of a prime order q and P is a generator of G. The Discrete Logarithm (DL) Problem is that on inputs $P, aP \in G$ (for unknown $a \in \mathbb{Z}_q^*$), compute $a \in \mathbb{Z}_q^*$.*

Definition 2. *The Decisional Diffie-Hellman (DDH) Problem is that on inputs $P, aP, bP, X \in G$ (for unknown $a, b \in \mathbb{Z}_q^*$), decide whether $X = abP$ (and returns $\gamma = 1$) or not (and returns $\gamma = 0$).*

2.2 CL-SC Scheme

2.2.1 Syntax

A key generation center (KGC), a sender (A) and a receiver (B) are three entities in a CL-SC scheme for an access control in a WBAN, which has five algorithms as follows [6]:

Setup. The KGC takes a security parameter k as input and outputs a master key α which is kept secret and the public parameters *params* which are published.

Partial Key Generation (ParKeyGen). When a user U with the identity ID_U registers to KGC, the KGC calculates a corresponding partial key $ParK_U$ and sends it to U.

Key Generatin (KeyGen). When the user U receives $ParK_U$ from KGC, he/she selects a secret value x_U randomly and calculates his/her public/private key pair (PuK_U, PrK_U) by the use of $ParK_U$ and x_U.

Signcryption. Suppose that the sender A wants to create a signcryption δ on a message m for the receiver B. A uses his/her private key PrK_A and the B's public key PuK_B to create such signcryption.

UnSigncryption. Upon receiving δ from A, B uses his/her private key PrK_B and the A's public key PuK_A to verify δ and obtain m.

2.2.2 Security Requirements

In a certificateless setting, there are two types of adversaries [2]:

- The type I adversary $\mathcal{A}_I$ who can replace public keys of the users, but does not have access to the master key which is called as the key replacement attacker. In adversarial models in the literature, $\mathcal{A}_I$ is assumed to have access to the Public-Key, Partial-Key, Replace-Public-Key, Private-Key, Signcrypt, Unsigncrypt and Hash oracles.

- The type II adversary $\mathcal{A}_{II}$ who has access to the master key, but is not able to replace public keys which is called as the malicious KGC attacker. In adversarial models in the literature, $\mathcal{A}_{II}$ is assumed to have access to the Public-Key, Private-Key, Signcrypt, Unsigncrypt and Hash oracles.

A CL-SC scheme must satisfy two basic security requirements, i. e. the confidentiality (in the sense of IND-CCA2) and the unforgeability (in the sense of EUF-CMA) against both $\mathcal{A}_I$ and $\mathcal{A}_{II}$. These security requirements are defined by four games described in [15].

2.3 Network Model of an Access Control for WBANs

According to the IEEE 802.15.6 standard, WBANs are deployed in a star topology in which a node located on the center of the body (e. g. the waist) plays the role of a controller [13] which can communicate to all sensor nodes directly. Sensor nodes which are located in, on or around the body, gather the vital information of the patient (Bob) and sends them to the central controller regularly. The controller sends the aggregated information to the receiver e. g. the physician (Dr. Alice) via the internet. Then the receiver (Dr. Alice) analyses the received information and sends the corresponding message e. g. the medical advice to the patient (Bob).

It is obvious that without considering the security aspects in this topology, the privacy of the patient (Bob) is not preserved at all, since everybody can access to his vital information and the corresponding medical advice from the insecure internet platform. Access control protocols provide solutions to overcome this problem by permitting to only authorized entities to have access to the private information. In [6], Gao *et al.* have proposed a CL-SC scheme and implied their proposal to design an access control protocol for WBANs. In their model, a service provider (SP) is responsible for deploying WBANs and registering all users (including the patients and the physicians) to the network. In fact, the SP plays the role of the KGC in the CL-SC scheme who generates the partial keys of the users as explained in Section 2.2.1. Figure 1 shows the star topology and the interactions between the SP and the users in Gao *et al.*'s network model.

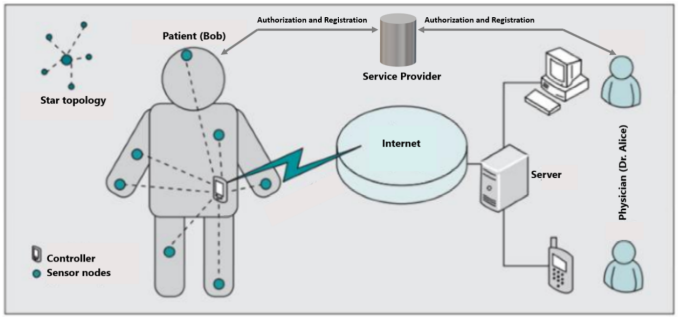


Figure 1: The star topology and the interactions between the SP and the users in Gao *et al.*'s model

3 Gao *et al.*'s Proposals

In [6], Gao *et al.* proposed a CL-SC scheme without bilinear pairing and proved the security of their proposal in the random oracle model (ROM). Afterwards, they designed an access control protocol for WBANs based on their CL-SC scheme. In this section, an overview of their CL-SC scheme and access control protocol is provided.

3.1 Gao *et al.*'s CL-SC Scheme

The algorithms of Gao *et al.*'s CL-SC scheme are as follows [6]:

Setup. On input a security parameter k, the SP selects a cyclic group G of a large prime order q, a generator P of G and three collision-resistant hash functions $H_1 : \{0,1\}^* \times G \longrightarrow \mathbb{Z}_q^*$, $H_2 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$ and $H_3 : \mathbb{Z}_q^* \longrightarrow \{0,1\}^{l_0+|\mathbb{Z}_q^*|}$, where l_0 is the bit length of the message and $|\mathbb{Z}_q^*|$ is the bit length of an element in $\mathbb{Z}_q^*$. Afterwards, the SP picks a random $\alpha \in_R \mathbb{Z}_q^*$ as the system's master key and calculates the corresponding public key $P_{pub} = \alpha P$. At last, the SP publishes public parameters $params = \{G, q, P, P_{pub}, H_1, H_2, H_3\}$ and keeps α secret.

ParKeyGen. In this algorithm, an entity U sends his/her identity ID_U to the SP. The SP chooses a random value $r_U \in_R \mathbb{Z}_q^*$ and calculates $R_U = r_U P$ and $d_U = r_U + \alpha H_1(ID_U, R_U)$ and sends $ParK_U = (R_U, d_U)$ to U via a secure channel. The user U can verify the correctness of the received partial keys by checking whether the equation $R_U + H_1(ID_U, R_U)P_{pub} = d_U P$ holds or not.

KeyGen. The entity U picks a random $x_U \in_R \mathbb{Z}_q^*$, computes $X_U = x_U P$ and sets $PrK_U = (d_U, x_U)$ as his/her private key and $PuK_U = (R_U, X_U)$ as his/her public key.

Signcryption. Suppose that an entity A wants to create a signcryption δ on a message m for an entity B. A executes the following steps:

1) Picks a random value $\beta \in_R \mathbb{Z}_q^*$ and computes $T = \beta P$.

2) Sets $h_B = H_1(ID_B, R_B)$.
3) Calculates $V = \beta(X_B + R_B + h_B P_{pub})$.
4) Sets $h = H_2(m||T||ID_A||ID_B||X_A||X_B)$.
5) Calculates $S = (x_A + \beta)/(h + d_A + x_A)$.
6) Calculates $C = H_3(V) \oplus (m||S)$.
7) Returns $\delta = (S, C, T)$ and sends it to B.

UnSigncryption. Upon receiving a signcryption $\delta = (S, C, T)$ from A, B executes the following steps to verify δ and obtain m:

1) Calculates $V = (x_B + d_B)T$.
2) Calculates $m||S = H_3(V) \oplus C$ and consequently recovers m as the first l_0 bits of $m||S$.
3) Sets $h = H_2(m||T||ID_A||ID_B||X_A||X_B)$.
4) Sets $h_A = H_1(ID_A, R_A)$.
5) Verifies the signcryption by checking the following equality:

$$S(X_A + R_A + h_A P_{pub} + hP) = X_A + T.$$

If the above equality holds, B accepts m as a signcryption from A, otherwise B rejects it and returns $\perp$.

3.2 Gao *et al.*'s Access Control Protocol

Gao *et al.*'s proposed access control protocol is summarized in Figure 2. Their proposal has four phases as follows:

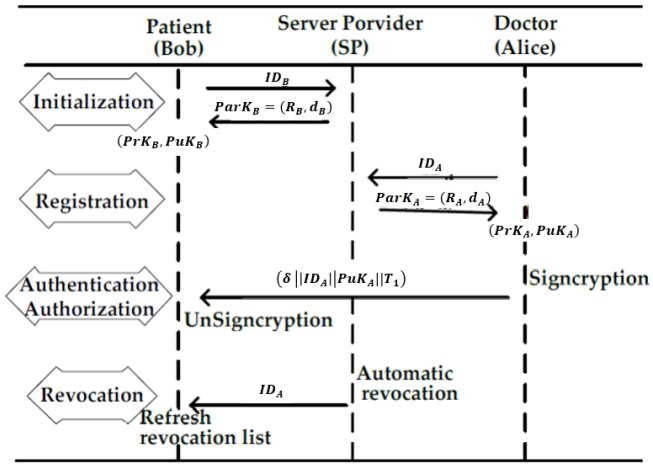


Figure 2: Gao *et al.*'s certificateless access control protocol

The Initialization Phase. In this phase, the SP who is the KGC introduced in Section 2.2.1, runs the Setup algorithm of the CL-SC scheme explained in Section 3.1, to generate *params* and α. The SP publishes *params* and keeps α secret. After deploying WBANs, when a user such as Bob requests SP for his partial key by sending his identity ID_B to the SP, it generates $ParK_B = (R_B, d_B)$ as explained in the ParKeyGen algorithm in Section 3.1 and sends it to Bob via a secure channel. Then Bob can produce his private and public keys (PrK_B, PuK_B) as explained in the KeyGen algorithm in Section 3.1.

The Registration Phase. In this phase, the receivers including the physician team such as Dr. Alice are registered by the SP. When Dr. Alice submits her identity ID_A to the SP, it checks whether the identity is valid or not. If not, the SP rejects the request. Otherwise, the SP sets an expiration date (ED) for Dr. Alice, generates $ParK_A = (R_A, d_A)$ as explained in the ParKeyGen algorithm in Section 3.1 and sends it to Dr. Alice via a secure channel. Then Dr. Alice can produce her private and public keys (PrK_A, PuK_A) as explained in the KeyGen algorithm in Section 3.1.

The Authentication Phase. When Dr. Alice wants to access the collected data of WBANs (e. g. the vital data of patient Bob), she first creates a signcryption $\delta = (S, C, T)$ on a request message m concatenated with a current timestamp T_1 (to prevent the replay attack), i. e. $m||T_1$. Then Dr. Alice sends $(\delta||ID_A||PuK_A||T_1)$ to Bob. Upon receiving the access request from Dr. Alice, Bob first checks whether $T_2 - T_1 \leq \Delta T$ or not, where T_2 is the current timestamp. If not, Bob rejects and terminates the session as a replay attack may be occurred. Otherwise, Bob runs the UnSigncryption algorithm by using his private key PrK_B. If the output of the UnSigncryption algorithm is $\perp$, Bob stops and terminates the session. Otherwise, Bob obtains m, accepts Dr. Alice's request, and starts to communicate with her using the session key $H_3(V)$ which is established between Bob and Dr. Alice.

The Revocation Phase. Due to the expiration date (ED), the SP revokes Dr. Alice access privilege by revoking her partial private key $ParK_A$ and sends ID_A to Bob which automatically makes Dr. Alice illegal to Bob. So, Bob stops the communication with Dr. Alice and she cannot be authenticated to Bob again, as her partial key is revoked by the SP.

4 Cryptanalysis of Gao *et al.*'s Proposals

4.1 Cryptanalysis of Gao *et al.*'s CL-SC Scheme

Gao *et al.* have claimed that their proposed CL-SC scheme is confidential (IND-CCA2) and unforgeable (EUF-CMA) against type I and type II adversaries $\mathcal{A}_I$ and $\mathcal{A}_{II}$, in the random oracle model (ROM), based on

CDH and DL assumptions in G (See Definition 1 and Definition 2) [6]. However, in this section, we design a key replacement attack against the confidentiality of Gao *et al.*'s CL-SC scheme. In our proposed attack, a type I adversary $\mathcal{A}_I$ can replace the public key of the receiver B to obtain all signcrypted messages sent from A to B without the knowledge of the corresponding private key of B. To this goal, $\mathcal{A}_I$ picks two random values $r_B^*, x_B^* \in_R \mathbb{Z}_q^*$ and computes:

$$R_B^* = r_B^* P,$$
$$X_B^* = x_B^* P - H_1(ID_B, R_B^*) P_{pub}.$$

Then $\mathcal{A}_I$ replaces the real public key of B, i. e. $PuK_B = (R_B, X_B)$, with $PuK_B^* = (R_B^*, X_B^*)$. By this public key replacement, A will use (R_B^*, X_B^*) to create a signcryption δ^* on a message m for B. In order to produce δ^*, A runs the following steps:

1) Picks a random value $\beta \in_R \mathbb{Z}_q^*$ and computes $T = \beta P$.
2) Sets $h_B^* = H_1(ID_B, R_B^*)$.
3) Calculates $V^* = \beta(X_B^* + R_B^* + h_B^* P_{pub})$.
4) Sets $h^* = H_2(m||T||ID_A||ID_B||X_A||X_B^*)$.
5) Calculates $S^* = (x_A + \beta)/(h^* + d_A + x_A)$.
6) Calculates $C^* = H_3(V^*) \oplus (m||S^*)$.
7) Returns $\delta^* = (S^*, C^*, T)$ and sends it to B.

By obtaining $\delta^* = (S^*, C^*, T)$ from the channel, $\mathcal{A}_I$ can easily calculate:

$$V^* = (x_B^* + r_B^*)T, \tag{1}$$

obtain:

$$m||S^* = H_3(V^*) \oplus C^*,$$

and recover m as the first l_0 bits of $m||S^*$. It is straightforward to check the correctness of Equation (1) as we have:

$$\begin{aligned} V^* &= \beta(X_B^* + R_B^* + h_B^* P_{pub}) \\ &= \beta(x_B^* P - H_1(ID_B, R_B^*) P_{pub} + r_B^* P + H_1(ID_B, R_B^*) P_{pub}) \\ &= \beta(x_B^* P + r_B^* P) = (x_B^* + r_B^*)\beta P = (x_B^* + r_B^*)T. \end{aligned}$$

As a result, $\mathcal{A}_I$ can deceive A to use $PuK_B^* = (R_B^*, X_B^*)$ instead of $PuK_B = (R_B, X_B)$ for generating a signcryption for B, and consequently obtain m and break the confidentiality of the scheme. So, the Gao *et al.*'s CL-SC scheme is not confidential against $\mathcal{A}_I$ in contrast to their claim.

4.2 Cryptanalysis of Gao *et al.*'s Access Control Protocol

Gao *et al.* have claimed that their protocol provides the confidentiality for future communications between Dr. Alice and Bob, i. e. no one can obtain the shared key between them. However, we will show that their claim isn't provided. As explained in the authentication phase of Gao *et al.*'s access control protocol in Section 3.2, upon receiving δ from Dr. Alice, Bob checks it and if it is valid, he obtains m. Then both Bob and Dr. Alice set the session key $H_3(V)$ for their future communications. Now, suppose that a type I adversary $\mathcal{A}_I$ has replaced the real public key of Bob $PuK_B = (R_B, X_B)$ with $PuK_B^* = (R_B^*, X_B^*)$ as explained in Section 4.1. So, Dr. Alice sets $H_3(V^*)$ as the session key for communicating with Bob, where:

$$V^* = \beta(X_B^* + R_B^* + h_B^* P_{pub}).$$

It is obvious that $\mathcal{A}_I$ can obtain the session key $H_3(V^*)$ by computing V^* as follows:

$$V^* = (x_B^* + r_B^*)T.$$

As a result, $\mathcal{A}_I$ can obtain the session key, communicate to Dr. Alice instead of Bob and access to all messages sent from Dr. Alice to Bob during the session. So, the confidentiality and the privacy of Bob will not be preserved at all.

5 Improvement of Gao *et al.*'s Proposals

In this section, we improve Gao *et al.*'s CL-SC scheme [6] to be robust against our proposed attack in Section 4.1. Then we provide the security proof of our improvement. Finally, we fix Gao *et al.*'s access control protocol based on our improved CL-SC scheme.

5.1 The Improved CL-SC Scheme

The algorithms of the improved CL-SC scheme are as follows:

Setup. It is similar to the Setup algorithm of Gao *et al.*'s CL-SC scheme, explained in Section 3.1.

ParKeyGen. In this algorithm, an entity U picks a random $x_U \in_R \mathbb{Z}_q^*$, computes $X_U = x_U P$ and sends X_U and ID_U to the SP. The SP chooses a random value $r_U \in_R \mathbb{Z}_q^*$, calculates $R_U = r_U P$ and $d_U = r_U + \alpha H_1(ID_U, R_U, X_U)$ and sends $ParK_U = (R_U, d_U)$ to U via a secure channel. The user U can verify the correctness of the received partial keys by checking whether the equation $R_U + H_1(ID_U, R_U, X_U)P_{pub} = d_U P$ holds or not.

KeyGen. The entity U sets $PrK_U = (d_U, x_U)$ as his/her private key and $PuK_U = (R_U, X_U)$ as his/her public key.

Signcryption. Suppose that an entity A wants to create a signcryption δ on a message m for an entity B. A executes the following steps:

1) Picks a random value $\beta \in_R \mathbb{Z}_q^*$ and computes $T = \beta P$.
2) Sets $h_B = H_1(ID_B, R_B, X_B)$.
3) Calculates $V = \beta(X_B + R_B + h_B P_{pub})$.
4) Sets $h = H_2(m||T||ID_A||ID_B||X_A||X_B)$.
5) Calculates

$$S = \frac{x_A + \beta}{h + d_A + x_A} = (x_A + \beta)(h + d_A + x_A)^{-1} \bmod q,$$

6) Calculates $C = H_3(V) \oplus (m||S)$.
7) Returns $\delta = (S, C, T)$ and sends it to B.

UnSigncryption. Upon receiving a signcryption $\delta = (S, C, T)$ from A, B executes the following steps to verify δ and obtain m:

1) Calculates $V = (x_B + d_B)T$.
2) Calculates $m||S = H_3(V) \oplus C$ and consequently recovers m as the first l_0 bits of $m||S$.
3) Sets $h = H_2(m||T||ID_A||ID_B||X_A||X_B)$.
4) Sets $h_A = H_1(ID_A, R_A, X_A)$.
5) Verifies the signcryption by checking the following equality:

$$S(X_A + R_A + h_A P_{pub} + hP) = X_A + T.$$

If the above equality holds, B accepts m as a signcryption from A, otherwise B rejects it and returns $\perp$.

Remark 5.1. *Note that in the improved scheme, in the ParKeyGen algorithm, d_U is computed as $d_U = r_U + \alpha H_1(ID_U, R_U, X_U)$ instead of $d_U = r_U + \alpha H_1(ID_U, R_U)$. Consequently, h_B and h_A are computed as $h_B = H_1(ID_B, R_B, X_B)$ and $h_A = H_1(ID_A, R_A, X_A)$ in the Signcryption and UnSigncryption algorithms. As a result, $\mathcal{A}_I$ cannot replace $PuK_B = (R_B, X_B)$ such as our proposed attack in Section 4.1.*

5.2 Analysis of the Improved CL-SC Scheme

5.2.1 Correctness

The correctness of the fixed scheme can be checked easily, as follows:

$$\begin{aligned} &R_U + H_1(ID_U, R_U, X_U)P_{pub} \\ &= r_U P + H_1(ID_U, R_U, X_U)\alpha P \\ &= (r_U + \alpha H_1(ID_U, R_U, X_U))P \\ &= d_U P, \end{aligned}$$

which shows the correctness of the partial key,

$$\begin{aligned} V &= (x_B + d_B)T \\ &= (x_B + r_B + \alpha H_1(ID_B, R_B, X_B))\beta P \\ &= \beta(x_B P + r_B P + H_1(ID_B, R_B, X_B)\alpha P) \\ &= \beta(X_B + R_B + h_B P_{pub}), \end{aligned}$$

which shows the correctness of the computed V in both sides, and:

$$\begin{aligned} &S(X_A + R_A + h_A P_{pub} + hP) \\ &= \frac{x_A + \beta}{h + d_A + x_A}(x_A P + r_A P + H_1(ID_A, R_A, X_A)\alpha P + hP) \\ &= \frac{x_A + \beta}{h + d_A + x_A}(x_A + r_A + H_1(ID_A, R_A, X_A)\alpha + h)P \\ &= \frac{x_A + \beta}{h + d_A + x_A}(x_A + d_A + h)P \\ &= (x_A + \beta)P = x_A P + \beta P = X_A + T, \end{aligned}$$

which shows the correctness of the verification part of the UnSigncryption algorithm.

5.2.2 Confidentiality

It can be shown that the improved CL-SC scheme is confidential (IND-CCA2) against type I and type II adversaries $\mathcal{A}_I$ and $\mathcal{A}_{II}$, in the random oracle model (ROM), based on the DDH assumption. The confidentiality against $\mathcal{A}_I$ and $\mathcal{A}_{II}$ are respectively defined according to Game I and Game II in [15], except that as our proof is provided in ROM, the adversary has access to Hash oracles, too.

Lemma 1. *If there is an adversary $\mathcal{A}_I$ who can win Game I in [15], with a non-negligible advantage ε, one can construct an algorithm $\mathcal{C}$, which can solve an instance of the DDH problem with an advantage at least $\frac{\varepsilon}{(n_q+1)^2}$, where n_q is the number of queries from Partial-Private-Key, Private-Key and Signcrypt oracles.*

Proof. Suppose that the algorithm $\mathcal{C}$ gets an instance $P, aP, bP, X \in G$, of a DDH problem and wants to decide whether $X = abP$ or not. First, $\mathcal{C}$ creates a list $\mathcal{L} = \{ID_U, h_{1,U}, h_{2,A,B,m,T}, T, V, h_{3,T}, d_U, x_U, r_U, X_U, R_U, c_U, ParK_U, PuK_U, PrK_U)\}$ which is initially empty. Then $\mathcal{C}$ plays Game I in [15] with $\mathcal{A}_I$ as follows:

Initialization: Given a security parameter k, $\mathcal{C}$ sets $P_{pub} = aP$. Then it produces other system parameters such that explained in the Setup algorithm of the improved scheme and sends $params = \{G, q, P, P_{pub}, H_1, H_2, H_3\}$ to $\mathcal{A}_I$. Note that as $P_{pub} = aP$, $\mathcal{C}$ does not know the corresponding master secret key $\alpha = a$.

Phase 1 Queries: $\mathcal{A}_I$ sends polynomially bounded number of queries to the Hash, Public-Key, Partial-Private-Key, Replace-Public-Key, Private-Key, Signcrypt and Unsigncrypt oracles and $\mathcal{C}$ responds to these queries as follows:

- H_1 queries: Receiving a $H_1(ID_U, R_U, X_U)$ query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether $h_{1,U}$ exists in $\mathcal{L}$ or not. If so, $\mathcal{C}$ picks it and sends it to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly picks $c_U \in_R \{0,1\}$ such that $\Pr[c_U = 1] = \frac{1}{n_q+1}$ [6]. Then $\mathcal{C}$ acts as follows:
  - If $c_U = 0$, $\mathcal{C}$ randomly selects $h_{1,U} \in_R \mathbb{Z}_q^*$, sends it to $\mathcal{A}_I$ and inserts $c_U = 0$ and $h_{1,U}$ in $\mathcal{L}$.
  - If $c_U = 1$, $\mathcal{C}$ sets $h_{1,U} = K$ (a constant value), sends it to $\mathcal{A}_I$ and inserts $c_U = 1$ in $\mathcal{L}$.
- H_2 queries: Receiving a $H_2(m||T||ID_A||ID_B||X_A||X_B)$ query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether $h_{2,A,B,m,T}$ exists in $\mathcal{L}$ or not. If so, $\mathcal{C}$ picks and sends it to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly selects $h_{2,A,B,m,T} \in_R \mathbb{Z}_q^*$, sends it to $\mathcal{A}_I$ and inserts it in $\mathcal{L}$.
- H_3 queries: Receiving a $(H_3(V), T)$ query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether $h_{3,T}$ exists in $\mathcal{L}$ or not. If so, $\mathcal{C}$ picks and sends it to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly selects $h_{3,T} \in_R \{0,1\}^{l_0+|\mathbb{Z}_q^*|}$, sends it to $\mathcal{A}_I$ and inserts $T, V, h_{3,T}$ in $\mathcal{L}$.
- Public-Key queries: Receiving a PuK_U query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether PuK_U exists in $\mathcal{L}$ or not. If so, $\mathcal{C}$ picks and sends it to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ checks c_U in $\mathcal{L}$ and acts as follows:
  - If $c_U = 0$, $\mathcal{C}$ picks random values $x_U, r_U, z \in_R \mathbb{Z}_q^*$, computes $R_U = r_U P$, $X_U = x_U P$, $d_U = r_U + zH_1(ID_U, R_U, XU)$, sends $Puk_U = (R_U, X_U)$ to $\mathcal{A}_I$. Then $\mathcal{C}$ inserts $d_U, x_U, X_U, R_U, ParK_U = (R_U, d_U), PuK_U = (R_U, X_U), PrK_U = (d_U, x_U)$ in $\mathcal{L}$.
  - If $c_U = 1$, $\mathcal{C}$ randomly selects $x_U, r_U \in_R \mathbb{Z}_q^*$, computes $R_U = r_U P$ and $X_U = x_U P$, sets $PuK_U = (R_U, X_U)$, sends PuK_U to $\mathcal{A}_I$ and inserts $x_U, r_U, X_U, R_U, PuK_U = (R_U, X_U)$ in $\mathcal{L}$.
- Partial-Private-Key queries: Receiving a $ParK_U$ query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether $ParK_U$ exists in $\mathcal{L}$ or not. If so, $\mathcal{C}$ picks and sends it to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ checks c_U in $\mathcal{L}$ and acts as follows:
  - If $c_U = 0$, $\mathcal{C}$ runs a Public-Key query as explained. Then $\mathcal{C}$ returns $ParK_U$ to $\mathcal{A}_I$.
  - If $c_U = 1$, $\mathcal{C}$ aborts the simulation.
- Private-Key queries: Receiving a PrK_U query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether PrK_U exists in $\mathcal{L}$ or not. If so, $\mathcal{C}$ picks and sends it to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ checks c_U in $\mathcal{L}$ and acts as follows:
  - If $c_U = 0$, $\mathcal{C}$ runs a Public-Key query as explained. Then $\mathcal{C}$ returns PrK_U to $\mathcal{A}_I$.
  - If $c_U = 1$, $\mathcal{C}$ aborts the simulation.
- Replace-Public-Key queries: When $\mathcal{A}_I$ wants to replace a public key $PuK_U = (R_U, X_U)$ with a new public key $PuK'_U = (R'_U, X'_U)$, $\mathcal{C}$ applies this query and replaces PuK_U with PuK'_U in $\mathcal{L}$.
- Signcrypt queries: When $\mathcal{A}_I$ sends a signcrypt query on (ID_A, ID_B, m) to the Signcrypt oracle, $\mathcal{C}$ checks c_A and acts as follows:
  - If $c_A = 0$, $\mathcal{C}$ picks PrK_A from $\mathcal{L}$. Note that if PrK_A does not exist in $\mathcal{L}$, $\mathcal{C}$ can obtain it by a Private-Key query as explained before. Then $\mathcal{C}$ runs the Signcryption algorithm of the improved scheme to produce the signcryption δ on m from A to B and sends it to $\mathcal{A}_I$.
  - If $c_A = 1$, $\mathcal{C}$ aborts the simulation.
- Unsigncrypt queries: When $\mathcal{A}_I$ sends an Unsigncrypt query on $(ID_A, ID_B, \delta = (S, C, T))$ to the Unigncrypt oracle, $\mathcal{C}$ checks c_B and acts as follows:
  - If $c_B = 0$, $\mathcal{C}$ picks PrK_B from $\mathcal{L}$. Note that if PrK_B does not exist in $\mathcal{L}$, $\mathcal{C}$ can obtain it by a Private-Key query as explained before. Then $\mathcal{C}$ runs the UnSigncryption algorithm of the improved scheme to obtain m and sends it to $\mathcal{A}_I$.
  - If $c_B = 1$, $\mathcal{C}$ checks all the values of $h_{3,T}$ stored in $\mathcal{L}$ one by one to compute $m||S = H_3(V) \oplus C$ and obtain m. Then $\mathcal{C}$ picks the corresponding $h_{1,A}$ and $h_{2,A,B,m,T}$ (For each T) from $\mathcal{L}$ and verifies whether the equation $S(X_A + R_A + h_{1,A}P_{pub} + h_{2,A,B,m,T}P) = X_A + T$ holds or not. If there exist a $h_{3,T}$, for which this equation holds, $\mathcal{C}$ returns the corresponding m to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ returns $\perp$ to $\mathcal{A}_I$.

Challenge: In this step, $\mathcal{A}_I$ sends two equal lengths messages m_0 and m_1 and two identities ID_{A^*} and ID_{B^*} to $\mathcal{C}$. $\mathcal{C}$ first checks c_{B^*} in $\mathcal{L}$ and acts as follows:

- If $c_{B^*} = 0$, $\mathcal{C}$ aborts the simulation.
- If $c_{B^*} = 1$, $\mathcal{C}$ sets $T^* = bP$. Then $\mathcal{C}$ obtains $(T^*, V^*, H_3(V^*))$ from $\mathcal{L}$, chooses random values $S^* \in_R \mathbb{Z}_q^*$ and $\gamma^* \in_R \{0,1\}$, sets $C^* = H_3(V^*) \oplus (m_{\gamma^*}||S^*)$ and sends $\delta^* = (S^*, C^*, T^*)$ to $\mathcal{A}_I$.

Phase 2 Queries: $\mathcal{A}_I$ can again send polynomially bounded number of queries similar to that explained in Phase 1 Queries and $\mathcal{C}$ responds to these queries such explained.

Guess: In this step, $\mathcal{A}_I$ returns a guess $\gamma' \in \{0,1\}$ of γ^*.

At the end of the game, $\mathcal{C}$ acts as follows:

- If the simulation is aborted in any steps, $\mathcal{C}$ randomly selects $\gamma \in_R \{0,1\}$ as its guess of the answer to the DDH problem.

- Otherwise, if $\gamma' = \gamma^*$, $\mathcal{C}$ retrieves x_{B^*} and r_{B^*} from $\mathcal{L}$. Note that as the simulation is not aborted in the Challenge step, we have $c_{B^*} = 1$, so $\mathcal{C}$ can retrieve x_{B^*} and r_{B^*}. Furthermore, we have $h_{1,B^*} = K$, as $c_{B^*} = 1$. Then $\mathcal{C}$ obtains $(T^*, V^*, H_3(V^*))$ from $\mathcal{L}$ and checks whether the equation:

$$\frac{V^* - (x_{B^*} + r_{B^*})T^*}{K} = X, \tag{2}$$

holds or not. If so, $\mathcal{C}$ returns $\gamma = 1$, otherwise it returns $\gamma = 0$, as its answer to the DDH problem.

Note that as $c_{B^*} = 1$, $\mathcal{C}$ does not know $d_{B^*} = r_{B^*} + \alpha H_1(ID_{B^*}, R_{B^*}, X_{B^*})$, as $P_{pub} = aP$ and $\alpha = a$ is unknown to $\mathcal{C}$. Moreover, remember that $\mathcal{C}$ sets $T^* = bP$ in the Challenge step which indicates that $\beta = b$ which is also unknown to $\mathcal{C}$. In this case, if δ^* is actually a valid signcryption on m_{γ^*}, we have:

$$\begin{aligned}
&\frac{V^* - (x_{B^*} + r_{B^*})T^*}{K} \\
&= \frac{\beta(X_{B^*} + R_{B^*} + h_{1,B^*}P_{pub}) - (x_{B^*} + r_{B^*})bP}{K} \\
&= \frac{b(X_{B^*} + R_{B^*} + KaP) - b(X_{B^*} + R_{B^*})}{K} \\
&= abP,
\end{aligned}$$

So, if Equation (2) holds, it is implied that $X = abP$, then $\mathcal{C}$ returns $\gamma = 1$. Otherwise, it returns $\gamma = 0$ as its answer to the DDH problem.

Probability Analysis: Suppose that $\Pr[\mathcal{C} \text{ wins}]$ is the success probability of $\mathcal{C}$ to solve the DDH problem and $\Pr[\mathcal{A}_I \text{ wins}]$ is the success probability of $\mathcal{A}_I$ in the above game. Note that if the simulation is aborted in any steps, $\mathcal{C}$ randomly selects its guess $\gamma \in_R \{0,1\}$ as its answer to the DDH problrm, so $\Pr[\mathcal{C} \text{ wins}] = \frac{1}{2}$. If the advantage of $\mathcal{A}_I$ in winning the game is ε, i. e. $\Pr[\mathcal{A}_I \text{ wins}] \geq \frac{1}{2} + \varepsilon$, we have:

$$\begin{aligned}
\Pr[\mathcal{C} \text{ wins}] &= \Pr[\mathcal{C} \text{ wins}|\text{abort}]\Pr[\text{abort}] \\
&\quad + \Pr[\mathcal{C} \text{ wins}|\overline{\text{abort}}]\Pr[\overline{\text{abort}}] \\
&= \frac{1}{2}\Pr[\text{abort}] + \Pr[\mathcal{A}_I \text{ wins}]\Pr[\overline{\text{abort}}] \\
&\geq \frac{1}{2}(1 - \Pr[\overline{\text{abort}}]) + (\frac{1}{2} + \varepsilon)\Pr[\overline{\text{abort}}] \\
&= \frac{1}{2} + \varepsilon\Pr[\overline{\text{abort}}]
\end{aligned}$$

On the other hand, $\mathcal{C}$ will not abort if all the following independent events happen:

- E_1**:** $c_U = 0$ in all Partial-Private-Key and Private-Key queries.

- E_2**:** $c_A = 0$ in all Signcrypt queries.

- E_3**:** $c_{B^*} = 1$ in the Challenge step.

Defining E_i as the event of $c_U = 1$ in the i'th query and noting $\Pr[c_U = 1] = \frac{1}{n_q+1}$, we have:

$$\Pr[E_i] = \Pr[c_U = 1] = \frac{1}{n_q + 1}.$$

So we have:

$$\Pr[E_3] = \Pr[E_i] = \frac{1}{n_q + 1},$$

and:

$$\begin{aligned}
\Pr[E_1 \bigcap E_2] &= \Pr[\bigcap_{i=1}^{n_q} \bar{E}_i] = 1 - \Pr[\bigcup_{i=1}^{n_q} E_i] \\
&\geq 1 - \sum_{i=1}^{n_q} \Pr[E_i] = 1 - \frac{n_q}{n_q + 1}.
\end{aligned}$$

Therefore:

$$\begin{aligned}
\Pr[\overline{\text{abort}}] &\geq \Pr[E_1 \bigcap E_2 \bigcap E_3] = \Pr[E_1 \bigcap E_2].\Pr[E_3] \\
&\geq (1 - \frac{n_q}{n_q + 1})(\frac{1}{n_q + 1}) = \frac{1}{(n_q + 1)^2}
\end{aligned}$$

Finally we have:

$$\Pr[\mathcal{C} \text{ wins}] \geq \frac{1}{2} + \frac{\varepsilon}{(n_q + 1)^2}$$

In summary, if $\mathcal{A}_I$ wins the game with a non-negligible advantage ε (i. e. guesses γ' correctly with probability at least $\frac{1}{2} + \varepsilon$ for a non-negligible value of ε), then $\mathcal{C}$ can solve an instance of the DDH problem with a non-negligible advantage ε' (i. e. guess γ correctly with probability at least $\frac{1}{2} + \varepsilon'$), where $\varepsilon' \geq \frac{\varepsilon}{(n_q+1)^2}$ which is a contradiction with the DDH assumption in complexity theory. □

Lemma 2. *If there is an adversary $\mathcal{A}_{II}$ who can win Game II in [15], with a non-negligible advantage ε, one can construct an algorithm $\mathcal{C}$, which can solve an instance of the DDH problem with an advantage at least $\frac{\varepsilon}{(n_q+1)^2}$, where n_q is the number of queries from Private-Key and Signcrypt oracles.*

Proof. Suppose that the algorithm $\mathcal{C}$ gets an instance $P, aP, bP, X \in G$, of a DDH problem and wants to decide whether $X = abP$ or not. First, $\mathcal{C}$ creates a list $\mathcal{L} = \{ID_U, h_{1,U}, h_{2,A,B,m,T}, T, V, h_{3,T}, x_U, X_U, c_U, PuK_U, PrK_U)\}$ which is initially empty. Then $\mathcal{C}$ plays Game II in [15] with $\mathcal{A}_{II}$ as follows:

Initialization: Given a security parameter k, $\mathcal{C}$ generates system parameters such that explained in the Setup algorithm of the improved scheme and sends $params = \{G, q, P, P_{pub}, H_1, H_2, H_3\}$ to $\mathcal{A}_{II}$. Note that $\mathcal{C}$ knows the master secret key α, here.

Phase 1 Queries: $\mathcal{A}_{II}$ sends polynomially bounded number of queries to the Hash, Public-Key, Private-Key, Signcrypt and Unsigncrypt oracles and $\mathcal{C}$ responds to these queries as follows:

- H_1, H_2 and H_3 queries: $\mathcal{C}$ responds to these queries similar to that explained in the proof of Lemma 1.
- Public-Key queries: Receiving a PuK_U query from $\mathcal{A}_{II}$, $\mathcal{C}$ first checks whether PuK_U exists in $\mathcal{L}$ or not. If so, $\mathcal{C}$ picks and sends it to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}$ checks c_U in $\mathcal{L}$ and acts as follows:
  - If $c_U = 0$, $\mathcal{C}$ picks random values $x_U, r_U \in_R \mathbb{Z}_q^*$, computes $R_U = r_U P$, $X_U = x_U P$, $d_U = r_U + \alpha H_1(ID_U, R_U, XU)$, sends $Puk_U = (R_U, X_U)$ to $\mathcal{A}_{II}$. Then $\mathcal{C}$ inserts $PuK_U = (R_U, X_U), PrK_U = (d_U, x_U)$ in $\mathcal{L}$.
  - If $c_U = 1$, $\mathcal{C}$ sets $R_U = aP$. then it randomly selects $x_U \in_R \mathbb{Z}_q^*$, computes $X_U = x_U P$, sets $PuK_U = (R_U, X_U)$, sends PuK_U to $\mathcal{A}_{II}$ and inserts $x_U, X_U, PuK_U = (R_U, X_U)$ in $\mathcal{L}$.
- Private-Key, Signcrypt and Unsigncrypt queries: $\mathcal{C}$ responds to these queries similar to that explained in the proof of Lemma 1.

Challenge: This step is also similar to that in the proof of Lemma 1.

Phase 2 Queries: This step is also similar to that in the proof of Lemma 1.

Guess: In this step, $\mathcal{A}_I$ returns a guess $\gamma' \in \{0,1\}$ of γ^*.

At the end of the game, $\mathcal{C}$ acts as follows:

- If the simulation is aborted in any steps, $\mathcal{C}$ randomly selects $\gamma \in_R \{0,1\}$ as its guess of the answer to the DDH problem.
- Otherwise, if $\gamma' = \gamma^*$, $\mathcal{C}$ retrieves x_{B^*} from $\mathcal{L}$. Note that as the simulation is not aborted in the Challenge step, we have $c_{B^*} = 1$ and so $h_{1,B^*} = K$. Then $\mathcal{C}$ obtains $(T^*, V^*, H_3(V^*))$ from $\mathcal{L}$ and checks whether the equation:

$$V^* - (x_{B^*} + K\alpha)T^* = X, \tag{3}$$

holds or not. If so, $\mathcal{C}$ returns $\gamma = 1$, otherwise it returns $\gamma = 0$, as its answer to the DDH problem.

Note that $\mathcal{C}$ sets $T^* = bP$ in the Challenge step. So, $\beta = b$ which is unknown to $\mathcal{C}$. Moreover, as $c_{B^*} = 1$, we have $h_{1,B^*} = K$ and $R_{B^*} = aP$. In this case, if δ^* is actually a valid signcryption on m_{γ^*}, we have:

$$\begin{aligned} &V^* - (x_{B^*} + K\alpha)T^* \\ &= \beta(X_{B^*} + R_{B^*} + h_{1,B^*}P_{pub}) - (x_{B^*} + K\alpha)bP \\ &= b(X_{B^*} + aP + K\alpha P) - b(X_{B^*} + K\alpha P) \\ &= abP, \end{aligned}$$

So, if Equation (3) holds, it is implied that $X = abP$, then $\mathcal{C}$ returns $\gamma = 1$. Otherwise, it returns $\gamma = 0$ as its answer to the DDH problem.

Probability Analysis: It is similar to that explained in the proof of Lemma 1, except that the number of Partial-Private-Key and Replace-Public-Key queries are 0 here. □

Theorem 1. *The improved CL-SC scheme is confidential (IND-CCA2) against $\mathcal{A}_I$ and $\mathcal{A}_{II}$ based on the DDH assumption.*

Proof. the proof is directly implied from Lemma 1 and Lemma 2. □

5.2.3 Unforgeability

It can be shown that the improved CL-SC scheme is unforgeable (EUF-CMA) against type I and type II adversaries $\mathcal{A}_I$ and $\mathcal{A}_{II}$, in the random oracle model (ROM), based on the DL assumption. The unforgeability against $\mathcal{A}_I$ and $\mathcal{A}_{II}$ are respectively defined according to Game III and Game IV in [15], except that as our proof is provided in ROM, the adversary has access to Hash oracles, too.

Lemma 3. *If there is an adversary $\mathcal{A}_I$ who can win Game III in [15], with a non-negligible advantage ε, one can construct an algorithm $\mathcal{C}$, which can solve an instance of the DL problem with an advantage at least $\frac{\varepsilon p_{frk}}{(n_q+1)^2}$, where n_q is the number of queries from Partial-Private-Key, Private-Key and Signcrypt oracles and p_{frk} is the success probability of the adversary in Forking Lemma [6].*

Proof. Suppose that the algorithm $\mathcal{C}$ gets an instance $P, aP \in G$, of a DL problem and wants to obtain $a \in \mathbb{Z}_q^*$. First, $\mathcal{C}$ creates a list $\mathcal{L}$ such that explained in the proof of Lemma 1, which is initially empty. Then $\mathcal{C}$ plays Game III in [15] with $\mathcal{A}_I$ as follows:

Initialization: This step is similar to that in the proof of Lemma 1.

Queries: This step is similar to the Phase 1 Queries step in the proof of Lemma 1

Output: After a polynomially bounded number of queries, $\mathcal{A}_I$ outputs a valid signcryption $\delta^* = (S^*, C^*, T^*)$ on a message m^* from A^* to B^*.

At the end of the game, $\mathcal{C}$ acts as follows:

- If the simulation is aborted in any steps or $c_{A^*} = 0$, $\mathcal{C}$ aborts.
- Otherwise, $\mathcal{C}$ obtains m^* such that explained in the Unsigncrypt queries in the proof of Lemma 1. If δ^* is a valid signcryption on m^*, according to the Froking Lemma [6], $\mathcal{C}$ can get two valid signcryptions from A^* to B^* on m^* with the same random value β and different values of the random oracle h_{2,A^*,B^*,m^*,T^*}. So, $\mathcal{C}$ gets these two valid signcryptions with the same $T^* = \beta P$ and different values of $h = h_{2,A^*,B^*,m^*,T^*}$

and $h' = h'_{2,A^*,B^*,m^*,T^*}$. Denote these two valid signcryptions by $\delta_1 = (S_1, C_1, T^*)$ and $\delta_2 = (S_2, C_2, T^*)$. According to the step 5 of the signcryption algorithm, we have:

$$S_1(h + d_{A^*} + x_{A^*}) = x_{A^*} + \beta \bmod q,$$
$$S_2(h' + d_{A^*} + x_{A^*}) = x_{A^*} + \beta \bmod q.$$

So, we have:

$$S_1(h + d_{A^*} + x_{A^*}) = S_2(h' + d_{A^*} + x_{A^*}) \bmod q.$$

Note that as $c_{A^*} = 1$, $h_{1,A^*} = K$ and $d_{A^*} = r_{A^*} + \alpha h_{1,A^*} = r_{A^*} + aK$. Moreover, $P_{pub} = aP$ and the master secret key $\alpha = a$ is unknown to $\mathcal{C}$. So, we have:

$$S_1(h + r_{A^*} + aK + x_{A^*}) = S_2(h' + r_{A^*} + aK + x_{A^*}) \bmod q.$$

In the above equation all values except a is known to $\mathcal{C}$. So $\mathcal{C}$ can obtain a as its answer to the DL problem.

Probability Analysis: Suppose that Pr[$\mathcal{C}$ wins] is the success probability of $\mathcal{C}$ to solve the DL problem and Pr[$\mathcal{A}_I$ wins] is the success probability of $\mathcal{A}$ in the above game. If the advantage of $\mathcal{A}_I$ in winning the game is ε, i. e. Pr[$\mathcal{A}_I$ wins] $\geq \varepsilon$, we have:

$$\begin{aligned}\Pr[\mathcal{C}\text{ wins}] &= \Pr[\overline{\text{abort}} \bigcap \mathcal{A}_I \text{wins}] \\ &= \Pr[\overline{\text{abort}}].\Pr[\mathcal{A}_I \text{wins}] \\ &\geq \varepsilon.\Pr[\overline{\text{abort}}]\end{aligned}$$

On the other hand, $\mathcal{C}$ will not abort if all the following independent events happen:

- E_1**:** $c_U = 0$ in all Partial-Private-Key and Private-Key queries.
- E_2**:** $c_A = 0$ in all Signcrypt queries.
- E_3**:** $c_{A^*} = 1$.
- E_4**:** $\mathcal{C}$ can get two valid signcryptions with the same random tape and deifferent values of random oracles in Forking Lemma.

Similar to the explanations in the probability analysis of the proof of Lemma 1, we have:

$$\Pr[\overline{\text{abort}}] \geq \Pr[E_1 \bigcap E_2 \bigcap E_3 \bigcap E_4] \geq \frac{p_{frk}}{(n_q+1)^2},$$

So:

$$\Pr[\mathcal{C}\text{ wins}] \geq \frac{\varepsilon p_{frk}}{(n_q+1)^2}$$

In summary, if $\mathcal{A}_I$ wins the game with a non-negligible advantage ε (i. e. forges a valid signcryption with probability at least ε for a non-negligible value of ε), then $\mathcal{C}$ can solve an instance of the DL problem with a non-negligible advantage ε' (i. e. obtains a with probability at least ε'), where $\varepsilon' \geq \frac{\varepsilon p_{frk}}{(n_q+1)^2}$ which is a contradiction with the DL assumption in complexity theory. □

Lemma 4. *If there is an adversary $\mathcal{A}_{II}$ who can win Game IV in [15], with a non-negligible advantage ε, one can construct an algorithm $\mathcal{C}$, which can solve an instance of the DL problem with an advantage at least $\frac{\varepsilon p_{frk}}{(n_q+1)^2}$, where n_q is the number of queries from Private-Key and Signcrypt oracles and p_{frk} is the success probability of the adversary in Forking Lemma [6].*

Proof. Suppose that the algorithm $\mathcal{C}$ gets an instance $P, aP \in G$, of a DL problem and wants to obtain a.. First, $\mathcal{C}$ creates a list $\mathcal{L}$ such that explained in the proof of Lemma 2, which is initially empty. Then $\mathcal{C}$ plays Game IV in [15] with $\mathcal{A}_{II}$ as follows:

Initialization: This step is similar to that in the proof of Lemma 2.

Queries: This step is similar to the Phase 1 Queries step in the proof of Lemma 2.

Output: After a polynomially bounded number of queries, $\mathcal{A}_{II}$ outputs a valid signcryption $\delta^* = (S^*, C^*, T^*)$ on a message m^* from A^* to B^*.

At the end of the game, $\mathcal{C}$ acts as follows:

- If the simulation is aborted in any steps or $c_{A^*} = 0$, $\mathcal{C}$ aborts.
- Otherwise, $\mathcal{C}$ obtains m^* such that explained in the Unsigncrypt queries in the proof of Lemma 2. If δ^* is a valid signcryption on m^*, according to the Forking Lemma [6], $\mathcal{C}$ can get two valid signcryptions from A^* to B^* on m^* with the same random value β and different values of the random oracle h_{2,A^*,B^*,m^*,T^*}. So, $\mathcal{C}$ gets these two valid signcryptions with the same $T^* = \beta P$ and different values of $h = h_{2,A^*,B^*,m^*,T^*}$ and $h' = h'_{2,A^*,B^*,m^*,T^*}$. Denote these two valid signcryptions by $\delta_1 = (S_1, C_1, T^*)$ and $\delta_2 = (S_2, C_2, T^*)$. According to the step 5 of the signcryption algorithm, we have:

$$S_1(h + d_{A^*} + x_{A^*}) = x_{A^*} + \beta \bmod q,$$
$$S_2(h' + d_{A^*} + x_{A^*}) = x_{A^*} + \beta \bmod q.$$

So, we have:

$$S_1(h + d_{A^*} + x_{A^*}) = S_2(h' + d_{A^*} + x_{A^*}) \bmod q.$$

Note that as $c_{A^*} = 1$, $h_{1,A^*} = K$ and $d_{A^*} = r_{A^*} + \alpha h_{1,A^*} = r_{A^*} + \alpha K$. Moreover, the master secret key α is known to $\mathcal{C}$, but as $R_{A^*} = aP$, $r_{A^*} = a$ is unknown to $\mathcal{C}$. So, we have:

$$S_1(h + a + \alpha K + x_{A^*}) = S_2(h' + a + \alpha K + x_{A^*}) \bmod q.$$

In the above equation all values except a is known to $\mathcal{C}$. So $\mathcal{C}$ can obtain a as its answer to the DL problem.

Table 1: Comparison of the Gao *et al.*'s scheme and our improvement

Scheme	Conf. Against $\mathcal{A}_I$	Conf. Against $\mathcal{A}_{II}$	Unf. Against $\mathcal{A}_I$	Unf. Against $\mathcal{A}_{II}$	Secrecy of the Shared Key
[6]	×	✓	✓	✓	×
Ours	✓	✓	✓	✓	✓

Probability Analysis: It is similar to that explained in the proof of Lemma 3, except that the number of Partial-Private-Key and Replace-Public-Key queries are 0 here. □

Theorem 2. *The improved CL-SC scheme is unforgeable (EUF-CMA) against $\mathcal{A}_I$ and $\mathcal{A}_{II}$ based on the DL assumption.*

Proof. the proof is directly implied from Lemma 3 and Lemma 4. □

5.3 The Improved Access Control Protocol

If our improved CL-SC scheme is used in Gao *et al.*'s access control protocol which is explained in Section 3.2, it will be robust against our attack in Section 4.2, as in the fixed scheme, $\mathcal{A}_I$ can not replace PuK_B to obtain the session key $H_3(V)$, according to Remark 5.1. Figure 3 shows the access control protocol, based on the improved CL-SC scheme. It should be noted that in the improved protocol, Dr. Alice and Bob must send X_A and X_B (in addition to ID_A and ID_B) to the SP to get their partial keys $ParK_A$ and $ParK_B$.

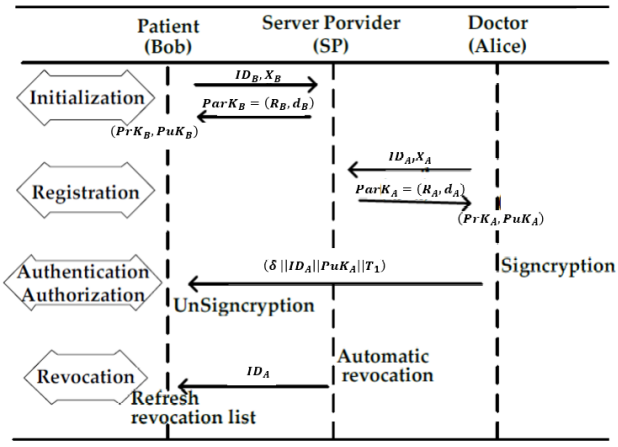


Figure 3: The improved access control protocol

6 Comparison

Table 1 provides a security comparison between the Gao *et al.*'s proposals and our improvements. As shown in Table 1, Gao *et al.*'s CL-SC scheme is not confidential against a key replacement attacker $\mathcal{A}_I$ and consequently the shared key will reveal in their proposed access control protocol and the secrecy of the shared key will not be guaranteed in their protocol. In our improvement, we fixed their CL-SC scheme to be confidential against $\mathcal{A}_I$ and consequently the secrecy of the shared key will be guaranteed in the access control protocol based on the improved CL-SC scheme. It is so important to note that this enhancement will not force any more computational and communications costs on Gao *et al.*'s proposals, as we have just replaced $H_1(ID_U, R_U)$ in Gao *et al.*'s proposal with $H_1(ID_U, R_U, X_U)$ to protect the improved scheme against the proposed attacks, which does not force any additional computational and communications costs.

7 Conclusion

In this work, we cryptanalyzed a recently proposed access control protocol for WBANs proposed by Gao *et al.* They first proposed a certificateless signcryption (CL-SC) scheme in the random oracle model (ROM) and claimed that their scheme is confidential (IND-CCA2) and unforgeable (EUF-CMA) against type I and type II adversaries, in the certificateless setting. Consequently, they designed an access control protocol for WBANs in which the physician (Dr. Alice) sends a signcrypted request message to the patient (Bob) and if she is authenticated to Bob, they establish a session key for their next communications. However, we showed that, in contrast to their claim, Gao *et al.*'s CL-SC scheme is not confidential against the type I adversary (a key replacement attacker) and consequently, this adversary can obtain the session key which is established by the physician for the next communications to the patient. Moreover, we fixed Gao *et al.*'s CL-SC scheme to be robust against our proposed attack. It is notable that the access control protocol based on our improved CL-SC scheme will not be vulnerable against the designed attack, too.

References

[1] E. Ahene, Z. Qin, A. K. Adusei, and F. Li, "Efficient signcryption with proxy re-encryption and its application in smart grid," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9722–9737, 2019.

[2] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *International Confer-*

ence on the Theory and Application of Cryptology and Information Security, pp. 452–473, 2003.

[3] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*, pp. 369–372, 2008.

[4] S. Belguith, N. Kaaniche, M. Hammoudeh, and T. Dargahi, "Proud: Verifiable privacy-preserving outsourced attribute based signcryption supporting access policy update for cloud assisted IoT applications," *Future Generation Computer Systems*, vol. 111, pp. 899–918, 2020.

[5] Z. Caixue, "Certificateless signcryption scheme without random oracles," *Chinese Journal of Electronics*, vol. 27, no. 5, pp. 1002–1008, 2018.

[6] G. Gao, X. Peng, and L. Jin, "Efficient access control scheme with certificateless signcryption for wireless body area networks." *International Journal Network Security*, vol. 21, no. 3, pp. 428–437, 2019.

[7] M. S. Hajar, M. O. Al-Kadri, and H. K. Kalutarage, "A survey on wireless body area networks: architecture, security challenges and research opportunities," *Computers & Security*, p. 102211, 2021.

[8] P. Kasyoka, M. Kimwele, and S. M. Angolo, "Towards an efficient certificateless access control scheme for wireless body area networks," *Wireless Personal Communications*, vol. 115, no. 2, pp. 1257–1275, 2020.

[9] M. Kumar and S. Chand, "Medhypchain: A patient-centered interoperability hyperledger-based medical healthcare system: Regulation in covid-19 pandemic," *Journal of Network and Computer Applications*, vol. 179, p. 102975, 2021.

[10] X. Liu, Z. Wang, Y. Ye, and F. Li, "An efficient and practical certificateless signcryption scheme for wireless body area networks," *Computer Communications*, vol. 162, pp. 169–178, 2020.

[11] S. Mandal, B. Bera, A. K. Sutrala, A. K. Das, K.-K. R. Choo, and Y. Park, "Certificateless-signcryption-based three-factor user access control scheme for iot environment," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3184–3197, 2020.

[12] V. S. Naresh, S. Reddi, S. Kumari, V. D. Allavarpu, S. Kumar, and M.-H. Yang, "Practical identity based online/off-line signcryption scheme for secure communication in internet of things," *IEEE Access*, vol. 9, pp. 21 267–21 278, 2021.

[13] B. Narwal and A. K. Mohapatra, "A survey on security and authentication in wireless body area networks," *Journal of Systems Architecture*, vol. 113, p. 101883, 2021.

[14] P. Rastegari, "On the security of some recently proposed certificateless signcryption schemes," in *17th International ISC Conference on Information Security and Cryptology (ISCISC'20)*, IEEE, pp. 95–100, 2020.

[15] P. Rastegari and M. Berenjkoub, "An efficient certificateless signcryption scheme in the standard model," *ISeCure*, vol. 9, no. 1, 2017.

[16] P. Rastegari and M. Dakhilalian, "Cryptanalysis of a certificateless signcryption scheme," in *16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC'19)*, IEEE, pp. 67–71, 2019.

[17] P. Rastegari and M. Khalili, "Cryptanalysis and improvement of an access control protocol for wireless body area networks," in *18th International ISC Conference on Information Security and Cryptology (ISCISC'21)*, 2021.

[18] P. Rastegari, W. Susilo, and M. Dakhlalian, "Efficient certificateless signcryption in the standard model: Revisiting luo and wan's scheme from wireless personal communications (2018)," *The Computer Journal*, vol. 62, no. 8, pp. 1178–1193, 2019.

[19] S. Shan, "An efficient certificateless signcryption scheme without random oracles," *International Journal of Electronics and Information Engineering*, vol. 11, no. 1, pp. 9–15, 2019.

[20] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) << cost (signature)+ cost (encryption)," in *Annual international cryptology conference*, Springer, pp. 165–179, 1997.

[21] T. G. Zimmerman, "Personal area networks: Near-field intrabody communication," *IBM Systems Journal*, vol. 35, no. 3.4, pp. 609–617, 1996.

Biography

Parvin Rastegari received the B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from Isfahan University of Technology, Isfahan, Iran, in 2008, 2011 and 2019, respectively. Since 2020, she has been with the Electrical and Computer Engineering Group, Golpayegan College of Engineering, Isfahan University of Technology, Golpayegan, Iran, as an assistant professor. Her current research interests include cryptographic primitives and protocols.

Mojtaba Khalili received a Ph.D. degree in electrical engineering from the department of electrical and computer engineering, Isfahan University of Technology in 2019. He is currently an assistant professor at the K.N Toosi University. His research interests are cryptographic primitives and protocols.

Ali Sakhaei received the B.S and M.S degrees from Shahid Beheshti University, Tehran, Iran in 2004 and 2006 respectively, and the Ph.D. degree from Payame Noor University, Tehran, Iran in 2018. He is currently an Assistant Professor with the statistics Department, Basic Sciences College, Payame Noor University, Tehran, Iran. His current research interests include stochastic process, Actuarial mathematics and Non-life insurance.

An Abnormal Login Detection Method Based on Local Outlier Factor and Gaussian Mixture Model

Wei Guo[1], Yue He[1], He-Xiong Chen[1], Fei-Lu Hang[1], and Yun-Jie Li[2]
(Corresponding author: Yun-Jie Li)
Information Center of Yunnan Power Grid Company Limited, Kunming,Yunnan 650034, China[1]
Network and Data Security Key Laboratory of Sichuan Province, University of Electronic[2]
Science and Technology of China, Chengdu,Sichuan 610054, China
Email: 1318743258@qq.com
(Received Aug. 15, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)

Abstract

Nowadays, various enterprises have gradually begun to provide Internet services for customers. And the security of customer accounts is constantly threatened by various factors, which may result in account password disclosure and theft. Therefore, it is an important task to identify and screen users' abnormal login behavior to protect the customers' interests and the servers' security. At the same time, it becomes a key security problem to extract important features from the server's user login data and detect abnormal login behaviors. This paper proposes a method that extracts features from the login log. It reduces the dimensions of data, then calculates the outlier score of time data and IP address data after dimension reduction by adopting the local outlier factor algorithm and Gaussian mixture model, and comprehensively analyzes outlier scores. The experimental results show that this method performs high accuracy, precision ratios, and so on, around 95%, which means it can correctly calculate the corresponding outlier score according to the abnormal degree of login behavior.

Keywords: Abnormal Login; Dimension Reduction; Gaussian Mixture Model; Local Outlier Factor; Login Behavior

1 Introduction

In recent years, emerging technologies such as cloud computing and big data have brought about great changes in the construction of information systems [13], which also brings great challenges to network security services. In the field of network security, abnormal login detection is the most basic network security scenario that all servers are faced with. It is an important link to ensure network security and purify the network space. Whether the service provider can make an accurate and rapid response to the increasing malicious login behavior becomes the evaluation standard for service quality.

In the past, the detection of abnormal logon-related attacks was usually analyzed by expert systems. With the expansion of the network scale, the number of users greatly increases. It has become an impossible task to clean and filter data from massive login logs and provide the judgment in a manually way. Meanwhile, the manual way cannot meet the timeliness and effectiveness of anomaly detection.

The abnormal login behavior detection involves two closely connected entities, the user and the server. The user login logs recorded on the server are mainly used as the input of the detection algorithm. Login logs are very important data sources in the field of abnormal behavior detection. Login logs record the login time, login IP address, and user account information and so on. Based on this information, user login behaviors can be analyzed over a period of time, and abnormal login behaviors can be located at some point.

However, the log types supported by different login servers are also different, and some servers can only provide simple log with fewer features. Therefore, how to make full use of these rare features extracted from login logs to detect abnormal login behaviors is a challenge to current researchers.

Aiming at these above problems, this paper proposes a common abnormal login detection method which can be applied to different types of login logs. In order to improve the universality of the method, only two features such as login time and login IP address, which are included in most login logs, are used as input features. In addition, this method adopts the local outlier factor algorithm [4], a classical algorithm in the field of anomaly detection, to judge the anomaly degree of login time. It also utilizes the IP address information after pre-processing by adopting the Gaussian mixture model to judge the anomaly degree

of IP address. Finally, the results of these two algorithms are integrated to calculate the decision abnormal score for each login behavior.

2 Related Works

At present, there are three types of anomaly detection methods in the industry: unsupervised anomaly detection method, semi-supervised anomaly detection method and supervised anomaly detection method [8]. Unsupervised anomaly detection method refers to comparing unlabeled data with other data to obtain abnormal points. Semi-supervised anomaly detection method refers to the establishment of normal behavior model based on normal label data, and detects anomaly based on the difference between the data and the normal model. Supervised anomaly detection refers to model training through labelled data which has been labeled as "abnormal" and "normal". Under the framework of these three broad approaches, there are a large number of multiple technologies.

In 2004, honeypot technology was used to pick up abnormal behavior [15]. In the area of anomaly behavior detection, an anomaly access analysis model based on the least square method of unitary linear regression was proposed, which takes the login success or failure as the input to create a regression model. Abnormal login behavior was analyzed based on time series [9] and historical offset [11], etc. In addition, machine learning and deep learning methods are often used in anomaly detection due to the large amount of data. For example, SVM-based intrusion detection system research [17] and Naive Bayes-based anomaly classification detection [18].

Liu *et al.* proposed to infer the possible attack on the host by analyzing the correlation between the performance of a series of network servers and the malicious attack they suffered [14]. They used random forests to classify and predict the extracted features. Marir *et al.* proposed a new cooperative structure, which combines the balanced decomposition mechanism and ensemble SVM to detect abnormal behavior [16]. Some papers have proposed using well-trained classifiers such as deep neural networks to classify detection data and find out the abnormal behavior [6] [7].

However, well-trained classifiers in supervised anomaly detection need a large number of labelled training data, but in practical applications, it is often unable to generate a large number of real calibration data, so the effective training of classifiers has become a difficult problem [5]. Several methods mentioned above depend heavily on the information given by a predetermined classification system [27]. For neural networks, classification systems use the information to judge anomalies, and then constantly adjust network parameters. For decision trees, classification systems use it to determine which attributes provide the most information. However, through the previous data analysis, we can clearly understand that the corresponding labelled training dataset cannot be obtained, so many supervised learning methods are not suitable for measurements in real network [25].

According to above problems, the unsupervised anomaly detection method is widely used in the current abnormal login detection area. Unsupervised anomaly detection method usually uses the similarity between point and adjacent point or the similarity between point and adjacent cluster to judge the anomaly degree of point. The anomaly degree can be reflected in various forms, such as density, distance [19], Angle [23], etc. These unsupervised anomaly detection methods show good performance in low dimensional cases. However, most unsupervised anomaly detection methods face the problem of dimensionality disaster. The performance of these algorithms decreases with the increase of the number of features [31]. When the number of features reaches an upper limit, it is necessary for these unsupervised methods to do feature preprocessing. By using feature screening or dimension reduction methods to process the original features, some key features that have the greatest impact on the results are found. Only these key features are input into the unsupervised detection model.

3 Abnormal Login Detection Method

In this section, we first describe the key technologies that will be used in our abnormal login detection method such as local outlier factor, Gaussian mixture model, robust scaler and PCA. And then the process of our method is described.

3.1 Local Outlier Factor

Local outlier factor (LOF) is a distance-based anomaly detection algorithm, which is mainly suitable for outlier detection in unlabeled datasets. The main idea is to obtain abnormal scores by comparing the density values of the test point P and its neighborhood. The density is mainly calculated by the distance between points. The closer the test point is to other points, the greater the density; on the contrary, the smaller the density is. If the density of the test point is lower, it means that the test point is more like an outlier, that is, the probability of abnormality is higher [3]. In general, LOF can describe the relative density of targets and can model local anomalies, and the time consumption of LOF algorithm primarily depends on the performance of the top K nearest neighbors search algorithm.

3.1.1 k-distance

The value of k is a natural number, and the k-distance of point P is defined as Formula (1):

$$d_k(p) = d(p, o) \tag{1}$$

The point O represents the k'th point far away from point P, $d_k(p)$ is the k-distance of point P. $d(p,o)$ represents the Euclidean distance between point P and point O. C{} represents a point set,$C\{x \neq p\}$ is the point set excluding P. k in Formula (1) meet the following criterions:

1) In the point set $C\{x \neq p\}$, there are at least k points $o^{'} \in C\{x \neq p\}$, it holds that $d(p,o^{'}) \leq d(p,o)$;

2) In the point set $C\{x \neq p\}$, there are at most k-1 points $o^{'} \in C\{x \neq p\}$ it holds that $d(p,o^{'}) \leq d(p,o)$.

3.1.2 Reachability Distance

The value of k is a natural number, and the reachability distance from point P to point O is defined as Formula (2):

$$reachdist(p,o) = max\{d_k(o), d(p,o)\} \tag{2}$$

On the left of Formula (2), $reachdist\ d(p,o)$ represents the reachability distance from point P to point O. The right of Formula (2) is the max value of the k-distance of point O and the distance between point P and point O.

3.1.3 Local Reachablity Density

The value of k is a natural number, and the local reachablity density of point P is defined as Formula (3):

$$lrd_k(p) = \frac{|N_k(p)|}{\sum_{o \in N_k(p)} reachdist(p,o)} \tag{3}$$

$N_k(p)$is the k-distance neighborhood of point P, represents all points within the k-distance range of point P, excluding point P itself. $|N_k(p)|$ represents the number of all points within the k-distance of point P. Formula (3) shows that the local reachablity density $lrd_k(p)$ of point P is defined as the inverse of the average reachability distance based on the number of k-distance neighborhood of P.

3.1.4 Local Outlier Factor

The value of k is a natural number, and the local outlier factor of point P is defined as Formula (4):

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|} \tag{4}$$

The local outlier factor of point P captures the degree to which P is called an outlier. It is the average of the ratios of the local reachablity density (lrd) of point P and those of P's k-distance neighborhood. If these are identical, which we expect for points in clusters of uniform density, the outlier factor is 1. If the lrd of P is only half of the lrds of P's k-distance neighborhood, the outlier factor of P is 2. Thus, the lower P's lrd is and the higher the lrds of P's k-distance neighborhood are, the higher is P's outlier factor.

3.2 Gaussian Mixture Model

Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities [20]. K Gaussian components are mixed, and K submodels are the hidden variables of the mixture model, that is, the training model is the weighted sum of K Gaussian submodels. Gaussian mixture model can be applied to any probability distribution and can estimate the probability density distribution of the test sample. Each Gaussian submodel represents a cluster. By projecting each data in the test sample on K Gaussian submodels, the probability of the sample belonging to each cluster can be obtained, and the maximum probability can be defined as its probability density [15, 28]. GMM can solve the two disadvantages of K-means: lack of flexibility in class shape and lack of cluster allocation probability. And GMM can model the overall distribution of the input data. After fitting, we can generate a new probability distribution function similar to the input data through the GMM model.

The probability distribution of gaussian mixture model is defined as Formula (5):

$$P(x|\theta) = \sum_{k=1}^{K} \alpha_k \psi(x|\theta_k) \tag{5}$$

X represents the measured data; K represents the number of Gaussian submodels in the Gaussian mixture model;α_k is the probability that the observed data belongs to the K'th submodel, where $\alpha_k \geq 0$, and satisfies $\sum_{k=1}^{K} \alpha_k = 1$, $\psi(x|\theta_k)$ is the Gaussian distribution density function of the K'th submodel, and $\theta_k = (\mu_k, \sigma_k^2).\theta = (\widetilde{\mu_k}, \widetilde{\sigma_k}, \widetilde{\alpha_k})$ are the parameters in this model.

In the single Gaussian model, in order to calculate the parameters of the model, it is necessary to use the maximum likelihood estimation method to estimate the parameters and find the extreme point by taking the derivative. In the Gaussian mixture model, maximum likelihood estimation cannot be used to obtain parameters by derivation, and the parameter estimation is generally carried out by iterative algorithm such as expectation-maximization (EM) algorithm [10, 21].

The basic idea of EM is to start with some initial guess of the parameter values $\theta^{(0)}$ and then iteratively search for better values for the parameters. Assuming that the current estimate of the parameters is $\theta^{(m)}$, our goal is to find another $\theta^{(m+1)}$ that can improve the likelihood $L(\theta)$ [29]. The input of EM is $x = (x^{(1)}, x^{(2)}, ...x^{(M)})$, and the procedure of the EM algorithm is as the following:

1) Initialize $\theta^{(0)}$ randomly or heuristically according to any prior knowledge about where the optimal parameter value might be.

2) Iteratively improve the estimate of θ by alternating between the following two-steps:

   a. The E-step (expectation): Given the estimate from the previous iteration $\theta^{(m)}$, compute the

conditional expectation $Q(\theta|\theta^{(m)})$ given in Formula (6):

$$Q(\theta|\theta^{(m)}) = \int_{X(y)} logp(x|\theta)p(x|y,\theta(m))\,\mathrm{d}x = E_{X|y,\theta^{(m)}}[log_p(X|\theta)] \quad (6)$$

b. The M-step (maximization): Re-estimate θ by maximizing the Q-function (7):

$$\theta^{(m+1)} = \arg\max_{\theta\in\Omega} Q(\theta|\theta^{(m)}) \quad (7)$$

3) Stop when the likelihood $L(\theta)$ converges.

Finally, model parameter θ is output. As we can see, The idea of stable rising of EM algorithm make EM algorithm find the "optimal convergence value" very reliably.

3.3 Robust Scaler

Robust Scaler is a robust normalization method suitable for data with outliers in statistics. It is found that most numerical features fall in compact regions with a small number of outliers. Therefore, the commonly used minmax scaler, which linearly maps the max value to 1 and the min value to 0, is not suitable. Instead, we use robust scaler technology. The robust scaler removed the median and scaled the data according to the interquartile range. Centering and scaling occurred independently on each feature by computing the relevant statistics on the samples in the dataset. This ensured that large changes in small singular values would not be washed out by the small changes in large singular values when aggregating. Because the extreme values are not used for calculation, the outlier features are reserved in the maximum to avoid the loss of outlier information.

3.4 Principal Component Analysis

Principal component analysis (PCA) is a simple, non-parametric method for extracting relevant information from confusing datasets. Through orthogonal transformation, PCA produces linear combinations of the original variables to generate the axes, also known as principal components. PCA is often used to make data easy to explore and visualize [2].

PCA can reduce the dimensionality of data by mapping the original high dimensionality data space to a low dimensionality space, and this mapping transformation loses little information. The main principle is to recombine multiple indicators to form comprehensive indicators. Comprehensive indicators are unrelated to each other, the number of comprehensive indicators is less than the original indicators, and the comprehensive indicators retain most of the information of the original indicators [1].

The mathematical processing of geometric changes is judged by variance. Generally, the larger the variance of a linear combination is, the more information this linear combination can preserve, so the linear combination with the largest variance is used as the first principal component. Usually, a single principal component cannot replace a large amount of information of the original index, so it is necessary to select the second principal component. When selecting the second principal component, it is needed to guarantee that the second principal component is not correlated with the first principal component, that is, to ensure that its covariance is 0. If the first principal component and the second principal component still do not meet the requirements, more principal components can be formed in turn.

3.5 Abnormal Login Detection Procedure

The abnormal login detection method based on local outlier factor and Gaussian mixture model mainly includes the following steps:

1) Perform data filtering and data cleaning on the original login logs, and extract the required features;

2) Use local outlier factor algorithm to process the feature of login time, and obtain the time anomaly score;

3) Use Gaussian mixture model to process the feature of login IP address, and obtain the IP anomaly score;

4) Normalize the scores based on robust scaler;

5) Combine the results of the two algorithms to obtain the final comprehensive score.

The detailed steps of these two algorithms are provided in pseudocode form (see Algorithms 1 and 2).

Algorithm 1 The anomaly detection algorithm of login time

Input: neighborhood size K, login log dataset D
Output: Abnormal login time data A

1: for time P in D do
   According to Formula (1), calculate K-neighborhood and K-distance of point P
2: For O in K neighborhood do
   According to Formula (2), calculate the reachability distance from each point O in K-neighborhood to P.
   According to Formula (3), calculate the local reachability density of P and O
3: Calculate the local outlier factor of P according to Formula (4).

Algorithm 2 The anomaly detection algorithm of login IP address
Input: login log dataset D
Output: Abnormal login IP data B

1: for IP in D do
Divide IP address into 4-dimensionality vectors according to IP domain.
Use PCA to reduce the dimensionalitys of segmentation results.
2: Use Bayesian information criterion to estimate and obtain the optimal parameters.
3: Build GMM model based on the optimal parameters and train.
4: Use the trained GMM model to analyze the IP vectors and obtain the anomaly score.

4 Experiments and Results Presentation

4.1 Experimental Environment

The experimental environment is divided into hardware and software. Hardware devices mainly include a PC and a network server. The software mainly includes Windows10 operating system environment, Python 3.8 environment, Mysql5.7 environment and so on. Table 1 lists the main hardware configurations and Table 2 lists the main software configurations.

Table 1: Hardware configuration

Hardware	Configuration
CPU	Intel® Core i5 2.4GHz
Hard Dist	128g
Memory	8g

Table 2: Software configuration

Software	Configuration
OS	Windows® 10 64 bit
Running Environment	Python 3.8
IDE	Pycharm
Database	Mysql 5.7

4.2 Experimental Dataset

In this experiment, a large number of login logs that record users' real operations is required. At present, there are some public traffic datasets on the Internet that record the operation traffic of real users. Datasets are widely used in the field of network security, such as KDD 99, LANL2017 [26] and CIC-IDS-2017 [22]. Due to sensitive information and privacy issues, some user information is blurred in THE CIC-IDS-2017 dataset, and users' identity information is not displayed in KDD 99 and LANL2017 datasets. Moreover, as these datasets are mostly flow datasets, users' login operation flows account for only a small part. It is difficult to distinguish them, so these datasets cannot meet the requirements of the algorithm in this paper.

In order to solve these problems existing in present datasets and obtain the dataset that meets the requirements of this experiment, this paper collected the log data of user login behavior from May to mid-June 2021 on the campus network of a university as the dataset of this experiment. This dataset contains 3512 users and more than 20,000 login behavior logs. And in the data set "S+ number" represents the student account, "T+ number" represents the faculty account, and "D+ number" represents the fixed assets. As shown in Table 3, each log in this data set contains the following information: user ID, online time, offline time, IPv4 address, IPv6 address, MAC address, VLAN, and traffic characteristics.

Generally, there are few abnormal login behaviors in the intranet, and the data collected is basically normal login data. In order to better calculate the evaluation indicators in the next step and effectively measure the accuracy and reliability of the detection method, this experiment added some abnormal login data to the dataset which accounts for about 10%.

Table 3: Dataset features

Features	Description
MASKID	User ID
Login Time	Login behaviors' Timestamp
Logout Time	Logout behaviors' Timestamp
IP	IPv4 Address
IPv6	IPv6 Address
MAC	MAC Address
VLAN	VLAN
Flow	Flow size

4.3 Experimental Results and Analysis

4.3.1 Abnormal Login Time Detection Result

The login log dataset from May to mid-June 2021 is preprocessed to extract the valid login time fields of each user. The login date is removed, the time-dependent data is retained, and the time-dependent data is converted to floating point hourly data.

In addition, due to different user behavior patterns and login habits, the data of all users cannot be uniformly analyzed as a whole, and the login time data of each user needs to be analyzed separately. Each user data is filtered and cleaned, and the obtained time data is used as the input of LOF algorithm to obtain its time outliers.

The login time data of a certain student user S233867 is analyzed through LOF algorithm, and the results are shown in Table 4. In the LOF algorithm, the number of neighborhood points is set to 10, and the "KDTree" algorithm is used to search for neighbor points. The results

of the LOF algorithm are the anomaly degree values, and the greater the absolute value of scores is, the more abnormal the time is. The closer the absolute value of score is to 1, the more consistent of local reachability densities of the neighborhood points and this point are, and the more normal the time is.

Table 4: LOF scores of user S233867

Time	IP	LOF Scores
8:18	192.165.3.233	-0.959184
10:27	192.165.7.27	-1.123488
8:30	192.163.3.13	-0.967749
10:14	192.166.3.82	-1.008852
8:29	192.168.3.1	-0.963066
9:32	192.168.4.249	-1.118052
10:34	192.160.8.103	-1.250154
8:49	192.160.6.117	-1.011906
10:49	192.163.3.76	-1.482631
10:22	192.164.6.164	-1.035697
8:30	192.161.8.41	-0.967749
.....	.....	.....

4.3.2 Abnormal Login IP Address Detection Result

Multiple login IP addresses used by each user during the period from May to mid-June 2021 are extracted from the login log dataset. Because different areas or buildings in the campus network own different subnet addresses, the user's login IP address can be used to distinguish the login location. In this paper, the extracted 32-bit IP data is divided into four fields. The first 8 bits were extracted as the first domain, 8-16 bits as the second domain, 16-24 bits as the third domain, and 24-32 bits as the fourth domain. The extracted four-dimensionality vectors were used as the input IP feature vectors for LOF. As shown in Table 5, the split login IP data domains of user S233867 are shown in Table 5.

Table 5: User S233867 login IP split result

IP	Oct1	Oct2	Oct3	Oct4
192.165.3.233	192	165	3	233
192.165.7.27	192	165	7	27
192.163.3.13	192	163	3	13
192.166.3.82	192	166	3	82
192.168.3.1	192	168	3	1
192.168.4.249	192	168	4	249
192.160.8.103	192	160	8	103
192.160.6.117	192	160	6	117
192.163.3.76	192	163	3	76
192.164.6.164	192	164	6	164
192.161.8.41	192	161	8	41
.....	.....	.....	.....	.....

In order to retain the vector information to the maximum extent, PCA is used to reduce the dimensionality of IP feature vectors. Four-dimensionality vectors are recombined into unrelated two-dimensionality comprehensive vectors. The two-dimensionality vector data of user S233867 after PCA reduction are shown in Table 6.

Table 6: PCA data of user S233867

IP	Pca1	Pca2
192.165.3.233	-0.83944	1.39606
192.165.7.27	-1.227506	-1.036094
192.163.3.13	-1.257309	-1.218422
192.166.3.82	-1.145483	-0.387031
192.168.3.1	-1.321514	-1.336496
192.168.4.249	-0.82759	1.601652
192.160.8.103	-1.032425	-0.158497
192.160.6.117	-1.013997	0.003788
192.163.3.76	-1.133004	-0.47248
192.164.6.164	-0.953685	0.579493
192.161.8.41	-1.162863	-0.887797
.....	.....	.....

Then, the login IP data after dimensionality reduction is taken as the input of GMM for training and testing. Parameters (number of mixture components) need to be set before the model trainin [24]. We use the Bayesian information criterion (BIC) to determine the best parameters of GMM. A lower BIC value indicates better performance. Table 7 presents the BIC values calculated during the GMM selection stage, with a 3 components model ultimately used.

Table 7: BIC values of GMM with differetnt components

Components	BIC Scores
1	588.591197
2	321.2657975
3	306.8626256
4	308.6550352
5	323.763971
6	324.2490362461543
7	332.8935619
8	350.1898125
9	366.2941619
....	.....

The selected GMM is trained using dimensionality reduction data. The results of GMM are shown in Table 8. In this table, the greater the absolute value of GMM scores is, the less similar this IP is to the adjacent IP, and the greater the anomaly degree of this IP is.

4.3.3 Comprehensive Outlier Score Results

According to LOF values of time anomaly scores and GMM values of IP anomaly scores of the same user S233867, the normalized scores are obtained by performing robust scaler processing. After weighting and averag-

Table 8: GMM scores of user S233867

IP	GMM Scores
192.165.3.233	-2.276909
192.165.7.27	-1.819922
192.163.3.13	-2.127074
192.166.3.82	-1.134199
192.168.3.1	-2.406284
192.168.4.249	-2.71535
192.160.8.103	-1.007782
192.160.6.117	-0.975995
192.163.3.76	-1.170537
192.164.6.164	-1.176711
192.161.8.41	-1.595632
.....	.....

ing, the comprehensive anomaly scores are obtained and sorted.

The anomaly login scores of user S233867 with top greater absolute values and the corresponding login information such as time and IP are displayed in Table 9.

Table 9: Comprehensive anomaly scores

Time	IP	Anormaly Scores
19:20	29.16.210.177	-5.750543
23:42	29.23.158.247	-5.513599
4:45	6.186.37.101	-4.4788329
23:47	192.161.6.255	-4.566675
23:13	192.168.7.12	-3.73085
5:24	24.62.236.139	-3.408691
22:03	192.164.2.154	-3.334973
21:25	112.4.112.137	-2.816587
.....	.....	.....

From Table 9, we can locate some abnormal login behaviors with the top 8 absolute anomaly scores. We analyzed these abnormal login behaviors in more detail. In the first row, the time point is relatively close to other time points in distance, the density is larger and the anomaly degree of the time is low; but the login IP address is quite different from other IP addresses, the anomaly degree of the IP address is very high; therefore, the comprehensive anomaly score combined from time and IP anomaly degree is also high. The high comprehensive anomaly scores in the second and third row are also based on the same reason. By analyzing the data in the fourth and fifth row, it can be seen that the login IP address has a high similarity with its neighboring IP addresses, and the IP anomaly degree is low. However, due to the large difference between the login time and the neighboring login times, the time anomaly degree is high, so it can be concluded that the comprehensive anomaly degree is high. The same method is used for manual analysis and processing of the subsequent data, and the reasons for the high degree of anomalies are known.

4.3.4 Comparative Experiment

In addition, in order to better demonstrate the effectiveness of the algorithm proposed in this paper, the abnormal login detection algorithm based on LOF & Gaussian mixture model proposed in this paper is compared with other two state-of-art unsupervised abnormal login detection algorithms, including: Anomaly login detection algorithm based on multi-dimensional probability [30] and anomaly detection algorithm based on isolated forest (iForest) [12]. Table 10 shows the experimental results of these three algorithms on the dataset of campus network login behavior logs collected in this experiment. The evaluation metrics include accuracy, recall, precision and F1-score.

Table 10: Performance comparison

Algorithm	Accuracy	Recall	Precision	F1 Score
LOF & GMM	0.9378	0.9639	0.9547	0.9681
Multi-dimensional probability	0.7376	0.7834	0.7904	0.7654
iForest	0.7728	0.7681	0.8994	0.8651

These 4 performance indexes present different measurement angles. Accuracy means correct proportion of classification, Recall means proportion of positive samples judged by the model in all positive samples, and Precision means proportion of real positive samples in positive samples judged by the model. In addition, F1 score gives consideration to precision and recall. As shown in Table 10, the abnormal login detection method based on LOF and GMM proposed in this paper performs better in four metrics than the other two algorithms. The detection method proposed by this paper has high performance indexes which are all around 95% which presents that this model can correctly and accurately classify the data and show good performance when the proportion of positive and negative samples is unbalanced. While the other two methods don't perform well. Although the iForest method has a relatively high precision, its accuracy and recall are low, which shows that iForest has a low sensitivity to abnormality and is easy to miss exceptions.

From the above series of experiment results, it can be proved that the proposed method in this paper can effectively detect the abnormal login behaviors. Each login behavior is given a comprehensive anomaly score, which can accurately reflect the outlier degree of user login behaviors. Through manual analysis, it can be found that the data items with obvious anomalies in the dataset all have high outliers. The values of four metrics in comparative experiments also demonstrate the advantages of this method in abnormal login behavior detection.

In addition, because our method uses only login IP address and login time features, there is no need for login authentication server to call additional modules to collect other data items during log collection, which solves the problem of how to analyze and extract rare features. The login IP address and login time are two general features, which can be extracted not only from login logs, but also from network traffic, SNMP-MIB database and

other data sources. So, our method has high portability and universality.

5 Conclusions

In this paper, a method of abnormal login detection based on local outlier factor and Gaussian mixture model is proposed. In this method, the login time and login IP address are used as the input features. It solves the problem of rare features faced by current abnormal login detection methods. Meanwhile, PCA method is introduced to reduce the risk of dimensionality disaster. Experimental results show that this method can detect abnormal login behaviors accurately and effectively.

In the future work, this method needs to be optimized and improved. It is planned to merge more features with strong generality for feature fusion, and integrate other features without destroying the high generality of the algorithm for more accurate and detailed judgment. In addition, after adding new features, in order to avoid even the appearance of dimensionality disaster, it is needed to analyze the importance of the various features. The PCA data dimensionality reduction for further analysis and optimization is needed to guarantee that the new features after dimensionality reduction operation can retain the original data information to the greatest extent, and the new structure between data is noninterference. Finally, after adding new features, we can try to fuse new algorithms. For example, we can detect abnormal behaviors by adding isolated forest algorithm. After adding new algorithms, it is necessary to design a new weighting function to judge the importance of each feature anomaly score through the scene, distribute the weight value according to the importance of different degrees, and finally obtain the comprehensive anomaly score fitting the scene.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62072074 No.62076054, No.62027827,No.61902054,No.62002047), the Frontier Science and Technology Innovation Projects of National Key R&D Program (No.2019QY1405), the Sichuan Science and Technology Innovation Platform and Talent Plan (No.2020TDT00020), the Sichuan Science and Technology Support Plan (No.2020YFSY0010).

References

[1] A. Boudou and S. Viguier-Pla, "Principal components analysis and cyclostationarity," *Journal of Multivariate Analysis*, vol. 189, p. 104875, 2022.

[2] R. G. Brereton, "Introduction to statistical, algorithmic and theoretical basis of principal components analysis," 2022.

[3] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LoF: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

[4] R. Chen, S. Gao, and E. Qiao, "Abnormal login judgment method and device," 2016.

[5] R. Chitrakar and C. Huang, "Selection of candidate support vectors in incremental svm for network intrusion detection," *computers & security*, vol. 45, pp. 231–241, 2014.

[6] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in *2016 8th IEEE international conference on communication software and networks (ICCSN)*, pp. 581–585, 2016.

[7] H. F. Eid, M. A. Salama, A. E. Hassanien, and T.-h. Kim, "Bi-layer behavioral-based feature selection approach for network intrusion classification," in *International Conference on Security Technology*, pp. 195–203, 2011.

[8] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial fraud:: A review of anomaly detection techniques and recent advances," 2022.

[9] Z. Q. Hu, "Design of abnormal behavior analysis model," Jan. 29, 2023. (`http://blog.nsfocus.net/abnormal-behavior-analysis-model-design`)

[10] K. Jia, Y. Xin, and T. Cheng, "Gaussian mixture modelling by exploiting competitive stop em algorithm," in *Journal of Physics: Conference Series*, vol. 2234, p. 012003, 2022.

[11] F. Li, P. Wang, and H. Chen. "Method and device for identifying abnormal login of account," 2020.

[12] F. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Eighth IEEE International Conference on Data Mining*, 2008.

[13] L. L. Liu and B. Lu, "Exploration and practice of network security level protection 2.0 system construction under the new situation," *Information security research*, pp. 196–199, 2022.

[14] Y. Liu, A. Sarabi, J. Zhang, P. Naghizadeh, M. Karir, M. Bailey, and M. Liu, "Cloudy with a chance of breach: Forecasting cyber security incidents," in *24th USENIX Security Symposium (USENIX Security 15)*, pp. 1009–1024, 2015.

[15] Z. Luo, K. He, and Z. Yu, "A robust unsupervised anomaly detection framework," *Applied Intelligence*, vol. 52, no. 6, pp. 6022–6036, 2022.

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

[17] M. Mohammadi, T. A. Rashid, S. H. T. Karim, A. H. M. Aldalwie, Q. T. Tho, M. Bidaki, A. M. Rahmani, and M. Hosseinzadeh, "A comprehensive survey and taxonomy of the svm-based intrusion detection systems," *Journal of Network and Computer Applications*, vol. 178, p. 102983, 2021.

[18] A. Rajesh and S. Kiran, "Anomaly detection using data mining techniques in social networking," *International Journal for Research in Applied Science and Engineering Technology*, vol. 6, pp. 1268–1272, 2018.

[19] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 427–438, 2000.

[20] D. A. Reynolds, "Gaussian mixture models.," *Encyclopedia of biometrics*, vol. 741, no. 659-663, 2009.

[21] P. Setoodeh, S. Habibi, and S. Haykin, "Expectation maximization," 2022.

[22] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, vol. 1, pp. 108–116, 2018.

[23] Z. Shou, H. Tian, S. Li, and F. Zou, "Outlier detection with enhanced angle-based outlier factor in high-dimensional data stream," *Int. J. Innov. Comput. Inf. Control*, vol. 14, no. 5, pp. 1633–1651, 2018.

[24] R. Suryanarayanan, "Anomaly detection on ip address data," Jan. 29, 2023. (`https://medium.com/geekculture/anomaly-detection-on-ip-address-data-1520955fa568`)

[25] J. Tao, W. Wang, N. Zheng, T. Han, Y. Chang, and X. Zhan, "An abnormal login detection method based on multi-source log fusion analysis," in *2019 IEEE International Conference on Big Knowledge (ICBK)*, pp. 229–235, 2019.

[26] M. J. Turcotte, A. D. Kent, and C. Hash, "Unified host and network data set," in *Data Science for Cyber-Security*, pp. 1–22, 2019.

[27] L. Wolf, A. Shashua, and D. Geman, "Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach," *Journal of Machine Learning Research*, vol. 6, no. 11, 2005.

[28] L. Xi, R. Wang, and Z. J. Haas, "Data-correlation-aware unsupervised deep-learning model for anomaly detection in cyber-physical systems," *IEEE Internet of Things Journal*, 2022.

[29] C. Zhai, "A note on the expectation-maximization (em) algorithm," *Course note of CS410*, 2007.

[30] W. Zhang, C. Zeng, Y. Cao, Z. Qin, and H. Chen, "An abnormal user login behavior detection method of industrial control system based on multi-dimensional probability analysis," in *Journal of Physics: Conference Series*, vol. 2246, p. 012082, 2022.

[31] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 5, pp. 363–387, 2012.

Biography

Guo Wei, born in 1986, is a network engineer with a bachelor's degree. His main research interests include network security maintenance and management.

He Yue, born in 1984, is a network engineer with a bachelor's degree. His main research interests include network security maintenance and management.

Chen Hexiong, born in 1984, is a network engineer with a master's degree. His main research interests include network technology and network security operation and maintenance.

Hang Feilu, born in 1984, is a network engineer with a master's degree. His main research interests include network security attack and defense technology.

Li Yunjie, born in 1999, is a graduate student. His main research interests include network security attack and defense technology.

Enhancing Transferability of Adversarial Examples by Successively Attacking Multiple Models

Xiaolin Zhang[1], Wenwen Zhang[1], Lixin Liu[2], Yongping Wang[1], Lu Gao[1], and Shuai Zhang[1]
(Corresponding author: Xiaolin Zhang)
School of Information Engineering, Inner Mongolia University of Science and Technology[1]
Baotou 014010, China
Email: zhangxl6161@163.com
School of Information,Renmin University of China, Beijing,100000, China[2]
(Received Aug. 1, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)

Abstract

Deep neural networks (DNNs) are very vulnerable to malicious attacks by adversarial examples, which is the process of deceiving the network by adding small perturbations to the original input. Moreover, the adversarial examples exhibit transferability that is more threatening to deep learning models: Adversarial examples generated by a specific network can mislead other black-box models. However, the adversarial examples tend to overfit the parameters of a particular network, which leads to their limited transferability. To boost the transferability of adversarial examples, we propose a successively attacking multiple high-accuracy models obtain adversarial examples toward the standard vulnerable directions of the models. Our approach differs from previous methods in that it adds modest adversarial perturbations to the image sequentially and progressively over multiple models. Our strategy can be well integrated into several state-of-the-art approaches to improve their transferability. Numerous experiments have demonstrated that our approach dramatically improves the ability of adversarial examples to transfer to unknown black-box models. The experiments also show that our attack strategy is superior to the traditional ensemble-based approach in terms of transferability improvement of adversarial examples.

Keywords: Adversarial Example; Deep Neural Networks; Multiple Models; Transferability

1 Introduction

In recent years, with the advancement of deep learning technology, deep neural networks (DNNs) have been more widely used in various fields, such as image classification [22] and object detection [3]. Meanwhile, adversarial attack techniques dedicated to misleading DNNs have emerged. In these attacks, attackers cause well-performing models to make incorrect predictions by adding subtle perturbations to the input. These maliciously modified inputs are known as adversarial examples [19]. More dangerous is that adversarial examples are transferable [15,20]; that is, adversarial examples generated by a specific network can mislead other black-box models with a high attack success rate. This property is very threatening in reality applications and poses a serious security problem for DNNs.

Since the concept of adversarial examples was first introduced by Szegedy *et al.* [19], many adversarial attack techniques have successfully deceived well-known architectures of neural networks, sometimes very astonishingly. Generally, adversarial attacks can be classified into two categories: white-box attacks and black-box ones [9]. In white-box attacks, attackers have full access to the target model, such as the model architectures and parameters. Black-box attacks can be further divided into two categories according to the mechanism attackers adopt [5]: query-based and transfer-based. In query-based black-box attacks, attackers have query access to the target model, which allows them to provide input images and obtain output predictions [6]. Transfer-based black-box attacks use existing known white-box models to design adversarial examples and directly use the resulting adversarial examples to fool the black-box target model [5].

Among all the categories of attacks, those transfer-based black-box attacks could be the most dangerous and mysterious, because the attacker does not need to know anything about the target model, including the input–output queries. As a result, research into such adversarial examples has significant societal and practical value. However, the adversarial examples generated by current attack methods are often highly coupled to the structure and parameters of the models, and their perturbations are difficult to perform efficiently to attack other models with different structures and parameters. There-

fore, there is still much room for research on transferable attacks of adversarial examples.

In this study, we propose a new attack strategy for DNN image classifiers that involves iteratively adding adversarial perturbations to images sequentially on multiple high-accuracy models to produce adversarial examples. Because high-accuracy models can accurately approximate the data's real decision boundaries, we believe that perturbations obtained from multiple high-accuracy models can be well directed to the direction in which the models are jointly vulnerable, allowing adversarial examples to enable better transfer to unknown black-box models.

We were inspired by the research conclusions of Liu *et al.* [12] and Seyed-Mohsen Moosavi-Dezfooli *et al.* [14]. Liu *et al.* [12] suggested that, if an adversarial image remains adversarial for multiple networks, then it is more likely to transfer to other networks as well. Their proposed ensemble-based approach is similar to our approach and also serves as a strategy to complement other advanced algorithms to further enhance the transferability of the adversarial examples. Unlike ours, the ensemble-based approach aggregates multiple models into one ensembled model; in other words, it extends the number of models horizontally. In contrast, our approach extends the number of models vertically as a way to generate adversarial perturbations that generalize well over the models.

Meanwhile, Seyed-Mohsen Moosavi-Dezfooli *et al.* [14] revealed the existence of universal perturbations and showed that such perturbations are not only universal across images but also generalize well across DNNs. It should be noted that our goal is to create adversarial examples with the best potential transferability, thus we should be willing to increase the adversarial perturbation suitably, sacrificing some visual perception in the process. We summarize the main contributions of this study as follows:

- We propose a new attack strategy to improve the transferability of adversarial examples. The method successively generates adversarial examples on multiple high-accuracy models to point in the common vulnerable directions of the models, reducing overfitting to specific models.

- Numerous experiments have shown that our approach significantly improves the transferability of adversarial examples. Experiments also demonstrate that our strategy outperforms the traditional ensemble-based approach in both white-box and black-box settings.

- Furthermore, our strategy as a complementary strategy, is well compatible with other transfer-based attacks and can be conveniently integrated into several state-of-the-art approaches to further improve their performance.

2 Related Works

2.1 Adversarial Attacks

In 2014, in the field of image recognition, the phenomenon of adversarial examples was first discovered by Szegedy *et al.* [19]. Since then, researchers in the field of deep learning have paid extensive attention to this phenomenon, and a series of attack methods have been proposed. According to the knowledge of attackers, adversarial attacks can be classified into two categories: white-box attacks and black-box ones [9].

In white-box attacks, attackers have full access to the target model, such as the model architectures and parameters. Szegedy *et al.* [19] proposed the box constraint algorithm L-BFGS. They view the generation of adversarial examples as a constrained optimization problem that maximizes the model prediction loss by optimizing the inputs. However, this method is costly. Immediately afterward, Goodfellow *et al.* [7] proposed the fast gradient sign method (FGSM), which is faster and more effective. This method finds the gradient direction of the loss function of the target model and adds adversarial perturbations to the input image in this direction to make the target model misclassify the adversarial examples. The FGSM is a one-step attack, which means that the adversarial examples are obtained by adding perturbations at once. Although this generation method is efficient, its attack success rate is low and it is easy to defend against. Therefore, based on the FGSM, Kurakin *et al.* [10] proposed an iterative FGSM (I-FGSM). The I-FGSM iteratively takes multiple small steps while adjusting the direction after each step. As a result, the I-FGSM has a higher success rate. Subsequently, Dong *et al.* [4] proposed a momentum-based I-FGSM (MI-FGSM) algorithm to boost adversarial attacks.

The method adds a momentum term to the I-FGSM, which stabilizes update directions and enables escape from poor local maxima during the iterations, resulting in more transferable adversarial examples. All the above methods generate adversarial examples based on the gradient of the target model. To further improve the attack success rate of the adversarial examples, as well as to reduce the gap between the adversarial examples and the original samples, researchers have proposed a series of optimization-based attack methods. This approach generates adversarial samples by optimizing the objective function being defined. For example, Papernot *et al.* [16] proposed a Jacobian-based saliency map attack (JSMA). The algorithm selects features with the greatest impact on the output target class, using forward derivative and adversarial saliency maps, and then obtains adversarial examples by modifying target features. Carlini and Wagner [1] transformed the process of finding adversarial examples into an optimization problem, where they put the requirements for high attack success and low perturbations of the adversarial examples into an objective function and obtained adversarial examples with better

performance.

In the white-box setting, the adversarial examples have a high attack success rate and small perturbations. However, in real-world application scenarios, the details of the models are usually not publicly available, so black-box attacks often pose a more realistic threat. The black-box attack can be divided into two categories: query-based and transfer-based.

In query-based black-box attacks, attackers have query access to the target model, which allows them to provide input images and obtain output predictions. Papernot *et al.* [17] trained a local model to substitute for the target DNN, using inputs synthetically generated by an adversary and labeled by the target DNN. Chen *et al.* [2] proposed zeroth-order-optimization-based attacks to directly estimate the gradients of the targeted DNN for generating adversarial examples. This method spares the need for training substitute models and avoids the loss in attack transferability. Su *et al.* [18] proposed a one-pixel attack, which uses differential evolution to find the optimal adversarial perturbations.

2.2 Transfer-based Black-box Attacks

Transfer-based black-box attacks use existing known models to design adversarial examples and directly use the resulting adversarial examples to fool the black-box target model. Because the attacker does not require any prior knowledge of the target model, transfer-based attacks have become one of the most threatening attacks on current practical applications to DNNs. Szegedy *et al.* [19] first investigated the phenomenon of transferability of adversarial examples, both across models and across datasets. Goodfellow *et al.* [7] continued their study on transferability and attributed the phenomenon of transferability of adversarial examples to the high matching of adversarial perturbations to the model parameters. Papernot *et al.* [15] went on to show through numerous experiments that examples largely transfer well across models trained with the same machine learning technique and across models trained with different techniques or ensembles making collective decisions. Liu *et al.* [12] conducted an extensive study of the transferability over large models and a large-scale dataset and proposed novel ensemble-based approaches to generating transferable adversarial examples.

They also showed that the decision boundaries of different models align with each other to explain the phenomenon of transferability of the adversarial examples. Dong *et al.* [4] introduced a momentum-based iterative algorithm. The method stabilizes update directions and escapes from poor local maxima during the iterations, resulting in more transferable adversarial examples. They attribute the transferability phenomenon to the fact that different machine learning models learn similar decision boundaries around a data point, allowing adversarial examples constructed for one model to be valid for other models as well. Wu *et al.* [20] systematically studied the factors affecting the transferability of adversarial examples. They found that model architecture similarity plays a crucial role. Moreover, they also found that models with lower capacity and higher test accuracy are endowed with stronger capability for transfer-based attacks.

3 Methodology

In this section, we elaborate on the proposed attack strategy. In Section 3.1, we give a description of the problem to transferability of the adversarial examples. In Section 3.2, we describe in detail our attack strategy. In Section 3.3, we list several generation algorithms targeted by our strategy.

3.1 Problem Description

Our goal is to generate adversarial examples with the best possible transferability. Specifically, we aim to generate a set of adversarial examples against the white-box model A that can deceive the black-box target model B with a high success rate. We do not need to obtain any information about model B, including the input and output queries. This question can be formulated as follows: Let X be the original sample and f be the black-box model B. We look for the adversarial example generated against model A that can satisfy

$$f(X) \neq f(X^*) \quad s.t. \|X^* - X\|_p \leq \xi, \tag{1}$$

where X^* is an adversarial example; $\|X^* - X\|_p$ is the p-norm of the perturbation, which measures the size of the added perturbation; and ξ is the upper limit of the perturbation.

3.2 Successively Attacking Multiple Models

Let $F = \{F_1, F_2, ..., F_n\}$ be a set of different types of high-accuracy DNNs, called network sets here. Our strategy is to acquire the adversarial perturbation υ iteratively over the network sets such that $\|\upsilon\|_p \leq \xi$, while fooling most networks in network sets. On each network, if the adversarial example X^*_{i-1} generated on the prior network F_{i-1} cannot mislead the current network F_i , the minimum perturbation υ_i that enables it to deceive the current network is computed, and this is added to X^*_{i-1} to obtain the current adversarial example $X^*_i = X^*_{i-1} + \upsilon_i$. More specifically, we first input the original sample X into the first network F_1 of the network sets and obtain the adversarial example $X^* = X + \upsilon_i$. Next, X^* is input into network F_2 , and, if it can successfully mislead F_2 , then it is directly input to the next network; otherwise, a small perturbation υ_2 is found on network F_2 so that it satisfies $F_2(X^*_1 + \upsilon_2) \neq F_2(X)$. A more general definition is as follows: On each network F_i of the network sets, we look for a small perturbation υ_i such that it satisfies

$$F_i(X^*_{i-1} + \upsilon_i) \neq F_i(X), \tag{2}$$

where F_i is the ith network in the network sets, with $i \in \{2, ..., n\}$, X^*_{i-1} is the adversarial example generated on the $(i-1)$th network, and υ_i is the minimum adversarial perturbation generated on F_i.

To ensure that the final obtained perturbation υ satisfies the constraint of $\|\upsilon\|_p \leq \xi$, the perturbations υ_i generated on each network are further projected on the l_p ball of radius ξ' and centered at 0. That is, let $P_{p,\xi'}$ be the projection operator defined as follows:

$$P_{p,\xi'}(\upsilon_i) = \arg\min_{\upsilon'_i} \|\upsilon_i - \upsilon'_i\|_2 \; subject\, to \, \|\upsilon'_i\|_p \leq \xi'. \tag{3}$$

Updating the adversarial example in each iteration gives

$$X^*_i = X^*_{i-1} + P_{p,\xi'}(\upsilon_i). \tag{4}$$

In general, we iteratively add small perturbations to images sequentially on the network sets F to generate the adversarial examples, reducing overfitting to the specific model. The algorithm is terminated when the perturbation υ exceeds a threshold ξ or all the networks in the network sets F are checked, and the samples X^* obtained at this time are the final adversarial examples. Finally, we feed X^* into a black-box target network $F^{'} \notin F = \{F_1, F_2, ..., F_n\}$ to test its transferability. The detailed algorithm is provided in Algorithm 1. The objective of Algorithm 1 is not to find a minimum perturbation that fools the network sets but rather to find a relatively small enough perturbation to maximize the transferability of the adversarial examples.

In our experiments, we selected a series of high-accuracy networks to form the network sets $F = \{F_1, F_2, ..., F_n\}$ and investigated the effect of the number of networks in the network sets on the transferability and the distortion rate of the adversarial examples. Interestingly, in practice, the number of networks need not be large to obtain adversarial examples with good transferability.

3.3 Generation Algorithms

In our approach using a successively attacking multiple high-accuracy model (SAMM), although we do not require that the adversarial examples generated on the first network of the network sets have good transferability, a generation algorithm with some transferability will make our method work better. Therefore, on the networks of network sets, we consider two types of algorithms for generating adversarial examples—a gradient-based approach and an optimization-based approach—to showcase the effectiveness of our strategy in alleviating the overfitting issue and improving the transferability of adversarial examples.

Algorithm 1 Crafting transferable adversarial examples

Input:
Benign image X;
Classifiers $F = \{F_1, F_2, ..., F_n\}$;
Global maximum disturbance ξ;
Local maximum disturbance ξ';
Output:
Adversarial example X^*

Begin
$X^*_0 \leftarrow X,\ i \leftarrow 1, n \leftarrow N$
while $i \leq N$ and $\|X^* - X\|_p \leq \xi$ **do**
  if $F_i(X^*_{i-1}) = F_i(X)$ **then**
    υ_i=generation_algorithm(F_i, X^*_{i-1})
    $X^*_i = X^*_{i-1} + P_{p,\xi'}(\upsilon_i)$
  else
    $X^*_i = X^*_{i-1}$
  end if
  $X^* = X^*_i$
  $i = i + 1$
end while
return X^*

3.3.1 Gradient-based Approach

In the gradient-based approach, we used the I-FGSM [10] and the MI-FGSM [4]. The I-FGSM can be expressed as

$$\begin{aligned} &X^{adv}_0 = X^*_i, \\ &X^{adv}_{n+1} = Clip^{\epsilon}_X\{X^{adv}_n + \alpha \cdot sign(\nabla_X L(X^{adv}_n, y^{true}; \theta))\}, \end{aligned} \tag{5}$$

where $Clip^{\epsilon}_X$ indicates the resulting image is clipped within the ϵ-ball of the original image X , n is the iteration number, and α is the step size. In the MI-FGSM, Equation (5) is replaced with

$$\begin{aligned} g_{n+1} &= \mu \cdot g_n + \frac{\nabla_X L(X^{adv}_n, y^{true}; \theta)}{\|\nabla_X L(X^{adv}_n, y^{true}; \theta)\|_1}, \\ X^{adv}_{n+1} &= Clip^{\epsilon}_X\{X^{adv}_n + \alpha \cdot sign(g_{n+1})\}, \end{aligned} \tag{6}$$

where μ is the decay factor of the momentum term and g_n is the accumulated gradient at iteration n.

3.3.2 Optimization-based Approach

In the optimization-based approach, we used the advanced JSMA [16] and C&W attacks [1]. The JSMA first computes the forward derivative to show the degree of influence of each input feature on the target class. The next step is to build a saliency map and find the most salient component i that will then be changed:

$$S(X,t)[i] \quad = \quad \begin{cases} 0, \text{ if } \frac{\partial F_t(X)}{\partial X_i} < 0 \text{ or } \sum\limits_{j \neq t} \frac{\partial F_j(X)}{\partial X_i} > 0 \\ (\frac{\partial F_t(X)}{\partial X_i}) \cdot |\sum\limits_{j \neq t} \frac{\partial F_j(X)}{\partial X_i}|, \text{ otherwise,} \end{cases} \tag{7}$$

where $\frac{\partial F_t(X)}{\partial X_i}$ and $\sum\limits_{j \neq t} \frac{\partial F_j(X)}{\partial X_i}$ in these maps quantify how much $F_t(X)$ will increase and $\sum\limits_{j \neq t} F_j(X)$ will decrease, given a modification of the input feature X_i .

Finally, the algorithm selects the component: $i_{max} = \arg\max_i S[x,t][i]$, in which usually a default value θ is added.

The C&W attack can be expressed as

$$\begin{aligned} &minimize \|\delta\|_p + c \cdot f(X+\delta) \\ &such\, that\, X+\delta \in [0,1]^n, \end{aligned} \tag{8}$$

where $\|\delta\|_p$ is the L_p distance of the perturbation δ added to original sample X and c is a constant greater than zero. The recommended choice of f is $f(x) = (max_{i \neq t} Z(x)_i - Z(x)_t)^+$.

4 Experiment

This section is dedicated to a variety of experiments conducted on the proposed attacks using the SAMM and several comparisons with the ensemble-based approach. In Section 4.1, we describe the specific setup of the experiments. In Section 4.2, we train a series of models with high accuracy. In Section 4.3, we measure the white-box performance and transferability of our SAMM approach and analyze the results in detail. In Section 4.4, we conduct detailed comparison experiments between our approach and the ensemble-based approach. In Section 4.5, we also study the effect of the number of models in the network sets of our method on the transferability and distortion rate.

4.1 Experimental Setup

4.1.1 Datasets

In our experiments, we used the MNIST [11], FMNIST [21], and CIFAR-10 [8] datasets. The MNIST dataset contains 70,000 28 × 28 grayscale images in 10 classes, divided into 60,000 training images and 10,000 test images. The possible classes are digits from 0 to 9. The FMNIST dataset also contains 70,000 28 × 28 grayscale images in 10 classes, divided into 60,000 training images and 10,000 test images. These images are divided into 10 different classes (T-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot). The CIFAR-10 dataset contains 60,000 32 × 32 × 32 RGB images. There are 50,000 training images and 10,000 test images. These images are divided into 10 different classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck), with 6,000 images per class.

4.1.2 Models

For each of the three datasets, we trained 12 DNNs: VGG16, VGG19, ResNet50, ResNet101, ResNet152, Inception_v3, Inception_v4, Inception_ResNet_v2, DenseNet121, DenseNet161, DenseNet169, and DenseNet201. We selected some of these high-accuracy models to build the network sets. We also trained the corresponding ensembled models for comparison experiments.

4.1.3 Metrics

Given a set of adversarial pairs, $\{(X_1^{adv}, y_1^{true}), (X_2^{adv}, y_2^{true}), \cdots, (X_m^{adv}, y_m^{true})\}$, we calculate their transferability (%) of fooling a given black-box model $f(x)$ by

$$100 \times \frac{1}{m} \sum_{i=1}^{m} 1_{f(x_i^{adv}) \neq y_i^{true}}, \tag{9}$$

where y_i^{true} is the original label and $f(x_i^{adv})$ is the adversarial label. If $f(x)$ is the model used to generate adversarial examples, then Equation (9) indicates the white-box attack performance.

In this study, we measure the distortion rate of the adversarial example by using the L_2 norm

$$\|X^* - X\|_2 = \Big(\sum_{i-1}^{n} |X_i^* - X_i|^2 \Big)^{-2}, \tag{10}$$

where X is the original sample, X^* is the adversarial example, and n is the dimension of X and X^* . $X^* \in (0,1)$ denotes the ith dimensional pixel value of X.

4.1.4 Implementation Details

We set the number of models to $n = 6$ in the network sets, the global maximum perturbation to $\xi = 10$, and the local maximum perturbation to $\xi' = 5$ on each network. For the generation algorithm, in the I-FGSM we set the step size to α =1, the total iteration number to $N = min(\epsilon+4, 1.25\epsilon)$, and the maximum perturbation of each pixel to ϵ =15 , which is still imperceptible to human observers [13]. For the momentum term, the decay factor μ was set to be 1 as in Ref. [4]. The C&W attack uses the L_2 norm to measure the distortion rate and set the hyper parameter $c = 1$. The JSMA set feature values to increase $\theta = 1$.

4.2 Training Networks

In our approach, we need multiple high-accuracy networks to build the network sets $F = \{F_1, F_2, ..., F_n\}$. Therefore, for the MNIST, FMNIST, and CIFAR-10 datasets, we trained 12 DNNs and several ensemble of networks, respectively. The results are listed in Table 1, where Ensemble is composed of ResNet152, Inception_v3, Inception_v4, and Inception_ResNet_v2.

On the MNIST and FMNIST datasets, the training epochs were 100 with a batch size of 64. The initial learning rate was set to 0.1 and reduced by a factor of 10 every 20 epochs, and the stochastic gradient descent method was used for optimization. Similarly, on the CIFAR-10 dataset, the training epochs were 200 with a batch size of 128. The initial learning rate was set to 0.01 and reduced by a factor of 10 every 50 epochs, and the stochastic gradient descent method was used for optimization.

Table 1: Classification accuracy of models.

Model/Dataset	MNIST	FMNIST	CIFAR-10
VGG16	99.24%	93.57%	93.49%
VGG19	99.25%	93.63%	93.49%
ResNet50	99.35%	94.91%	93.52%
ResNet101	99.58%	95.51%	93.95%
ResNet152	99.62%	95.53%	93.96%
Inc_v3	99.63%	95.56%	94.58%
Inc_v4	99.65%	95.59%	93.71%
Inc_Res_v2	99.65%	95.55%	93.68%
DenseNet121	99.61%	96.21%	94.86%
DenseNet161	99.68%	96.24%	94.97%
DenseNet169	99.71%	96.45%	95.56%
DenseNet201	99.73%	96.51%	95.94%
Ensemble	**99.75**%	**96.63**%	**96.39**%

4.3 Results and Analysis of SAMM

In this section, we integrate our strategy into various advanced generation algorithms, including the I-FGSM, MI-FGSM, JSMA, and C&W attack, to show the effectiveness of our approach in alleviating the overfitting issue and improving the transferability of adversarial examples.

We first tested the white-box attack success rate and transferability of these four algorithms, that is, their performance on a single network. The results are listed in Table **??**, where the first column gives the success rate of white-box attacks and the remaining columns give the success rates of transfer-based attacks. It can be observed from the table that these generation algorithms maintain attack success rates of ¿99% in the white-box setting, but the transferability of their adversarial examples is relatively low, especially for C&W attacks. For example, in the MNIST dataset, the adversarial examples generated on the Inception_v3 attack on the Inception_v4 dataset exhibited transfer success rates of 15.1%, 39.7%, 10.5%, and 6.7%, respectively. This is because these generation algorithms overfit the parameters of the target network, and while their adversarial examples have a fairly high success rate of white-box attacks, these samples are difficult to transfer to unknown black-box models.

Next, we combined our SAMM approach with four generation algorithms to produce adversarial examples. In this experiment, we selected six high-accuracy models to construct the network sets for generating the adversarial examples and a black-box model to test the transferability of the adversarial examples. Specifically, we selected a total of seven networks—Inception_v3 (Inc_v3), Inception_v4 (Inc_v4), Inception_ResNet_v2 (Inc_Res_v2), ResNet152 (Res152), DenseNet161 (Den161), DenseNet169 (Den169), and DenseNet201 (Den201)—six of which were constructed as network sets and one as a black-box test network, and we conducted multiple experiments. The networks in the network sets were then sorted from lowest to highest classification accuracy.

The results are listed in Table **??**, where the first column gives the success rate of white-box attacks and the remaining columns give the success rates of transfer-based attacks. The white-box attack is the adversarial example attack of Inception_v3 generated on the network sets containing Inception_v3. In the white-box setting, it can be seen that the combination of our SAMM approach and these generation algorithms all achieve essentially a 100% attack success rate. We mark the white-box attack success rate with $*$. This is because our approach superimposes multiple layers of perturbations on the image. More importantly, in the black-box setting, we observe the transferability of the adversarial examples, where the sign '-' indicates that this network is a black-box test model and the rest of the networks form the network sets to generate the adversarial examples. We combine Tables **??** and **??** to observe the enhancement of our SAMM for the transferability of adversarial examples. It can be seen that the SAMM significantly improves the transferability of the adversarial examples for each generation algorithm. For example, Inception_v4 as a black-box test model for the MNIST dataset, I-FGSM, MI-FGSM, JSMA, and C&W achieved 68.1%, 84.7%, 40.7%, and 32.4% attack success rates, respectively, by successively adding adversarial perturbations onto network sets, while their attack success rates by obtaining perturbations on a single model were only 15.1%, 39.7%, 10.5%, and 6.7%, respectively. It is also observed from Table **??** that the adversarial examples generated by the combination of our SAMM approach and momentum were transferred to each test network with a high success rate. This is because the MI-FGSM can mitigate overfitting to some extent, which verifies that our method is well compatible with other advanced transfer-based approaches to further improve their performance.

4.4 Comparing SAMM and Ensemble-based Approaches

In this section, we compare our approach with the ensemble-based approach in terms of transferability and distortion rate. We refer to Ref. [17] on the setup of ensembled models and select five models: Inception_v3 (Inc_v3), Inception_v4 (Inc_v4), Inception_ResNet_v2 (Inc_Res_v2), ResNet152 (Res152), and DenseNet201 (Den201). Adversarial examples are generated on an ensemble of four networks and tested on the ensembled network and one black-box network, using I-FGSM and MI-FGSM, respectively. Although the network sets of our approach work best with six models (Section 4.5), here we use four models to generate adversarial examples to unify the variables with the ensemble-based approach.

The results are listed in Table **??**, where the sign '-' indicates that the model was removed and used as a black-box test network, and the other four models were used to generate adversarial examples. It can be ob-

Table 2: White-box performance and transferability of generation algorithms.

Dataset	Model	Attack	Inc_v3	Inc_v4	Inc_Res_v2	Res152	Den201	Den161	Den169
MNIST	Inc_v3	I-FGSM	99.2%	15.1%	13.7%	11.3%	9.5%	10.5%	9.8%
		MI-FGSM	99.9%	**39.7%**	**36.3%**	**29.1%**	**23.2%**	**25.7%**	**24.6%**
		JSMA	99.0%	10.5%	9.6%	9.3%	7.2%	8.9%	8.2%
		C&W	**100%**	6.7%	6.3%	6.1%	2.1%	3.3%	2.8%
FMNIST	Inc_v3	I-FGSM	99.0%	13.7%	12.2%	9.8%	7.2%	8.9%	8.2%
		MI-FGSM	99.9%	**36.9%**	**34.5%**	**28.1%**	**21.3%**	**24.5%**	**22.3%**
		JSMA	99.2%	9.3%	9.1%	8.9%	6.9%	7.5%	7.3%
		C&W	**100%**	6.1%	5.8%	5.4%	1.9%	3.1%	2.2%
CIFAR-10	Inc_v3	I-FGSM	99.0%	13.2%	12.3%	8.6%	6.5%	7.4%	6.9%
		MI-FGSM	99.8%	**36.3%**	**33.2%**	**26.2%**	**19.1%**	**23.3%**	**21.6%**
		JSMA	99.4%	8.4%	8.2%	8.0%	6.5%	7.5%	7.1%
		C&W	**99.9%**	5.6%	5.2%	5.1%	1.2%	2.5%	1.8%

Table 3: White-box performance and transferability of SAMM + generation algorithms.

Dataset	Attack	Inc_v3	-Inc_v3	-Inc_v4	-Inc_Res_v2	-Res152	-Den201	-Den161	-Den169
MNIST	SAMM+I-FGSM	**100%***	70.5%	68.1%	66.6%	64.2%	52.5%	56.5%	54.4%
	SAMM+MI-FGSM	**100%***	**86.9%**	**84.7%**	**83.9%**	**80.1%**	**70.8%**	**75.3%**	**73.6%**
	SAMM+JSMA	**100%***	43.2%	40.7%	36.3%	35.1%	25.6%	29.7%	26.3%
	SAMM+C&W	**100%***	33.9%	32.4%	32.3%	30.2%	19.8%	25.2%	23.6%
FMNIST	SAMM+I-FGSM	**100%***	67.0%	66.7%	66.2%	62.8%	57.2%	59.2%	58.6%
	SAMM+MI-FGSM	**100%***	**82.7%**	**80.3%**	**80.5%**	**77.1%**	**70.9%**	**73.4%**	**72.6%**
	SAMM+JSMA	**100%***	40.7%	38.3%	36.2%	34.1%	24.6%	28.6%	25.5%
	SAMM+C&W	**100%***	30.9%	30.3%	30.2%	27.1%	20.9%	23.4%	22.6%
CIFAR-10	SAMM+I-FGSM	99.9%*	62.7%	59.1%	58.5%	53.7%	47.5%	51.3%	49.5%
	SAMM+MI-FGSM	**100%***	**79.7%**	**78.3%**	**78.1%**	**76.2%**	**70.3%**	**73.7%**	**71.9%**
	SAMM+JSMA	99.9%*	38.7%	37.3%	36.2%	33.2%	23.5%	27.7%	25.4%
	SAMM+C&W	**100%***	28.6%	27.4%	27.2%	26.2%	16.3%	20.7%	18.5%

served that our SAMM approach has better transferability than the ensemble-based approach. For example, in the MNIST dataset, in the adversarial example attack on the black-box model Inception_v3, our method achieves 61.5% and 79.1% attack success rates using the I-FGSM and MI-FGSM, respectively, whereas the ensemble-based approach achieves 47.2% and 71.4% attack success rates using the I-FGSM and MI-FGSM, respectively. This convincingly illustrates that the adversarial examples generated by our SAMM approach are better able to migrate to unknown networks than those of the ensemble-based approach.

We also tested the success rate of white-box attacks for our approach and the integrated approach, and the results are listed in Table **??**. The white-box attack success rate of the ensemble-based approach is the adversarial examples generated on the ensembled model attacking the ensembled model, while our SAMM method is the adversarial examples generated on the network sets attacking each model in the network sets. In the white-box setting, we can see in combination with Tables **??** and **??** that the success rates of the I-FGSM and MI-FGSM on the ensembled model are slightly lower than those of the I-FGSM and MI-FGSM on the single model. This is because attacking an ensemble of multiple networks is much more difficult than attacking a single model. However, the success rates of white-box attacks of our approach are both higher than those of the I-FGSM and MI-FGSM on a single model. This is because our approach superimposes multiple layers of perturbations on the image. Taken together, this convincingly demonstrates that our SAMM approach outperforms the ensemble-based approach in terms of both white-box attack success rate and transferability.

In addition, we evaluated the distortion rate of the adversarial examples generated by the two approaches using the L_2 norm. We selected 5,000 corresponding adversarial examples on each of the three datasets to measure the average distortion rate, and the results are listed in Table 6. It can be observed that the adversarial examples generated by our approach have a relatively large L_2 norm compared to those from the ensemble-based approach. Therefore, we also visualize the adversarial examples of our approach in Figure 1 to observe the effect of adversarial perturbations on visual perception. Figure 1 shows the adversarial examples and their corresponding clean images for each class generated by SAMM+I-FGSM on the three datasets. These visualization results show that, although our method produces relatively large adversarial perturbations, we can still visually discriminate the images clearly.

Table 4: Transferability of SAMM and the ensemble-based approach.

Dataset	Attack	-Inc_v3	-Inc_v4	-Inc_Res_v2	-Res152	-Den201
MNIST	Ensemble+I-FGSM	47.2%	45.1%	43.7%	37.3%	35.4%
	SAMM+I-FGSM	61.5%	68.1%	66.6%	64.2%	62.5%
	Ensemble+MI-FGSM	71.4%	66.7%	64.3%	58.1%	53.6%
	SAMM+MI-FGSM	**79.1%**	**78.7%**	**78.9%**	**75.1%**	**70.8%**
FMNIST	Ensemble+I-FGSM	44.0%	42.6%	40.1%	37.5%	33.6%
	SAMM+I-FGSM	57.0%	56.7%	56.2%	52.8%	47.2%
	Ensemble+MI-FGSM	68.7%	66.3%	64.5%	64.1%	59.6%
	SAMM+MI-FGSM	**72.7%**	**70.3%**	**70.5%**	**67.1%**	**64.9%**
CIFAR-10	Ensemble+I-FGSM	41.5%	39.7%	35.2%	32.7%	27.5%
	SAMM+I-FGSM	52.7%	49.1%	48.5%	43.7%	37.5%
	Ensemble+MI-FGSM	64.7%	60.3%	56.2%	53.2%	50.5%
	SAMM+MI-FGSM	**69.7%**	**67.3%**	**67.2%**	**65.2%**	**63.3%**

Table 5: White-box performance of SAMM and the ensemble-based approach.

Dataset	Attack	-Inc_v3	-Inc_v4	-Inc_Res_v2	-Res152	-Den201
MNIST	Ensemble+I-FGSM	98.5%	98.6%	99.2%	98.1%	99.0%
	SAMM+I-FGSM	**100%**	**100%**	**100%**	**100%**	**100%**
	Ensemble+MI-FGSM	98.9%	98.8%	99.3%	98.6%	99.2%
	SAMM+MI-FGSM	**100%**	**100%**	**100%**	**100%**	**100%**
FMNIST	Ensemble+I-FGSM	97.1%	97.5%	98.9%	96.8%	98.1%
	SAMM+I-FGSM	99.8%	99.8%	99.9%	99.9%	**100%**
	Ensemble+MI-FGSM	97.6%	97.4%	99.0%	97.2%	97.9%
	SAMM+MI-FGSM	**99.9%**	**99.9%**	**100%**	**100%**	**100%**
CIFAR-10	Ensemble+I-FGSM	96.6%	96.9%	98.7%	96.2%	97.0%
	SAMM+I-FGSM	99.5%	99.5%	99.6%	99.6%	99.7%
	Ensemble+MI-FGSM	96.9%	96.9%	98.8%	96.8%	96.8%
	SAMM+MI-FGSM	**99.7%**	**99.7%**	**99.8%**	**99.8%**	**99.9%**

Table 6: Distortion rate (L_2 norm) of SAMM and the ensemble-based approach.

Attack	MNIST	FMNIST	CIFAR-10
Ensemble+I-FGSM	3.01	3.36	3.91
SAMM+I-FGSM	4.21	4.34	4.62
Ensemble+MI-FGSM	**2.84**	**3.04**	**3.42**
SAMM+MI-FGSM	3.42	3.94	4.31

Figure 1: Adversarial examples and clean samples of SAMM+I-FGSM.

4.5 Effect of the Number of Models on Transferability and Distortion Rate

Finally, we investigated the effect of the number of models in the network sets on the transferability and distortion rate of the adversarial examples. We added 11 networks sequentially to the network sets (VGG16, VGG19, ResNet50, ResNet101, ResNet152, Inception_v4, Inception_ResNet_v2, DenseNet121, DenseNet161, DenseNet169, and DenseNet201) to observed changes in the transferability and distortion rate of the adversarial examples. We generated 5,000 adversarial examples on the MNIST dataset and calculated their mean value for transfer-based attack success rate and distortion rate, and the results are shown in Figure 2.

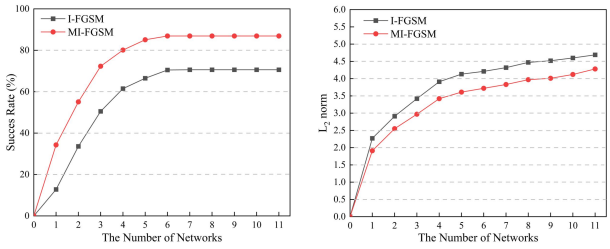


Figure 2: Effect of the number of models on transferability and distortion rate.

From Figure 2, it can be observed that, as the number of models increases, the transferability and distortion rate of the adversarial examples generated by SAMM+I-FGSM and SAMM+MI-FGSM increase rapidly and then gradually level off. This is because, as iterations add small adversarial perturbations to the image, these perturbations gradually move toward the common vulnerable di-

rections of the models. Therefore, based on the trade-off between transferability and distortion rate of adversarial examples, we chose to use six networks in our experiments to construct the network sets.

5 Conclusion

In this work, we propose a new approach to improve the transferability of adversarial examples. Specifically, we obtain the adversarial examples by successively attacking multiple high-accuracy models and finding the common vulnerable directions of the models to improve the transferability of the adversarial examples. Extensive experiments have shown that our proposed SAMM attack method significantly improves the transferability of adversarial examples of several advanced generation algorithms. In particular, the adversarial examples generated by the integration of our method and the momentum method are transferred to the various black-box models with a very high success rate. This confirms that our approach is well compatible with advanced algorithms to further improve their performance. We have also conducted detailed comparison experiments with the traditional ensemble-based approach, and the experimental results demonstrate that the adversarial examples generated by our approach have a better white-box attack success rate and transferability.

We should mention that, although our method significantly improves the transferability of the adversarial examples, it also sacrifices some visual perception accordingly. Therefore, we will continue our research in greater depth next, expecting to obtain adversarial examples with better transferability and smaller distortion. Finally, a depth study of the transferability of the adversarial samples can facilitate understanding of the robustness of the model, which in turn can lead to the design of better defense strategies to resist malicious attacks in real applications. Therefore, the study of adversarial example portability has important social significance and application value.

Acknowledgments

This work was supported by the Natural Science Foundation of China under Grant 61562065 and the Inner Mongolia Natural Science Foundation Project under Grant 2019MS06001.

References

[1] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy (SP'17)*, pp. 39–57, 2017.

[2] P. Chen, H. Zhang, Y. Sharma, J. Yi, and C. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26, 2017.

[3] Y. Ding, Z. Wang, B. Li, G. Xu, and L. Deng, "Automatic small target detection in complex background: a state-of-the-art survey," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 11884, p. 118841S, 2021.

[4] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193, 2018.

[5] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4307–16, 2019.

[6] A. Eyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *35th International Conference on Machine Learning*, vol. 5, pp. 3392–3401, 2018.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, vol. 3, 2015.

[8] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *University of Toronto, Tech. Rep*, vol. 1, 2009.

[9] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, and C. Xie, "Adversarial attacks and defences competition," in *Competition: Building Intelligent Systems*, 2018.

[10] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *5th International Conference on Learning Representations - Workshop Track Proceedings*, 2019.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–324, USA, 1998.

[12] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *The 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[13] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao, "Foveation based mechanisms alleviate adversarial examples," in *Computer Science*, 2015.

[14] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2017.

[15] N. Papernot, P. McDaniel, and I Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," in *arXiv*, 2016.

[16] N. Papernot, P. McDaniel, S Jha, M. Fredrikson, Z. B Celik, and A. Swami, "The limitations of deep

learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pp. 372–387, Saarbruecken, Germany, 2016.
[17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security*, pp. 506–519, 2017.
[18] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," in *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 828–841, 2019.
[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *The 2nd International Conference on Learning Representations (ICLR 2014)*, 2014.
[20] L. Wu, Z. Zhu, C. Tai, and E. Weinan, "Understanding and enhancing the transferability of adversarial examples," in *arXiv*, 2018.
[21] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," vol. abs/1708.07747, 2017.
[22] Z. Q. Zhao, S. T. Xu, D. Liu, W. D. Tian, and Z. D. Jiang, "A review of image set classification," in *NEUROCOMPUTING*, vol. 335, pp. 251–260, 2019.

Biography

Xiaolin Zhang was born in Baotou, China, in December 1966. She received the bachelor's degree in computer science and technology from Northeastern University, in 1988, the master's degree in automation from the Beijing University of Science and Technology, in 1995, and the Ph.D. degree in computer science and technology from Northeastern University, in 2006. Since 1988, she has been with the Inner Mongolia University of Science and Technology, where she is currently the Deputy Director of the Head of the Computer Science Department, Professor Committee of the Information Technology College, and the Director of the Department of the Computer Science. She has trained more than 70 master's degree students and now is training 12 master's degree students. She has published over 80 academic articles, including more than 30 articles in EI and 7 articles in SCI. She is responsible for many projects, such as the National Natural Science Foundation of China, the National Social Science Fund Project, the Chunhui Project of the Ministry of Education, the Natural Science Foundation of Inner Mongolia Project, and the Inner Mongolia Education Department Fund Project. Her current research interests include image processing, natural language processing, adversarial attacks of image and text, machine learning security, big data processing technology, social network privacy protection technology. Dr. Zhang is a member of the Chinese Computer Society, the Information System Professional Committee, the China Computer Society, and the Director of the Inner Mongolia Autonomous Region Computer Society.

Wenwen Zhang was born in Heze, China, in November 1996. She received the B.S. degree in computer science and technology from the Jining Medical University, in 2019. She is currently pursuing the master's degree in computer science and technology with the Inner Mongolia University of Science and Technology. Her research interests include adversarial attacks based on image classification.

Lixin Liu was born in Baotou, China, in 1984. She received the bachelor's degree in information security from Central South University, in 2007, and the master's degree in computer science and technology from Central South University, in 2010. She received the Ph.D. degree at the Renmin University of China in 2021. Since 2010, she has been with the Inner Mongolia University of Science and Technology, where she is currently a Lecturer with the Department of Computer Science, School of Information Engineering. She has presided over one provincial and ministerial level scientific research project, one school-level project, four scientific research projects, and four academic articles, including one EI journal. Her main research interests include privacy protection and blockchain, machine learning security, image processing, and adversarial attacks of image.

Yongping Wang was born in Baotou, China, in 1984. She received the bachelor's degree in computer science and technology and the master's degree from the Wuhan University of Technology, in 2007 and 2010, respectively. Since 2010, she has been with the Inner Mongolia University of Science and Technology, where she is currently a lecturer with the School of Information Engineering. She has participated in a number of research projects of the Inner Mongolia Natural Science Foundation. Her main research interests include image processing, machine learning security and adversarial attacks of image.

Lu Gao was born in Baotou, China, in January 1979. She received the bachelor's degree in computer science and technology from Inner Mongolia Agricultural University, in 2001, and the master's degree from the Inner Mongolia University of Science and Technology, in 2012. Since 2001, she has been with the Inner Mongolia University of Science and Technology, where she is currently an Associate Professor with the School of Information Engineering. She has participated in a number of research projects of the National Natural Science Foundation of China and the Inner Mongolia Natural Science Foundation. Her current research interests include machine learning security, image processing, adversarial attacks of image.

Shuai Zhang was born in Dongying, China, in November 1996. He received the B.S. degree in computer science and technology from the Inner Mongolia University, in 2019. He is currently pursuing the master's degree in computer science and technology with the Inner Mongolia University of Science and Technology. His research

interests include adversarial attacks and defense based on image classification.

TTP-free Ownership Transfer Protocol Based on R_LWE Cryptosystem

Hong-Wei Qiu and Dao-Wei Liu
(Corresponding author: Hong-Wei Qiu)

Engineering College, Guangzhou College of Technology and Business
Guangzhou 510006, China
Email:qiuhwei123@163.com

(Received Aug. 24, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 17, 2023)

Abstract

Aiming at the privacy information disclosure problem caused by the change of ownership in the tag life cycle, ownership of RFID tags transfer protocol without a trusted third party is proposed. The protocol uses R_LWE password system to encrypt and transmit private information. Each communication entity can obtain the public key of each other and encrypt the message through the public key; the receiver decrypts the message using their private key to realize the security of private information. From different security attack aspect analyses, the security of the proposed protocol is high. Moreover, from calculation amount analysis at the tag end shows that the protocol cost calculation is suitable for the current RFID system.

Keywords: R_LWE Cryptosystem; RFID Technology; Trusted Third Party (TTP)

1 Introduction

Radio frequency identification is a technology that can read the data stored in a particular object without touching it. RFID can be tracked back to the beginning of the last century, due to the limitations of science and technology and other factors, it has not been widely developed. In this century, with the development of Cloud-computing, Big Data, Internet of Things and other new technologies, RFID has been widely developed and applied [1, 7]. In a typical RFID system, the tag is an essential entity, because the tag owner may change during its life cycle. For example, A first owns the tag, and after a period of time, A resells the tag to B. After B owns the tag, B has no right to access the privacy information stored in the tag by A. Correspondingly, after B owns the tag, A has no right to access any private information put into the tag during the use of B [11–13].

In reality, however, most of the time it's not what people wants it to be. That is, B may still have access to A's previously stored private information. In order to ensure the security of each user's private information, various ownership transfer protocols have been designed. All kinds of protocols can be classified into two categories: one is the ownership transfer protocol based on the trusted third party [8, 20], the other is the ownership transfer protocol without the participation of the trusted third party [2, 16]. The former protocol increase the number of communication entities due to the participation of the trusted third party, which makes the communication process more complicated and limited. The latter protocol reduces the number of communication entities and simplifies the communication process because there is no trusted third party involved, so it is widely used.

This paper proposes a lightweight ownership transfer protocol based on no trusted third party mechanism. There is no trusted third party to participate in the protocol, so the number of protocol communication entities is reduced, the protocol process is optimized to shorten the communication time. This protocol uses lightweight R_LWE [15] cryptosystem to encrypt information. R_LWE cryptosystem is a lightweight asymmetric algorithm that can disclose algorithm steps and public key. Session entities only need to store their own private key, which increase security and reduces the number of parameters to be stored.

2 Related Research Works

RFID tag ownership transfer protocol was first proposed by MOLNAR *et al.* in 2005 [10], which has epoch-making significance. However, there are some security defects in the protocol design process. For example, the lack of a tag to validate one of the parties allows a third party to launch a impersonation attack.

In reference [18], an ownership transfer protocol is designed based on the Chinese remainder theorem. The protocol has all aspects of security performance, but some message encryption process doesn't introduce random numbers, so the third party can implement reply attack. That is, by replying the previous round message, it can

pass the verification of the other party, there are security defects.

In reference [3], an ownership transfer protocol is presented based on physical unclonable function. This protocol solves the problem of impersonation attack well because it adopts unclonable function. However, due to the owner's failure to store multiple rounds of shared secret values, the third party can carry out asynchronous attacks. After a successful attack, third party can eavesdrop on the information and carry out tracking attacks.

In reference [14], an ownership transfer protocol is presented based on trusted third party. In terms of security, the protocol can resist all kinds of attacks and has very strong security performance. However, the introduction of a trusted third party makes the protocol more expensive than other protocols in terms of the number of communication entities, the communication process and the communication time. For tags with limited cost, this protocol is not widely available.

In reference [19], an ownership transfer protocol is proposed based on the difficult problems on elliptic curve. Firstly, from the perspective of computing load amount, the protocol could not be applied to tags with computational limitations. Secondly, some messages of the protocol have no random numbers to participate in the operation, which makes the protocol unable to provide backward-secure or forward-secure.

Due to the paper limitation of length, more ownership transfer protocols can be found in reference [4–6, 9, 17].

3 None Ownership Transfer Protocol of TTP

The ownership transfer protocol without a trusted third party based on the R_LWE cryptographic system (the specific algorithm steps of the R_LWE cryptographic system can be found in reference [15]) contains the original tag owner, the new tag owner and the tag communication entity.

The following are some symbols in protocol design:

- O_{new} is the new owner of the tag ownership.
- O_{old} is the original owner of the tag ownership.
- tag is the tag whose ownership is to be transferred.
- ID_{tag} is the unique identifier of tag.
- gy_{new} is the public key stored at one end of O_{new}.
- gy_{old} is the public key stored at one end of O_{old}.
- sy_{new} is the private key stored at one end of O_{new}.
- sy_{old} is the private key stored at one end of O_{old}.
- x_{new} is a random number generated from one end of O_{new}.
- y_{old} is a random number generated from one end of O_{old}.
- z_{new} is a random number generated from one end of O_{new}.
- t_{tag} is a random number generated from one end of tag.
- mmz is the secret value shared between O_{new} and O_{old}.
- mmz_{new}^{tag} is the secret value shared between tag and O_{new}.
- hh_i and xx_i are session messages.
- $\oplus$ is the nonequivalence operation.
- & is the and operation.
- $JIAM_{gy}(meg)$ is an encryption algorithm in the R_LWE cryptosystem, which uses the public key gy to encrypt information meg.
- $JIAM_{gy}(meg)$ is the decryption algorithm in the R_LWE cryptosystem, which uses the private key sy to decrypt information meg.

There are four stages to realize the secure transfer of tag ownership: initialization, ownership transfer request initiation, tag verification and key update.

1) Initialization Phase
The initialization phase mainly completes the initialization of the session entity before the ownership begins to transfer. After the initialization phase is complete, one end of O_{new} stores the following information:gy_{new}, sy_{new}, gy_{old}, and mmz; One end of O_{old} stores the following information:gy_{old}, sy_{old}, gy_{new}, mmz, and ID_{tag}; One end of stores the following information:ID_{tag} and gy_{new}.

2) Initiate the Ownership Transfer Request Phase
The initiate the ownership transfer request phase mainly to complete the verification of O_{old} to O_{new} and O_{new} to O_{old}. A diagram of this phase is shown in Figure 1.

A diagram of this phase is shown in Figure 1.

Combined with Figure 1, the steps can be described as follows:

Step one. O_{new} generates a random number x_{new} and computes $hh_1 = x_{new} \oplus mmz$, while sending message hh_1 to O_{old}.

Step two. O_{old} receives the message, processes hh_1 to get $x_{new} = hh_1 \oplus mmz$, and generates a random number y_{old}, calculates hh_2 and hh_3 in turn, and finally sends hh_2 and hh_3 to O_{new}.

$$\begin{aligned} hh_2 &= y_{old} \oplus mmz. \\ hh_3 &= JIAM_{gy_{new}}(gy_{old} \oplus x_{new}, y_{old} \& mmz). \end{aligned}$$

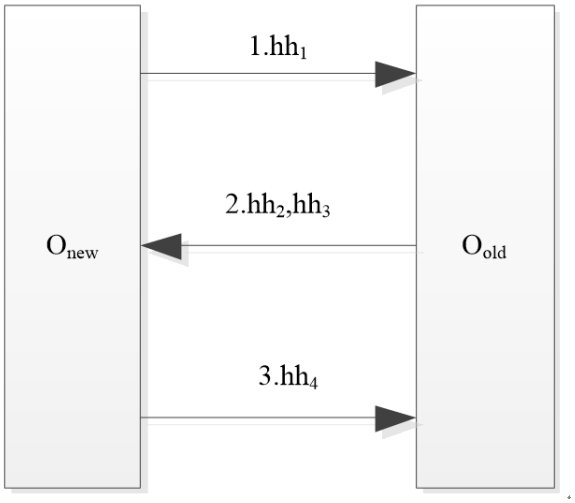


Figure 1: Initiate Ownership Transfer Request Phase Diagram

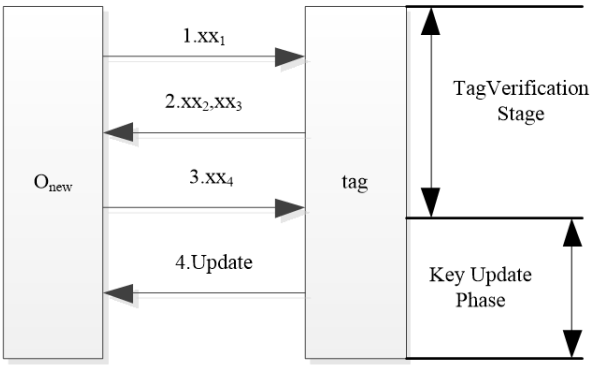


Figure 2: Tag Verification Phase and Key Update Phase Diagrams

Step three. O_{new} receiving the message, processes hh_2 to obtain $y_{old} = hh_2 \oplus mmz$. Then O_{new} decrypts message hh_3 with its own private key sy_{new} to obtain $gy^1_{old} = JIAM_{sy_{new}}(hh_3)$. Then it compares the relationship between the public key gy^1_{old} decrypted by O_{old} and the public key gy_{old} disclosed by O_{old}.

$gy^1_{old} \neq gy_{old}$, indicating that O_{old} can't pass the verification of O_{new} and the ownership stops transferring.

$gy^1_{old} = gy_{old}$, indicating that O_{new} has successfully verified O_{old}, and O_{new} continues to calculate hh_4 and finally sends hh_4 to O_{old}.

$hh_4 = JIAM_{gy_{old}}(gy_{new}\&mmz, x_{new}\&y_{old})$.

Step four. O_{old} receives the message and decrypts hh_4 to obtain $gy^1_{new} = JIEM_{sy_{old}}(hh_4)$. Then it compares the relationship between the public key gy^1_{new} decrypted by O_{new} and the public key gy_{new} disclosed by O_{new}.

$gy^1_{new} \neq gy_{new}$, indication that O_{new} can't pass the verification of O_{old} and the ownership stops transferring.

$gy^1_{new} = gy_{new}$, indicating that O_{old} has successfully verified O_{new}.

At this point, O_{new} and O_{old} can verify each other, and they can exchange information. O_{old} sends to O_{new} the ID_{tag} of tag whose ownership is to be transferred.

3) Tag Proof Phase

Tag proof phase is mainly to complete the verification between O_{new} and tag whose ownership is to be transferred. That is, O_{new} verifies tag and tag verifies O_{new}. A diagram of this phase is shown in Figure 2.

Combined with Figure 2, the steps of tag proof phase can be described as follows:

Step one. O_{new} again generates a random number z_new (which is used for verification between O_{new} and tag), then computes message xx_1 and sends it to tag.

$$xx_1 = z_{new} \oplus ID_{tag}.$$

Step two. tag receiving the message, processes xx_1 to obtain $z_{new} = xx_1 \oplus ID_{tag}$, then tag generates a random number t_{tag}, then starts to calculate messages xx_2 and xx_3, and finally sends xx_2 and xx_3 to O_{new}.

$$\begin{aligned} xx_2 &= t_{tag} \oplus ID_{tag}. \\ xx_3 &= JIAM_{gy_{new}}(t_{tag}\&ID_{tag}, z_{new}). \end{aligned}$$

Step three. O_{new} receiving the message and decrypts xx_3 first to obtain $t^2_{tag} = JIEM_{sy_{new}}(xx_3)$, then deforms xx_2 to obtain $t^2_{tag} = xx_2 \oplus ID_{tag}$, and then compares the size between t^1_{tag} and t^2_{tag}.

$t^1_{tag} \neq t^2_{tag}$, indicating that tag can't be verified by O_{new}, that is, tag is not a tag to be transferred, and the ownership stops transferring.

$t^1_{tag} = t^2_{tag}$, indicating that O_{new} has successfully verified tag, and O_{new} starts to calculate xx_4 and finally sends xx_4 to tag.

$$xx_4 = JIAM_{gy_{new}}(ID_{tag}, t_{tag}\&z_{new}).$$

Step four. tag receives the message, uses the same parameters to carry out the same algorithm to calculate $xx^1_4 = JIAM_{gy_{new}}(ID_{tag}, t_{tag}\&z_{new})$, and compares the size between xx^1_4 and xx_4.

$xx^1_4 \neq xx_4$, indicating that O_{new} can't pass the verification of tag and the ownership stops transferring.

$xx_4^1 = xx_4$, indicating that tag has successfully verified O_{new}, and tag can perform subsequent operations.

4) Key Update Phase
When the tag proof phase is completed, the verification between O_{new} and tag is realized, and the key update phase can be carried out at both ends of O_{new} and tag. The key update phase is still shown in Figure 2.

After the tag proof phase is complete, Tag starts to update the key, that is $mmz_{new}^{tag} = JIAM_{mmz}(t_{tag}, z_{new})$. After the key of tag is updated, tag sends $Update$ message to O_{new} to inform O_{new} that the key can be updated.

O_{new} receives the message and performs the same operations to update key $mmz_{new}^{tag} = JIAM_{mmz}(t_{tag}, z_{new})$.

At this point, the synchronized update of the shared key between O_{new} and tag is complete, and the ownership of tag is transferred, then the ownership of tag goes to O_{new}.

4 Protocol Security Analysis

This section analyzes the security and reliability of protocols based on common attack types.

1) Exclusivity
Exclusivity is to determine that a tag is the target tag whose ownership is to be transferred and not another tag. The proposed protocol can realize the exclusivity requirement in the label verification stage, which is detailed in step 3. After receiving the message sent by tag, O_{new} shall first verify the authenticity of the source party through xx_2 and xx_3. Only if the verification passes, can O_{new} state that tag is the tag whose ownership is to be transferred.

2) Reply Attack
Whether in the ownership transfer request initiation phase or in the tag verification phase, an attacker may launch a reply attack in an attempt to replay the previous round of messages to achieve the purpose of passing the verification of the other party. In order to avoid the occurrence of the above events, the proposed protocol will add different random numbers in the encryption of each message to ensure the freshness of each message. When the attacker replays the last round of message, the corresponding message value also changes because the random number used in this round of session changes. The message replayed by the attacker has the wrong value and can't be verified by the other party, so the attacker fails to attack.

3) Impersonation Attack
In the ownership transfer request initiation phase, the attacker can impersonate O_{new} or O_{old}. In the tag proof phase, the attacker can impersonate O_{new} or tag. Here, the attacker impersonates O_{old} as an example for analysis.

In the ownership transfer request initiation phase, the attacker impersonates O_{old} to talk with O_{new}, and O_{old} tries to send fake hh_2 and hh_3 messages to O_{new} in order to pass O_{new}'s verification and obtain more privacy information. When O_{new} receives the message, it first deforms message hh_2 to obtain the random number generated by the attacker, and then put the random number into message hh_3 and decrypts hh_3 to obtain the public key of the attacker. By comparison, it can be found that the value of the public key obtained through decryption is not equal to that of the public key published by O_{old}. Therefore, the impersonation of O_{old} by attacker can't pass the verification of O_{new} and fails to obtain any useful information.

4) Backward Secure
The attacker obtains the current session message in round i by listening, and attempts to crack and analyze the session message in various ways to obtain the privacy information used in the last round, namely round $i-1$. The protocol needs to prevent the attacker from carrying out this type of attack, which can be called backward-secure. The proposed protocol introduces random numbers to keep the message fresh, so as to resist the attack carried out by the attacker. Because random numbers are generated randomly, so the attacker can't reverse analyze the privacy information of the last round.

5) Forward Secure
The attacker obtains the session messages of the current round i through eavesdropping and attempts to calculate the session messages of the next round $i+1$ through prediction or forgery to pass the entity verification. The protocol needs to block this type of attack launched by attacker, this attack type can be called forward-secure. During the protocol design, each message will be encrypted with random number. Some messages have one random number added, while others have two random numbers added. Random numbers have randomness, heterogeneity and unpredictability, so the probability that the random number used by the attacker is the same as the random number used by the real entity is negligible, which makes the message sent by the attacker can't be verified by the other party.

6) Brute Force Attack
In this paper, messages hh_2 and hh_3 are selected as examples for detailed analysis of brute force attack.

The attacker can obtain the above two messages by various means. It can process hh_2 to obtain $y_{old} = hh_2 \oplus mmz$ (because the attacker doesn't have

the correct shared key, it is assumed that the attacker randomly selects a parameter as the value to participate in the cracking operation), and put the obtained random number into message hh_3 to decrypt it to obtain $gy_{old}^1 = JIEM_{sy_{new}}(hh_3)$. When the attacker gets to this point, things get complicated. For the attacker, two parameters can't be obtained at this time, one is the shared key value and the other is the random number generated by the new owner. The two parameters can't be known, which makes it impossible for the attacker to cite the correct value of either parameter, the attacker's idea of brute force attack fails.

Table 1: Security Requirements Comparison between Protocols

Attack Type	Ref [15]	Ref [6]	Ref [19]	Ref [20]	This protocol
Exclusivity	√	√	√	√	√
Reply Attack	×	√	√	√	√
Impersonation Attack	√	√	√	√	√
Backward Secure	√	√	√	×	√
Forward Secure	√	√	√	×	√
Brute Force Attack	√	√	×	√	√
Pursuit Attack	√	×	√	√	√

7) Pursuit Attack
The pursuit attack is that the attacker continuously monitors the session process and attempts to analyze the specific position of the tag in the continuous message, so as to launch the tracing attack on the tag and obtain the private information. When a tag sends a message, it is first encrypted and mixed with random numbers during encryption, which can ensure the real-time and freshness of the message and make the value of the message monitored by the attacker different from round to round. For the attacker, the attacker thinks that the position of the tag is in change, so they can't determine the true position of the tag, and the pursuit attack naturally can't be successfully implemented.

The security requirements of the protocol in this paper can be compared with those of other classical protocols, as shown in Table 1.

5 Protocol Performance Analysis

This section is mainly used to analyze the performance indicators of the communication entities in the protocol, such as computing load. According to the above description, O_{old} and O_{new} are not limited in terms of computing power and storage capacity, so they are not used as performance analysis objects. In this paper, the last *tag* will be selected as the research object, the computing load and storage capacity of *tag* will be studied.

Specific analysis can be seen in Table 2.

Table 2: Performance Analysis of Each Protocol Tag

Reference	Number of random numbers	Calculations	Memory space
Ref[15]	2	$3xor+5mod$	$3l$
Ref[6]	2	$5xor+3PUF$	$4l$
Ref[19]	3	$4xor+4hash(x)$	$2l$
Ref[20]	3	$6ECC(x)+1hash(x)$	$3l$
This protocol	1	$2xor+3E_h(x)$	$2l$

The meanings of the symbols in Table 2 above are as follows:xor represents the computing load of and operation; mod represents the computing load of modular operation (in-depth analysis of Chinese remainder theorem, its essence is modular operation);PUF represents the computing load of physical unclonable function; $hash(x)$ represents the computing load of hash function;$ECC(x)$ represents the computing load of elliptic curve encryption algorithm;$JIAM_{gy}(meg)$ represents the computing load of encryption algorithm in R_LWE cryptosystem.l indicates the parameter length.

In the above numerous calculations, the order of calculation quantity from small to large is $xor, xor, hash(x), PUF, mod, ECC(x)$. From this ranking, it can be seen that the overall computing load of xor and $JIAM_{gy}(meg)$ used in encryption should be significantly better than other protocols. The R_LWE cryptosystem is divided into two types: encryption algorithm and decryption algorithm. In general, calculation amount of the decryption algorithm is greater than that of the encryption algorithm. Therefore, this point is fully considered in the design process of the proposed protocol, so that only the encryption algorithm step is carried out at one end of the protocol tag, but not the encryption algorithm step. This reduces the computational burden and allows the protocol to be used in computationally restricted tags. In terms of storage and number of random numbers, the proposed protocol is better than or equal to some protocols. On the whole, the sum of computing load of the proposed protocol on the tag is less than that other protocols, which has certain advantages. This protocol can make up for some

hidden security problems existing in other protocols, so that the proposed protocol has better practicability and popularization under the same conditions.

6 Conclusion

Aiming at the security problems such as the change of ownership and the easy disclosure of privacy information of different users during the life cycle of tags, this paper proposes an ownership transfer protocol without the participation of trusted third parties. The protocol uses R_LWE cryptosystem to encrypt the information to be sent, which makes the computing load of the entity low and meets the high security at the same time. The proposed protocol is divided into four phases: initialization, ownership transfer request initiation, tag proof and key update. Different phases accomplish different objectives. To ensure information security, each step needs to verify the authenticity of the source first. From the perspective of multiple attack types analysis, it shows that the proposed protocol can resist a variety of common attacks and has relatively high security. From the computing load analysis at the tag, it shows that the proposed protocol is superior to other comparison protocols in terms of computing load.

Acknowledgments

This paper is supported by the build a virtual SDN environment based on VSphere (project number: pzxjyb51).

References

[1] M. Ajtai, C. Dwork, "A public-key cryptosystem with worst-case average-case equivalence," in *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, pp. 284–293, 1997.

[2] P. Y. Cui, "An improved ownership transfer and mutual authentication for lightweight RFID protocols," *International Journal of Network Security*, vol. 18, no. 6, pp. 1173–1179, 2016.

[3] Y. P. Duan, "Lightweight RFID group tag generation protocol," *Control Engineering of China*, vol. 27, no. 4, pp. 751–757, 2020.

[4] K. Fan, W. Jiang, H. Li, *et al.*, "Lightweight RFID protocol for medical privacy protection in IoT," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1656–1665, 2018.

[5] Q. Jiang, Z. R. Chen, B. Y. Li, *et al.*, "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1061–1073, 2018.

[6] W. Liang, S. Xie, J. Long, *et al.*, "A double puf-based RFID identity authentication protocol in service-centric internet of things environments," *Information Sciences*, vol. 503, pp. 129–147, 2019.

[7] R. Lindner, C. Peikert, "Better key sizes (and attacks) for LWE-based encryption," in *Processing of Topics in Cryptology-CT-RSA 2011*, pp. 319–339, 2011.

[8] D. W. Liu, J. Ling, "An improved RFID authentication protocol with backward privacy," *Computer Science*, vol. 43, no. 8, pp. 128–130, 2016.

[9] S. Q. Mei, X. R. Deng, "Mobile RFID bidirectional authentication protocol based on shared private key and bitwise operation," *Computer Applications and Software*, vol. 37, no. 7, pp. 302–308, 2020.

[10] D. Molnar, A. Soppera, D. Wagner, "A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags," in *Proceedings of the 12th International Conference on Selected Areas in Cryptograph*, pp. 276–290, 2005.

[11] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, 2011.

[12] G. F. Shen, S. M. Gu, and D. W. Liu, "An anti-counterfeit complete RFID tag grouping proof generation protocol," *International Journal of Network Security*, vol. 21, no. 6, pp. 889–896, 2019.

[13] F. Tan, "An improved RFID mutual authentication security hardening protocol," *Control Engineering of China*, vol. 26, no. 4, pp. 783–789, 2019.

[14] J. Q. Wang, Y. F. Zhang, and D. W. Liu, "Provable secure for the ultra-lightweight RFID tag ownership transfer protocol in the context of IoT commerce," *International Journal of Network Security*, vol. 22, no. 1, pp. 12–23, 2020.

[15] P. Wang, Z. P. Zhou, J. Li, "Improved serverless RFID security authentication protocol," *Journal of Frontiers of Computer Science and Technology*, vol. 12, no. 7, pp. 1117–1125, 2018.

[16] Y. Wei, J. Chen, "Tripartite authentication protocol RFID/NFC based on ECC," *International Journal of Network Security*, vol. 22, no. 4, pp. 664–671, 2020.

[17] R. Xie, B. Y. Jian, D. W. Liu, "An improved ownership transfer for RFID protocol," *International Journal of Network Security*, vol. 20, no. 1, pp. 149–156, 2018.

[18] R. Xie, J. Ling, D. W. Liu, "A wireless key generation algorithm for RFID system based on bit operation," *International Journal of Network Security*, vol. 20, no. 5, pp. 938–950, 2018.

[19] X. H. Zhao, "Attack-defense game model: Research on dynamic defense mechanism of network security," *International Journal of Network Security*, vol. 22, no. 6, pp. 1037–1042, 2020.

[20] F. Zhu, P. Li, H. Xu, *et al.*, "A lightweight RFID mutual authentication protocol with puf," *Sensors*, vol. 19, no. 13, pp. 2957–2978, 2019.

Biography

Hong-wei Qiu received a master's degree in School of Computer from Hubei University of Technology (China) in June 2017. He is now an associate professor and works in Guangzhou College of Technology and Business. His currnet research interest fields include information security and network security.

Dao-wei Liu received a master's degree in School of Computers from Guangdong University of Technology (China) in June 2016. His current research interest fields include information security.

An Efficient Heterogeneous Multi-message and Multi-receiver Signcryption IBC-CLC Scheme for Industrial Internet of Things

Pengshou Xie, Nannan Li, Zongliang Wang, Jiafeng Zhu, Pengyun Zhang, and Pengyun Zhang
(Corresponding author: Nannan Li)

School of Computer and Communications, Lanzhou University of Technology
No. 36 Peng Jia-ping road, Lanzhou, Gansu 730050, China
Email: 2500466296@qq.com

(Received July 29, 2022; Revised and Accepted Jan. 28, 2023; First Online Feb. 18, 2023)

Abstract

Industrial data security communication faces the problems of so many devices at the receiving end and low efficiency at the sending end, the high computational cost of existing signature schemes, and low security due to high dependence on data security channels. Given the problems, the paper proposes an efficient Heterogeneous Multi-message Multi-recipient Signcryption (HMMSC) and Identity Based Cryptosystem- Certificateless Cryptosystem (IBC-CLC)scheme for the Industrial Internet of Things by studying heterogeneous signcryption schemes, which combines multi-message multi-recipient mechanism to improve the efficiency of the sender in the message-intensive scenario. The proposed scheme is analyzed in terms of security, functional, and performance analysis in theory and simulation. The analysis results show that the proposed scheme is more efficient and secure than existing schemes.

Keywords: Heterogeneous Message; Identity-based Password System; Industrial Internet of Things; Signcryption; Without Bilinear Pairing

1 Introduction

Industrial Internet of Things is one of the popular research areas, which is a product of the integration of new generation information technology and the manufacturing industry. The industrial Internet is used to build a new fully connected manufacturing service system by fully interconnecting people, machines, and things [18]. Traditional industrial networks are often vulnerable due to insufficient security considerations, and the convergence with the Internet can lead to security issues such as a proliferation of vulnerabilities leading to frequent cyber-attacks. Cisco estimates that 500 billion IoT devices worldwide will be connected to the Internet by 2025 [15]. Efficient and secure data communication between control systems and edge device systems [2, 19–21] has become one of the priorities and hotspots for future industrial Internet security protection.

1.1 Related Works

The Multi-message and Multi-receiver Signcryption, first proposed by Seo and 1999 [11], allows the sender to send multiple different messages to multiple receivers at once with only one signature, and only the designated user can verify the validity of the message and decrypt it. In 2006, Duan *et al.* [10] proposed the first Identity-based Cryptosystem (IBC), based multi-recipient signing scheme with guaranteed recipient anonymity. The literature [1] proposed a multi-recipient certificate-free signing scheme based on bilinear pairs. Islam et al [6]proposed a certificate-free multi-recipient signing scheme without bilinear pairs. Pang *et al.* [8] constructed a certificate-free multi-message multi-recipient signing scheme based on Elliptic Curves Cryptography (ECC), which is better than Elliptic Curves Cryptography. Sun *et al.* [12] proposed a heterogeneous multi-recipient signing scheme for secure communication between IBC and Traditional Public Key Cryptosystem (TPKC). Niu *et al.* [17] proposed a multi-message multi-recipient signing scheme between IBC and CLC, a cryptosystem that combines a hybrid encryption mechanism to secure symmetric keys and data once and uses user pseudo-identity to protect them privacy.

The literature [6, 7] designed an efficient and certificateless multi-receiver signcryption scheme without bilinear pairings. The literature [9, 13] has proposed signature schemes for several fields, such as medical cyber physical system in standard model and Low-Power IoT Devices in a Wireless Sensor Network.

The literature [3] designed an efficient and certificateless conditional privacy-preserving authentication scheme. Literature [14] proposed a new convertible authenticated encryption scheme with message link-

ages.Their scheme provides confidentiality, unforgeability, and internal security. However, both of these schemes use bilinear pair operations, which makes their schemes inefficient. Heterogeneous means that it can be used for secure communication between network systems with different cryptographic systems. Currently, it is a challenge to study efficient, Heterogeneous Multi-message and Multi-receiver Signcryption schemes.

At present, there is no signcryption scheme for industrial Internet heterogeneous cryptosystem, and the existing heterogeneous signcryption scheme is often not applicable to industrial Internet due to its high overhead, and cannot meet the low overhead demand of its resource-constrained devices. The number of devices at the receiving end of the Industrial Internet is large, and the consumption of resources is huge and inefficient for the sender of communication data, and the Industrial Internet always has the disadvantage of limited resources and cannot adopt security protection measures with excessive overhead. Multi-message multi-recipient signing is an effective way to solve this problem. Multi-message multi-receiver signing can complete the encryption and signing process of different messages sent to different receivers in one logical step, which can protect data privacy and communication security, reduce the computational overhead at the sender's end, and not increase the computational overhead at the receiving device of communication data under the industrial Internet. For the industrial Internet one-to-many multi-message broadcast scenario, it is of great significance to design a one-to-many secure broadcast signing and encryption scheme that meets the requirements of industrial Internet heterogeneous cryptosystems to promote secure industrial Internet broadcast communication.

1.2 Our Contribution

In order to meet the needs of resource-constrained devices in the industrial Internet scenario, and to address the problems that existing schemes are inefficient and do not take into account the heterogeneity of industrial Internet cryptosystems, this paper proposes a bilinear pair-free, one-to-many, efficient heterogeneous multi-message broadcast signing scheme. The specific work of this paper is summarized as follows. The paper proposes an IBC-CLC heterogeneous, one-to-many multi-message signcryption scheme, referred to as HMMSC IBC-CLC scheme.

1) The IBC-CLC heterogeneous mechanism allows this paper to adapt to complex cryptosystems without the overhead of certificate management and storage.

2) The paper combines the multi-message multi-recipient mechanism, which allows the sender to send several different messages to different receivers securely and efficiently at one time, improving the efficiency of the sender in a message intensive scenario.

3) The paper use scalar multiplication operations on elliptic curves instead of bilinear pair operations to reduce the computational overhead and still transfer part of the receiver's overhead to the gateway safely, aiming to improve the performance of the receiver.

4) The paper improves the system's adaptability to complex environments by improving the private key extraction algorithm to reduce the system's dependence on secure channels and protect receiver user identity information using Lagrangian interpolation polynomials.

2 Preliminaries

This section describes related work including elliptic curve and lagrangian interpolation polynomials.

2.1 Elliptic Curve

An elliptic curve is the set of all points and an infinity point o (which can also be called a unit element) that satisfy the Weierstrass equation of Equation (1) in the projective plane.

$$Y^2Z + a_1XYZ + a_3Z^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \quad (1)$$

The elliptic curve equation is a chi-square equation, and the partial derivatives at any point on the curve are not simultaneously zero, let x=X/Z,y=Y/Z can be obtained from the general equation of the elliptic curve can be written as:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (2)$$

The use of elliptic curves in cryptography necessitates the discretization of continuous elliptic curves into discrete points, thus defining elliptic curves to a finite field.

2.2 Lagrangian Interpolation Polynomials

Lagrangian Unitary n - 1st polynomial

$$\begin{aligned} F(x) &= \sum_{i=1}^{n} F_i(x) = \sum_{i=0}^{n-1} a_i x^j \\ &= a_0 + a_1x + a_2x^2 + \cdots + a_{(n-1)}x^{(n-1)} (n > 1) \end{aligned} \quad (3)$$

consists of n groups of reciprocal points $(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)$.Of which

$$\begin{aligned} f_i(x) &= \frac{(x-x_1)\cdots(x-x_{t-1})(x-x_{t+1})\cdots(x-x_n)}{(x_t-x_1)\cdots(x_t-x_{t-1})(x_t-x_{n+1})\cdots(x_t-x_n)} \\ &= \prod_{j=1,j\neq i}^{n} \frac{x-x_j}{x_i-x_j} \\ &= \begin{cases} 1, x = x_i \\ 0, x \in \{x_i \mid i = 1, 2, \cdots, i-1, i+1, \ldots, n\} \end{cases} \end{aligned} \quad (4)$$

is the Lagrangian interpolation basis function. Then the paper has:

$$\begin{aligned} F_i(x) &= f_i(x)y_i \\ &= \begin{cases} 1, x = x_i \\ 0, x \in \{x_i \mid i = 1, 2, \cdots, i-1, i+1, \cdots, n_f\} \end{cases} \end{aligned} \quad (5)$$

2.3 Hash Function

Hash function can be of arbitrary length string input into fixed length output, its function can be written as $h = H(m)$, one of $h$is the result of the output, we call it the hash value or the message digest, H is hash function. The input space of a hash function is much larger than its output space (the hash value space), so different inputs may map to the same output. In addition, of arbitrary length message x, the hash value of the calculation on the software and hardware realization is easy, so I have availability hash function. For a hash function H, usually has the following properties:

1) Mono-direction: if a given arbitrary hash value h, want to find a message h makes $h = H(x)$, which is not feasible in the calculation. This means that there is no way to work backwards from the hash to the original message.

2) Weak collision resistance:if given any message x_1, want to find another message x_2, and indicates $x_1 \neq x_2$ make the two messages of hash values are equal, namely $H(x_1) = H(x_2)$ is not feasible in the calculation.

3) Resistance to strong collision:resistance to find any news x_1,x_2 and indicates $x_1 \neq x_2$, is to make the two messages hash values are equal, namely $H(x_1) = H(x_2)$ is not feasible in the calculation.

For a good hash function, a change in any bit of a message will cause its hash value to change dramatically. Hash function is widely used in cryptography, including encryption algorithm and digital signature algorithm. Hash functions are commonly used to authenticate messages in encryption algorithms. The hash value of the original file can be obtained by the hash function. The integrity verification of the file can be realized by checking and comparing the hash value of the file and the hash value later. In digital signatures, because the message digest is fixed in length and is usually much shorter than the message, signing the message digest is often more efficient and faster than signing the message directly.

2.4 Random Oracle Model

The proposal of this model makes the application of provable security theory in practical cryptography develop rapidly and becomes the basis of many effective security schemes in the future. A hash function that satisfies the following properties is called a Random Oracle(RO):

1) Uniformity: By RO output all y obey random uniform distribution.

2) Effectiveness:RO can compute its hash value in polynomial time for any input.

3) Determinacy: For the same input, the output must be the same.

The random predictor model enables the provable security theory to play its practical role in cryptography, and the security proof for various schemes has been recognized by the majority of researchers. However, since there is no random prophecy machine satisfying the above properties in the real environment, hash function is generally used to imitate the behavior of random prophecy machine, so the provable security scheme under the random prophecy machine model is widely considered to be safe in theory, but not necessarily safe in actual operation and implementation. Although the random Seer model is not actually secure, current researchers agree that a provably secure cryptosystem under random Seer is acceptable and persuasive.

3 The Proposed Scheme

The heterogeneous one-to-many signing scheme consists of four main algorithms: system initialization algorithm, private key extraction algorithm, and signing and decryption algorithm.

Step 1: System Initialization Algorithm

Enter the safety parameter λ and the system randomly selects a large prime number p, order number $q(q \geq pk$, k is a large integer). Then choose the elliptic curve E defined over the finite field Fp and choose the additive group G of order p on the elliptic curve, and note that the generating element of the group G is p. Due to the heterogeneity, this paper needs to be initialized in Private Key Generator (PKG) and Key Generating Center (KGC) respectively.

Step 2: Private Key Extraction Algorithm

The private key extraction algorithm is divided into two parts, which are the private key extraction algorithm under the identity-based system (IBC-KG) and the private key extraction algorithm under the certificate-free system (CLC-KG), as follows.

1) IBC-KG:
   a. The data sender DS sends the identity IDS to the PKG.
   b. The PKG randomly selects the integer $t_s \in Z_p^*$,and calculates $T_s = t_s P$,$d_s = t_s + s_1 h_s (modp)$,where $h_s = H_0(ID_S, T_s, P_1)$ is the binding value of user identity and public key. Then PKG binds the private key of user DS $SK_s = d_s$ through the secret channel and the public key PK through the public channel $PK_S = (ID_s, T_s)$ to DS.

2) CLC-KG:
   The data recipient DR_i under the certificateless cryptosystem needs to obtain his partial private key as well as the fully private and public keys by the following algorithm.
   a. Set the secret value: the data recipient DR_i, whose identity is identified as ID_i,

randomly selects the integer $x_i \in Z^*$ as its own secret value and calculates $X_i = x_i P$ as the partial public key value, and then sets $(ID_i,\ X_i\ )$ is sent to KGC for registration.

b. Extracting the partial private key: After KGC receives the registration message from DR_i, it first selects a random integer $t_i \in Z_p^*$, and calculates $T_i = t_i P$, $d_i = t_i + s_2 h_i (\mathrm{mod} p)$,where $h_i = H_1(ID_i, T_i, P_2)$. Then hide d_i into u_i, and calculate $u_i = d_i + H_1(ID_i, s_2 X_i, T_i)$. KGC sends both (T_i, u_i) to DR_i through the public channel.

c. Setting up the public key algorithm: when DR_i receives part of the public and private key information, it first verifies its authenticity and correctness by the following equation.

$$u_i P = T_i + h_i P + H_1(ID_i, x_i P_2, T_i) P \quad (6)$$

d. Setting private key algorithm: the data recipient DR_i extracts the partial private key from u_i by the following equation.

$$d_i = u_i - H_1(ID_i, x_i P_2, T_i) \quad (7)$$

Then set your own complete private key as $SK_i = (x_i, d_i)$.

Step 3: Signcryption Cryptography Algorithm

The data sender DS under an identity-based cryptosystem uses its own private key SKS with the system public parameter $param$, a set of data receivers $DR = \{DR_i | i = 1, 2, ..., n\}$ is selected and each DR_i corresponds to the public key PK_i, and each corresponds to the message m_i, and the sign-cipher algorithm is run to generate the sign-cipher text.

Step 4: Decryption Algorithm

The gateway receives the signed cipher message and uses the public key parameters of the sender DS to calculate the authentication parameters:

$$R_1' = v\,(hR_2 + t_s + h_s P_1) \quad (8)$$

Where $h_s = H_0\,(ID_s, T_s, h_s P_1)$, it is then constructed into a signed ciphertext $\delta = (\delta, R')$ which is then forwarded to the data recipient DR. The data recipient DR uses its own private key SKR to run the decryption algorithm.

4 Proof of Safety

This section gives proof of correctness for the proposed HMMSC IBC-CLC scheme.

4.1 Correctness Analysis

1) **Partial private key verification correctness**

In the private key extraction algorithm, this paper verifies that the recipient's partial private key is correct.

$$\begin{aligned} u_i &= (t_i + h_i + H_1\,(ID_i, s_2 X_i, T_i))\,P \\ &= t_i P + h_i s_2 P + H_1\,(ID_i, s_2 x_i P, T_i)\,P \\ &= T_i + h_i P + H_1\,(ID_i, x_i P_2, T_i)\,P \end{aligned} \quad (9)$$

The equation guarantees the correctness of the partial private key.

2) **Verify the correctness of the parameters**

The verification of the signature in the proposed signature-encryption algorithm is guaranteed by the equation, and the correctness of the calculation of the verification parameter $h' = h$. Therefore, it is necessary to prove the correctness of the computation of R', and the proof process is as follows.

$$\begin{aligned} R_1' &= \nu\,(hR_2 + T_s + h_s P_1) \\ &= (hr_2 + d_s)^{-1}\, r_1\,(hR_2 + T_s + h_s P_1) \\ &= r_1\,(hr_2 + t_s + s_1 h_s)^{-1}\,(hr_2 + t_s + s_1 h_s)\,P \\ &= r_1 P = R_1 \end{aligned} \quad (10)$$

The equation Verifies the correctness of the parameters.

3) **Correctness of key parameters**

In the sign and un-sign algorithms, this paper computes the key parameters U_i and U', respectively, uses the public and private keys of the data recipient DR_i, so this paper needs to prove. The proof process is as follows.

$$\begin{aligned} U_i' &= R_1'\,(x_i + d_i) = r_1 P\,(x_i + d_i) \\ &= r_1\,(x_i P + t_i P + h_i s_2 P) \\ &= r_1\,(X_i + T_i + h_i P_2) \\ &= r_1 h_i\left(h_i^{-1}\,(X_i + T_i) + P_2\right) \\ &= r_1 h_i\,(pk_i + P_2) = U_i \end{aligned} \quad (11)$$

The equation guarantees the correctness of the key parameters.

4.2 Proof of Safety

1) **Confidentiality**

Theorem 1. *Based on the assumption that the CDH hard problem holds, the HMMSC IBC-CLC scheme satisfies the IND-CCA2 security under the stochastic prediction machine model.*

Lemma 1. *Under the stochastic prediction machine model, if there exists a probabilistic polynomial-time adversary AI winning the game by a non-negligible margin ε, then there exists a challenger C who can win the game in polynomial time $\tau' \le \tau + O(nps + qpk + qd)\tau pm$ within*

$$\varepsilon' \ge \varepsilon\,(1 - q_s\,(nq_s + q_{H_2})\,/2^n)\,(1 - q_d/2^n) \quad (12)$$

The advantage of successfully solving the CDH difficulty problem.

2) **Unforgivable**

Lemma 2. *Under the stochastic prediction machine model, if there exists a probabilistic polynomial-time adversary A who wins the game by a non-negligible advantage CDH to win the game, then there exists a CDH hard problem that challenger C can successfully solve in polynomial time with the advantage of*

$$\varepsilon' \geq \varepsilon \left(1 - \frac{q_d q_{H_2}}{2^n}\right) \tag{13}$$

Proof. The security of the hypothetical probabilistic polynomial-time adversary A attack scheme. Challenger C and adversary A interact with the tuple $\langle P, aP, bP\rangle$ in an attempt to solve the CDH hard problem.

Initialization: Challenger C sets and sends the system public parameters to adversary A.A selects a set of target data receivers DR_i whose identity is $I = \{ID_i \mid i = 1, 2, \cdots, n\}$.

Challenger C maintains the lists LH_i and $\langle P, aP, bP\rangle$, which holds the random predicator queries for $H_i\,(i = 1, 2, \ldots, n)$, and the associated data generated during the key queries. The adversary A can perform the following polynomially bounded subinterrogation.

a. $(i = 0, 1, \cdots, 5)\, ask\, (qH_i)$;

b. Key inquiry: Enter ID_j for interrogation and challenger C checks if the list LPK already holds the relevant tuple. If it does, it is directly removed and returned to the adversary A.Otherwise, it is processed as follows.
If $ID_j \in I$, set $x_j = \perp$.
If $ID_j \notin I$, a randomly chosen integer t_j, $h_j \in Z^*$,set $x_j = a$. Save the associated tuple. Then, save the corresponding private key of ID_j as $SK_j = (x_j, d_j)$, where the partial private key $d_j = t_j + s_2 h_j (\mathrm{mod} p)$, the public key is preserved as $PK_j = (D_j, T_j, X_j, pk_j)$, where $T_j = t_j P$, $X_j = x_j P$, $pk_j = h - 1\,(T_j + X_j)$. Save the information to the list LPK.

c. Public key inquiry (qpk): input ID_j for interrogation and challenger C checks the list LPK for relevant tuple data, and if so, removes and returns the corresponding public key. Otherwise, Challenger C performs a key interrogation to obtain the relevant public key.

d. Secret value inquiry (qsv): input ID_j for interrogation, challenger C first checks $ID_j \in i$. If it belongs to the set I, then terminate and return the symbol $\perp$. Otherwise, the list LPK is queried for any relevant tuple data. If there is it takes out the corresponding secret value x_j for return; otherwise, challenger C performs a key interrogation to obtain the relevant secret value.

e. Partial private key inquiry ($qppk$): enter ID_j to interrogate and Challenger C checks the list LPK to see if it holds the relevant tuple data and if so, retrieves and returns the corresponding partial private key.Otherwise, Challenger C performs key interrogation to obtain the relevant partial private key.

f. Signed secret inquiry (qs): adversary A takes the sender identity ID_s, the receiver identity ID_j, and message M as input to perform a signed-encryption query, where $ID_j \notin I$. Challenger C runs the signed cipher algorithm and returns the signed cipher result.

g. Unsigned cipher query (qd): adversary A as sender ID_s, receiver identity ID_j, and the signature ciphertext $\sigma = (S, R_2, \upsilon, h, A)$ are used as input to perform the decryption query, where $ID_j \notin I$, challenger C runs the decryption algorithm and returns the result.

□

5 Performance Evaluation

5.1 Functional Analysis

In this subsection, this paper compares the proposed multi-message multi-recipient signature scheme with some existing related classical schemes for functional analysis, and the results are summarized as shown in Table 1. The literature [5] proposed a CLC multi-recipient sign-off scheme, but their scheme uses a bilinear pair operation with high overhead, which is not suitable for resource-constrained scenarios. The bilinear-pair-free certificate-free multi-recipient signing and encryption scheme proposed in the literature [4] is a significant improvement over the literature [5]. To support broadcast scenarios that require authentication and data protection, literature [16] and literature [17] propose multi-message multi-recipient sign-off schemes, respectively, and both consider heterogeneous cryptosystems, but their schemes are not efficient. In contrast, the certificate-free multi-message multi-recipient signing scheme proposed in [8] improves the efficiency to a large extent, but they do not consider the complex cryptosystem environment. In this paper, the scheme considers the cross-domain problem of IBC-CLC heterogeneous cryptosystem, does not use bilinear pair operation thus improving the efficiency, and does not require a secure transmission channel in the key genera-

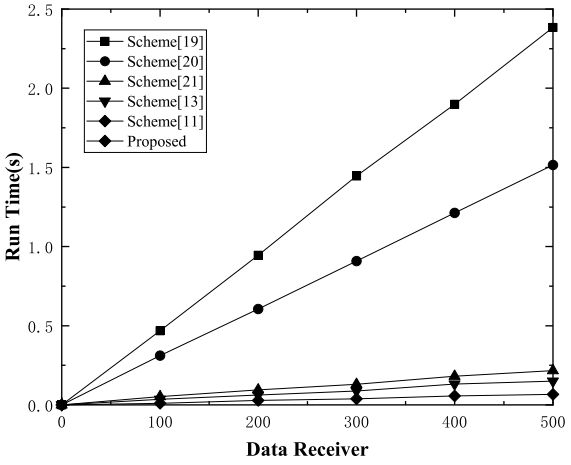


Figure 1: DS Signcryption phase computing

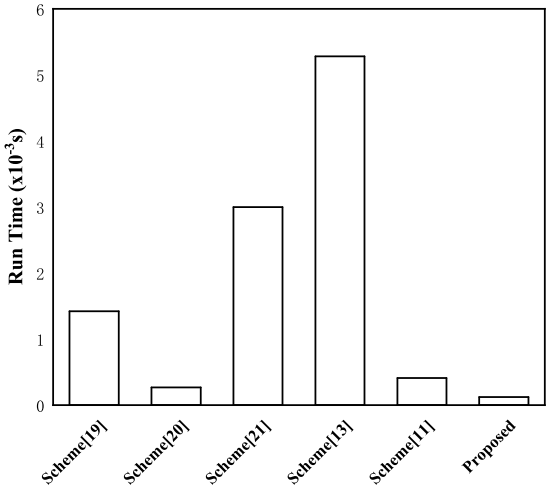


Figure 2: DR decryption phase computing time

tion phase of the certificateless cryptosystem, further improving the system's security.

5.2 Performance Analysis

In this subsection, the paper will analyze the performance and computational overhead of the proposed scheme. The required environment for the experimental simulation is carried out in the Ubuntu 18.04 system environment in VMware16 pro virtual machine, and the used computer parameters are Intel(R) Core(TM) i5-1240 CPU @1.7 GHz, RAM 16 GB. by theoretically analyzing the process of each scheme, the performance analysis of [4, 5, 8, 16, 17] of these five schemes and the performance analysis of this paper's scheme. In both stages of the signing and declassification algorithms, the comparison results are shown in Table 2.

In addition, since the scheme in this paper transfers part of the verification computation of the data receiver DR to the gateway GW, the computation overhead of the decryption algorithm operations, all of which require a large stage is divided into two parts, GW and DR so that the operations performed by the data receiver DR can be compared more clearly in each scheme.

1) **Performance Analysis of the Signed Encryption Phase**
From Table 2, it can be seen that the scheme of literature [5] requires n bilinear pairs, n exponential operations, n Map-to-Point hashing operations, and $(n+1)$ scalar multiplication operations, while the scheme of literature [17] requires $2n$ bilinear pairs and $2n$ exponential operations and $(n+2)$ scalar multiplication operations, all of which require a large computational overhead. The scheme of [4] requires $(3n+1)$ scalar multiplication operations and n dot addition operations, and the scheme of [16] requires $(n+1)$ exponential operations and n scalar multiplication operations.

The scheme in this paper requires only $(n+2)$ scalar multiplications and n dot additions, which is more efficient than the above-mentioned schemes, and only uses one more scalar multiplication operation than the scheme in [8]. It can be seen from the simulation that with 500 data receivers (i.e., $n = 500$), the scheme of [4] takes about 0.15s and the scheme of [16] takes about 2.38s, while the algorithm of this paper takes about 0.074s, which is 50.67% and 96.89% better than that of [16] and [5], respectively. As shown in Figure 1 and Figure 2, this paper has a more obvious advantage over the above schemes.

2) **Performance analysis of the decryption phase**
For the decryption algorithm stage, in the proposed scheme, the data receiver DR only needs to perform the decryption operation, and the intermediate gateway GW will assist in the calculation of the authentication parameters, so only one scalar multiplication operation is required at the data receiver side. Among the other schemes compared,Schemes [5, 16, 17] require bilinear pair operations at the receiver side, among which the scheme in [4] requires 1 bilinear pair and 1 scalar multiplication operation, the scheme in [17] requires 4 bilinear pairs, 1 scalar multiplication and 1 point addition operation, and the scheme in [16] requires 2 bilinear pairs, 2 scalar multiplication and 1 exponential operation. The scheme of [16] requires 2 bilinear pairs, 2 scalar multiplications and 1 exponential operation, and the operation time is more than 1ms. The scheme of [17] has the largest operation time, which is more than 5ms. Although the schemes of [4] and [8] are within 1ms because they do not require bilinear pair operations, they have higher overhead than the scheme of this

Table 1: Functional Analysis

Scheme	Scheme [5]	Scheme [4]	Scheme [16]	Scheme [17]	Scheme [8]	Proposed
Heterogeneous	×	×	√	√	×	√
Nocertifificated management burden	√	√	×	√	√	√
No bilinear pairs	×	√	×	×	√	√
Multiple messages	×	×	√	√	√	√
Multi-recipient	√	√	√	√	√	√
Weak dependency on secure channels	×	×	×	×	×	√

Table 2: Performance overhead comparison time

Scheme	Signature Algorithem	Dcryption algorithm	
		Gateway	Data Recipients(DR)
Scheme [5]	nTp+nTe+nTh+(n+1)Tpm	-	Tpm+Tp
Scheme [4]	(3n+1)Tpm+nTpa	-	2Tpm
Scheme [16]	(n+1)Te+nTpm	-	2Tp+2Tpm+Te
Scheme [17]	(n+2)Tpm+2nTp+2nTe	-	4Tp+Tpm+Tpa
Scheme [8]	(n+1)Tpm+nTpa	-	3Tpm+Tpa
proposed	(n+1)Tpm+nTpa	3Tpm+2Tpa	Tpm

paper.

6 Conclusions

The paper propose an efficient heterogeneous multi-message multi-recipient signing scheme by incorporating a multi-message multi-recipient signing mechanism. The paper improve the key generation algorithm in CLC to make the scheme less dependent on secure channels, more suitable for vulnerable communication environments, and use Lagrangian interpolation polynomials to protect the identity privacy of recipient users. In the stochastic prediction machine model, the security analysis in this paper proves that the proposed scheme satisfies confidentiality and unforgeability, and is more efficient than some existing schemes. In addition, the proposed signature confidentiality scheme is suitable for industrial Internet.

Acknowledgments

This study was supported by the National Science Council of Taiwan under grant NSC 95-2416-H-159-003. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] R. Behnia, A. A. Yavuz, M. O. Ozmen, and T. H. Yuen, "Compatible certificateless and identity-based cryptosystems for heterogeneous IoT," iIn *International Conference on Information Security*, pp. 39–58, 2020.

[2] H. Guo and L. Deng, "Certificateless ring signcryption scheme from pairings. *International Journal of Network Security*, vol. 22, no. 1, pp. 102–111, 2020.

[3] Z. Guo, "Cryptanalysis of a certificateless conditional privacy-preserving authentication scheme for wireless body area networks," *International Journal of Electronics and Information Engineering*, vol. 11, no. 1, pp. 1–8, 2019.

[4] D. He, H. Wang, L. Wang, J. Shen, and X. Yang, "Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices," *Soft Computing*, vol. 21, no. 22, pp. 6801–6810, 2017.

[5] Y. H. Hung, S. S. Huang, Y. M. Tseng, and T. T. Tsai, "Efficient anonymous multireceiver certificateless encryption," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2602–2613, 2015.

[6] S. K. H. Islam, M. K. Khan, and A. M. Al-Khouri, "Anonymous and provably secure certificateless multireceiver encryption without bilinear pairing," *Security and Communication Networks*, vol. 8, no. 13, pp. 2214–2231, 2015.

[7] L. Pang, M. Kou, M. Wei, and H. Li, "Efficient anonymous certificateless multi-receiver signcryption scheme without bilinear pairings," *IEEE Access*, vol. 6, pp. 78123–78135, 2018.

[8] L. Pang, M. Wei, and H. Li, "Efficient and anonymous certificateless multi-message and multi-receiver signcryption scheme based on ECC," *IEEE Access*, vol. 7, pp. 24511–24526, 2019.

[9] R. Y. Patil and Y. H. Patil, "Identity-based signcryption scheme for medical cyber physical system in standard model," *International Journal of Information Technology*, vol. 14, pp. 2275–2283, 2022.

[10] C. Peng, J. Chen, M. S. Obaidat, P. Vijayakumar, and D. He, "Efficient and provably secure multireceiver signcryption scheme for multicast communication in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6056–6068, 2019.

[11] M. Seo and K. Kim, "Electronic funds transfer protocol using domain-verifiable signcryption scheme," in *International Conference on Information Security and Cryptology*, pp. 269–277, 1999.

[12] Y. X. Sun and H. Li, "Efficient signcryption between tpkc and idpkc and its multi-receiver construction," *Science China Information Sciences*, vol. 53, no. 3, pp. 557–566, 2010.

[13] P. Y. Ting, J. L. Tsai, and T. S. Wu, "Signcryption method suitable for low-power iot devices in a wireless sensor network," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2385–2394, 2017.

[14] S. F. Tzeng, Y. L. Tang, and M. S. Hwang, "A new convertible authenticated encryption scheme with message linkages," *Computers & Electrical Engineering*, vol. 33, no. 2, pp. 133–138, 2007.

[15] A. Upadhyaya, C. Mistry, D. Kedia, D. Pal, and R. De, "Applications and accomplishments in internet of things as the cutting-edge technology: An overview," *Brainwave: A Multidisciplinary Journal*, vol. 3, pp. 1-11, 2022.

[16] C. Wang, C. Liu, Y. Li, H. Qiao, and L. Chen, "Multi-message and multi-receiver heterogeneous signcryption scheme for ad-hoc networks," *Information Security Journal: A Global Perspective*, vol. 26, no. 3, pp. 136–152, 2017.

[17] C. Wang, C. Liu, S. Niu, L. Chen, and X. Wang, "An authenticated key agreement protocol for cross-domain based on heterogeneous signcryption scheme," in *13th International Wireless Communications and Mobile Computing Conference (IWCMC'17)*, pp. 723–728, IEEE, 2017.

[18] Y. Wu, N. Guo, B. Wang, and L. Zhang, "Research on situational awareness technology of industrial control network based on big data," *Journal of Physics: Conference Series*, vol. 2216, no. 1, pp. 012079, 2022.

[19] F. Yan, L. Xing, and Z. Zhang, "An improved certificateless signature scheme for iot-based mobile payment," *International Journal of Network Security*, vol. 23, no. 5, pp. 904–913, 2021.

[20] M. Zhao and Y. Peng, "A novel certificateless aggregation signcryption scheme under cloud computing," *International Journal of Network Security*, vol. 23, no. 2, pp. 238–245, 2021.

[21] F. Zhou, Y. Li, and C. Lin, "A revocable certificateless aggregate signature scheme with enhanced security," *International Journal of Network Security*, vol. 22, no. 4, pp. 645–654, 2020.

Biography

Pengshou Xie was born in 1972. He is a professor and a supervisor of master student at Lanzhou University of Technology. His major research field is Security on Internet of Things. E-mail: xiepsh lut@163. com

Nannan Li was born in Feb.1 997. She is a master student at Lanzhou University of Technology. Her major research field is network and information security. E-mail: 2500466296@qq. com

Zongliang Wang was born in Mar. 1997. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 1292094887@qq.com

Jiafeng Zhu was born in Jan. 1997. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: zhujiafeng688@163.com

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

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

An Improvement of Babai's Rounding Procedure for CVP

Shuying Yang
Department of Data and Computer Science, Shandong Women's University
No. 2399, University Road, Jinan 250022, Shandong, P. R. China
Email: ysystudy2005@163.com

(Received Aug. 15, 2022; Revised and Accepted Dec. 1, 2022; First Online Feb. 18, 2023)

Abstract

CVP, together with SVP, are two of the central problems in lattice-based cryptography. Their hardness paved the way for the proposals of many different lattice-based cryptographic schemes. Meanwhile, efficient algorithms for solving or approximately solving these problems have become essential tools in public key cryptanalysis and have successfully attacked many cryptosystems. Among these algorithms, Babai's rounding procedure is one of the most classic. Recently, the rounding procedure has been implemented in RNS and improved using optimal Hermite Normal Form lattices. In this paper, we propose four improved rounding procedures to approximately solve the CVP problem based on the famous Babai's rounding procedure and Gram-Schmidt orthogonalization technique. The first two procedures are general for any lattice basis, while the latter two algorithms are unique versions of the first two for which the input basis is in HNF form. We also show that all four algorithms perform better than Babai's procedure concerning errors by examples and experiments, although some efficiency loss is taken as the cost.

Keywords: Closest Vector Problem (CVP); Hermite Normal Form (HNF); Rounding Procedure

1 Introduction

Recently in the literature of cryptology, lattice-based cryptographic constructions hold a great promise for post-quantum cryptography [3,4,11–13,15,21], since they enjoy relatively strong security proofs, remarkable efficient implementations, as well as very simplicity. Furthermore, lattice-based cryptography is believed to be quantum resistant. At the bottom of the constructions of lattice-based cryptography, however, it is the hardness of the computational problem in lattices that lays a secure foundation [9, 20].

The most two basics of computational problems in lattices are the *Shortest Vector Problem*(SVP) and the *Closest Vector Problem*(CVP) [19]. The *Shortest Vector Problem* asks to find the shortest nonzero vector in a lattice L, however, the *Closest Vector Problem* is the inhomogeneous version of SVP, and asks to find the lattice point closest to a given target. The CVP has been proved to be NP-complete by reduction from subset sum [10], and therefore no algorithm can solve CVP in deterministic polynomial time, unless $P = NP$ [10]. Reducing CVP to SVP is an interesting problem [7], as it is widely believed that SVP is not harder than CVP, and many even believe that SVP is strictly easier. Empirical evidence to these beliefs is provided by the gap between known hardness results for both problems. Whearas it is relatively easy to establish the NP-hardness of CVP, the question of whether SVP is NP-hard was open for almost two decades, originally conjectured in [10] and resolved in the affirmative in [5], and only for randomized reductions.

The hardness of solving SVP and CVP has led reseachers to consider approximation versions of these problems [19]. Approximation algorithms return solutions that are only guaranteed to be within some specified factor γ from the optimal. Approximation CVP in n-dimensional lattices is known to be NP-hard for any constant approximation factor or even some slowly increasing function of the dimension [14].

However, NP-hardness itself does not exclude the possiblity of sub-exponential time algorithms. To rule out such algorithms, several hypothesis have been introduced, such as the Strong Exponential Time Hypothesis (SETH), the Exponential Time Hypothesis(ETH), or the Gap-Exponential Time Hypothesis (Gap-ETH), and are by now quite standard. Based on these, a few recent results have shown hardness for approximation CVP and SVP are closely related [1, 2].

On the algorithmic side, known polynomial-time algorithms like the one of LLL algorithm [16] and its descendants such as [6] obtain slightly subexponential approximation factors $\gamma = 2^{\Theta(n \log\log n/\log n)}$ for SVP. Based on these algorithms, Babai's nearest plane algorithm can obtain a similar approximation factor for CVP. In [7], Babai also propose a rounding procedure for CVP which is more efficient, although it loses a little approximation factor. Recently, the rounding procedure has been implemented

in RNS [8] and improved with the use of lattices of optimal Hermite Normal Form [17, 18]. In this paper, we impove the Babai's rounding procedure.

Our Contributions. We provide four improved rounding procedures and show that the proposed rounding procedures are stronger than Babai's rounding procedure with respect to the appoximation parameters.

1) The first two improved rounding procedures are general for arbitrary lattice basis. The underlying main ideas are inspired by the following three observations: The first one is that every lattice basis can be easily decomposed as multiplication of a orthogonal basis (a basis of $\mathbb{R}^n$ for some n, not necessarily a lattice basis) and an upper triangular matrix by Gram-Schmidt orthogonalization procedure. Second, the orthogonal basis makes it easy to quantify the distance between vector in lattice and the input target vector. The third observation is that optimizing this distance can be done by appropriately choosing round up or round down of corresponding parameters.

2) The second two improved rounding procedures are special for lattice basis in its Hermite Nornal Form (HNF). The underlying main ideas are based on the observation: The special form of the HNF basis makes the Gram-Schmidt orthogonal decomposition of this basis very special. The obtained orthogonal basis is a digonal matrix, and the diagonal elements are the corresponding diagonal elements of the original HNF basis. Furthermore, the resulting upper triangular matrix has diagonal entries of 1, and all nonzero entries are positive. These characteristics make our algorithm easy to do some checking operations, thus improving the efficiency. Since every lattice has a unique HNF basis which can be efficiently computed from any basis of the lattice, the second two rounding algorithms can actually be made general easily, just by converting the basis to an HNF basis.

2 Preliminaries

Notation. We use $\mathbb{Z}$ for the set of integers, and $\mathbb{R}$ for the set of real numbers. Elements of these sets are denoted by lowercase letters. We use uppercase letters to denote matrices M, and usually arrange a set of vectors in columns into a matrix. We denote the i-th coordinate of vector v by v_i.

2.1 Lattices

Definition 1. *A lattice is defined as the set of all integer linear combinations*

$$L(b_1, \cdots, b_n) = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \text{ for } 1 \le i \le n \right\}$$

of n linearly independent vectors $b_1, \cdots, b_n$ in $\mathbb{R}^d$, where d and n are called dimension and rank of the lattice. If $n = d$, the lattice is called full rank. The set of vectors $b_1, \cdots, b_n$ is called a basis for the lattice. A basis is usually represented by the matrix $B = [b_1, \cdots, b_n] \in \mathbb{R}^{d \times n}$ having the basis vectors as columns.

When studying lattices from an algorithm point of view, it is customary to assume that the basis vectors (and therefore any lattice vector) have all rational coordinates. Moreover, by appropriately scaling the lattice, all rational lattices can be easily converted to integer lattices. So, without loss of generality, we concentrate on integer lattices.

2.2 Minimum Distance

Definition 2. *For any lattice L, the minimum distance of L is the smallest distance between any two lattice points:*

$$\lambda(L) = \inf\{\|x - y\| : x, y \in L, x \neq y\}$$

Equivalently, the minimum distance can also be defined as the length of the shortest nonzero lattice vector:

$$\lambda(L) = \inf\{\|v\| : v \in L \backslash 0\}$$

Minkowski's first theorem states the following with regards to the minimun distance:

Theorem 1. *For any rank n lattice L, the length of the shortest nonzero vector satisfies $\lambda(L) < \sqrt{n} \det(L)^{1\backslash n}$.*

2.3 Computational Problems

Minkowski's first theorem implies that any lattice of rank n contains a nonzero vector of length at most $\sqrt{n} \det(L)^{1\backslash n}$. Its proof, however, is non-constructive: it does not give us an algorithm to find such a lattice vector. To discuss such computational issues, let us define the most two basic computational problems involving lattices.

Definition 3 (Approximate SVP)**.** *Given a lattice basis $B \in \mathbb{Z}^{d \times n}$, find a nonzero lattice vector $Bx (x \in \mathbb{Z}^n \backslash 0)$ such that $\|Bx\| \le \gamma \lambda(L(B))$. In particular, the problem is called SVP if $\gamma = 1$.*

Definition 4 (Approximate CVP)**.** *Given a lattice basis $B \in \mathbb{Z}^{d \times n}$ and a target vector $t \in \mathbb{Z}^m$, find a lattice vector $Bx (x \in \mathbb{Z}^n)$ such that $\|Bx - t\| \le \gamma \|By - t\|$ for any other $y \in \mathbb{Z}^n$. In particular, the problem is called CVP if $\gamma = 1$.*

3 Babai's Rounding Procedure

In this section, we provide a brief overview of Babai's rounding procedure (Algorithm 1). Given an arbitrary lattice basis B and a target t, in order to find a lattice point close to a target t we may

- first apply the inverse transformation B^{-1} to get $B^{-1}t$,
- round $B^{-1}t$ to the closest integer vector $\lfloor B^{-1}t \rceil \in \mathbb{Z}^n$,
- map the resulting integer vector to the lattice point $v = B\lfloor B^{-1}t \rceil$.

Algorithm 1 Babai's Rounding Procedure

Input:
Lattice basis B, and vector $t \in \mathbb{R}^n$

Output:
1: Compute B^{-1} and get $B^{-1}t$;
2: Round $B^{-1}t$ to the closest integer vector $\lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
3: **return** the resulting integer vector to the lattice point $v = B\lfloor B^{-1}t \rceil$.

In order to analyse Babai's algorithm easily, we introduce the two quantities

$$s_{\min} = \min_{x \in \mathbb{R}^n} \|Bx\| / \|x\|$$

$$s_{\max} = \max_{x \in \mathbb{R}^n} \|Bx\| / \|x\|$$

which express by how much the transformation B can shrink or expand the length of a vector.

Theorem 2. *Babai's rounding procedure always outputs a lattice point within distance $\sqrt{n} \cdot S_{\max}/2$ from t.*

Proof.

$$\begin{aligned}
&\left\|B\lfloor B^{-1}t \rceil - t\right\| \\
&= \left\|B(\lfloor B^{-1}t \rceil - B^{-1}t)\right\| \\
&= \left\|B\frac{\lfloor B^{-1}t \rceil - B^{-1}t}{\|B(\lfloor B^{-1}t \rceil - B^{-1}t)\|} \cdot \left\|B(\lfloor B^{-1}t \rceil - B^{-1}t)\right\|\right\| \\
&= \left\|(\lfloor B^{-1}t \rceil - B^{-1}t)\right\| \cdot \left\|B\frac{\lfloor B^{-1}t \rceil - B^{-1}t}{\|B(\lfloor B^{-1}t \rceil - B^{-1}t)\|}\right\| \\
&\le \left\|(\lfloor B^{-1}t \rceil - B^{-1}t)\right\| \cdot S_{\max}(B) \\
&\le \frac{1}{2}\left\|\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}\right\| \cdot S_{\max}(B) \\
&= \frac{\sqrt{n}}{2} \cdot S_{\max}(B)
\end{aligned}$$

where the first inequality is due to the definition of $S_{\max}(B)$, and the second inequality is since the rule of notation $\lfloor \cdot \rceil$. □

Theorem 3. *Let t is within distance $S_{\min}/2$ from the lattice, then Babai's rounding procedure returns the (necessarily unique) lattice point within distance $S_{\min}/2$ from t.*

Proof. Since t is within distance $S_{\min}/2$ from the lattice, there exists a integer vector $Y \in \mathbb{Z}$ such that

$$\|BY - t\| \le \frac{1}{2}S_{\min}(B)$$

Hence,

$$\begin{aligned}
&\frac{1}{2}S_{\min}(B) \\
&\ge \|BY - t\| \\
&= \left\|B(Y - B^{-1}t)\right\| \\
&= \left\|B\frac{Y - B^{-1}t}{\|Y - B^{-1}t\|} \cdot \left\|Y - B^{-1}t\right\|\right\| \\
&= \left\|Y - B^{-1}t\right\| \cdot \left\|B\frac{Y - B^{-1}t}{\|Y - B^{-1}t\|}\right\| \\
&\ge \left\|Y - B^{-1}t\right\| \cdot S_{\min}(B),
\end{aligned}$$

Therefore, $\left\|Y - B^{-1}t\right\| \le \frac{1}{2}$. We claim that the absolution of each coordinate of vector $Y - B^{-1}t$ is less than $\frac{1}{2}$. Otherwise, the norm of vector $Y - B^{-1}t$ will be strictly greater than $\frac{1}{2}$ which contracts to $\left\|Y - B^{-1}t\right\| \le \frac{1}{2}$. Since $Y \in \mathbb{Z}$ is an integer vector, $Y = \lfloor B^{-1}t \rceil$. This also shows that t which is within distance $S_{\min}/2$ from the lattice is necessarily unique. □

4 Improved Rounding Procedure

In this section,we provide Four improved rounding procedures. Two are general for arbitrary lattice basis, the other two special for lattice basis in its Hermite normal form. And we also show that the proposed rouding procedures are stronger than Babai's rounding procedure with respect to the appoximation parametres.

4.1 General Version

Given an arbitrary lattice basis B and a target t, in order to find a lattice point close to a target t we conduct the following steps (Algorithm 2).

1) First compute the Gram-Schmidt orthogonalization B^* of basis B and the corresponding upper-triangular matrix Λ such that $B = B^* \cdot \Lambda$;
2) Apply the inverse transformation B^{-1} to get $B^{-1}t$;
3) Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
4) Compute the vector $w = u - B^{-1}t$;
5) If $\lambda_{ij} \cdot w_j \ge 0$ for all $j > i$, goto step 6); Otherwise, goto step 7);
6) For j from 1 to $n-1$, if $|w_j| = 0.5$, set $w_j = -w_j$. Then, goto step 7) ;
7) Map the resulting integer vector to the lattice point $v = B^{-1}t + w$.

Theorem 4. *Given an arbitrary lattice B and a target vector t. Let v and $v^\dagger$ be the outputs of Algorithm 1 and Algorithm 2, respectively. Then the distance of t from*

Algorithm 2 General Edition I

Input:
Lattice basis B, and vector $t \in \mathbb{R}^n$

Output:

1: Compute the GS-basis B^* of B and Λ such that $B = B^*\Lambda$;
2: Compute B^{-1} and get $B^{-1}t$;
3: Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
4: Compute the vector $w = u - B^{-1}t$;
5: $k = 0$;
6: **for** $i = 1$ to n **do**
7: **for** $j = i$ to n **do**
8: **if** $\lambda_{ij} \cdot w_j \geq 0$ **then**
9: k++;
10: **end if**
11: **end for**
12: **end for**
13: **if** $k = \frac{n(n+1)}{2}$ **then**
14: **for** $j = 1$ to $n-1$ **do**
15: **if** $|w_j| = 0.5$ **then**
16: $w_j = -w_j$;
17: **end if**
18: **end for**
19: **end if**
20: **return** the resulting integer vector to the lattice point $v = B(B^{-1}t + w)$

$v^\dagger$ is not greater than that of t from v. In particular, if there is at least one coordinate of w appeared in Algorithm 2 whose absolute value is exactly the value 0.5, then the distance of t from $v^\dagger$ will be strictly smaller than that of t from v.

Proof. Due to the step 1 in Algorithm 2, B^* is the Gram-Schmidt orthogonalization of basis B, and Λ is the corresponding upper-triangular matrix with 1 on it's principal diagonal such that $B = B^* \cdot \Lambda$. Denote

$$B^* = [b_1^*, b_2^*, \cdots, b_n^*],$$

and

$$\Lambda = \begin{bmatrix} 1 & \lambda_{12} & \cdots & \lambda_{1n} \\ 0 & 1 & \cdots & \lambda_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Since v is the output of Algorithm 1, the square of the distance between the input vector t and v can be presented as

$$\begin{aligned} \|t - v\|^2 &= \left\|B(B^{-1}t - \lfloor B^{-1}t \rceil)\right\|^2 \\ &= \left\|B^*\Lambda(B^{-1}t - \lfloor B^{-1}t \rceil)\right\|^2 \\ &= \|B^*\Lambda y\|^2 \end{aligned}$$

where $B^{-1}t - \lfloor B^{-1}t \rceil$ is denoted by y, which is also equal to the vector w in the step 4 of Algorithm 2. After finishing excuting all steps of Algorithm 2, the vector w would be exchanged. We denote w at this point as $y^\dagger$. Since $v^\dagger$ is the output of Algorithm 2, the square of the distance between the input vector t and $v^\dagger$ can be presented as

$$\begin{aligned} \left\|t - v^\dagger\right\|^2 &= \left\|t - B(B^{-1}t - w)\right\|^2 \\ &= \left\|t - B(B^{-1}t - y^\dagger)\right\|^2 \\ &= \left\|By^\dagger\right\|^2 \\ &= \left\|B^*\Lambda y^\dagger\right\|^2 \end{aligned}$$

Now we denote Λy and $\Lambda y^\dagger$ by x and $x^\dagger$ respectively, and analyse their relation due to the process between Step 5 and Step 19 in Algorithm 2. The functionality from step 5 to step 12 is to check whether $\lambda_{ij} \cdot w_j \geq 0$ for all $j > i$. If so, set $w_j = -w_j$ when $|w_j| = 0.5$ for all j; Otherwise, follow Babai's rounding procedure without any change. Thus

$$\begin{aligned} x_i^\dagger &= \sum_{j=1}^{j=n} \lambda_{ij} \cdot y_j^\dagger = \sum_{j=i}^{j=n} \lambda_{ij} \cdot y_j^\dagger \leq |\sum_{j=i}^{j=n} \lambda_{ij} \cdot y_j^\dagger| \\ &\leq \sum_{j=i}^{j=n} |\lambda_{ij} \cdot y_j^\dagger| = \sum_{j=i}^{j=n} |\lambda_{ij} \cdot y_j| = \sum_{j=1}^{j=n} |\lambda_{ij} \cdot y_j| = x_i. \end{aligned}$$

Therefore,

$$\begin{aligned} \left\|t - v^\dagger\right\|^2 &= \left\|B^*\Lambda y^\dagger\right\|^2 = \left\|B^* x^\dagger\right\|^2 \\ &= |x_1^\dagger| \left\|b_1^*\right\|^2 + |x_2^\dagger| \left\|b_2^*\right\|^2 + \cdots + |x_n^\dagger| \left\|b_n^*\right\|^2 \\ &\leq |x_1| \left\|b_1^*\right\|^2 + |x_2| \left\|b_2^*\right\|^2 + \cdots + |x_n| \left\|b_n^*\right\|^2 \\ &= \|B^* x\|^2 = \|B^*\Lambda y\|^2 = \|t - v\|^2 \end{aligned}$$

where the third and fourth equalities follow from the fact that B^* is the Gram-Schmidt orthogonalization of basis B.

In particular, if there is at least one coordinate of w appeared in Algorithm 2 whose absolute value is exactly the value 0.5, Step 16 in Algorithm 2 will be performed. Thus, there is at least one coordinate $x_i^\dagger$ which would be strictly smaller than it's corresponding element x_i, i.e. $x_i^\dagger < x_i$. Then

$$\begin{aligned} \left\|t - v^\dagger\right\|^2 &= \left\|B^*\Lambda y^\dagger\right\|^2 = \left\|B^* x^\dagger\right\|^2 \\ &= |x_1^\dagger| \left\|b_1^*\right\|^2 + |x_2^\dagger| \left\|b_2^*\right\|^2 + \cdots + |x_n^\dagger| \left\|b_n^*\right\|^2 \\ &< |x_1| \left\|b_1^*\right\|^2 + |x_2| \left\|b_2^*\right\|^2 + \cdots + |x_n| \left\|b_n^*\right\|^2 \\ &= \|B^* x\|^2 = \|B^*\Lambda y\|^2 = \|t - v\|^2 . \end{aligned}$$

□

Theorem 5. *Given an arbitrary lattice B and a target vector t. Let v and $v^\ddagger$ be the outputs of Algorithm 1 and Algorithm 3, respectively. Then the distance of t from $v^\ddagger$ is not greater than that of t from v. In particular, if there is at least one coordinate of w appeared in Algorithm 2 whose absolute value is exactly the value 0.5, then the distance of t from $v^\ddagger$ will be strictly smaller than that of t from v.*

Algorithm 3 General Edition II

Input:
Lattice basis B, and vector $t \in \mathbb{R}^n$
Output:
1: Compute the GS-basis B^* of B and Λ such that $B = B^*\Lambda$;
2: Compute B^{-1} and get $B^{-1}t$;
3: Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
4: Compute the vector $w = u - B^{-1}t$;
5: $k = 0; l = 0; r = 0$
6: **for** $i = 1$ to n **do**
7: **for** $j = i$ to n **do**
8: **if** $\lambda_{ij} \cdot w_j \geq 0$ **then**
9: k++;
10: **end if**
11: **end for**
12: **end for**
13: **if** $k = \frac{n(n+1)}{2}$ **then**
14: **for** $j = n - 1$ to 1 **do**
15: **if** $|w_j| = 0.5$ **then**
16: $l+ = w_j$;
17: $r+ = -w_j$;
18: **for** $i = j + 1$ to n **do**
19: $l+ = \lambda_{ji} \cdot w_i$;
20: $r+ = \lambda_{ji} \cdot w_i$;
21: **end for**
22: **if** $\mid l \mid \geq \mid r \mid$ **then**
23: $w_j = -w_j$;
24: **end if**
25: **end if**
26: **end for**
27: **end if**
28: **return** the resulting integer vector to the lattice point $v = B(B^{-1}t + w)$

Proof. For simplicity, we continue to use the notations in the proof of Theorem 3. Same as Algorithm 2, the vector w in Step 4 of Algorithm 3 is also equal to y. After finishing excuting all steps of Algorithm 3, the vector w would be exchanged. We denote w at this point as $y^\ddagger$. Since $v^\ddagger$ is the output of Algorithm 3, the square of the distance between the input vector t and $v^\ddagger$ can be presented as

$$\begin{aligned}\left\|t - v^\ddagger\right\|^2 &= \left\|t - B(B^{-1}t - w)\right\|^2 \\ &= \left\|t - B(B^{-1}t - y^\ddagger)\right\|^2 \\ &= \left\|By^\ddagger\right\|^2 \\ &= \left\|B^*\Lambda y^\ddagger\right\|^2\end{aligned}$$

Now we denote $\Lambda y^\ddagger$ by $x^\ddagger$ and ananlyse it's relation to x. The functionality from Step 5 to Step 12 in Algorithm 3 is to check whether $\lambda_{ij} \cdot w_j \geq 0$ for all $j > i$. If so, set $w_j = -w_j$ only when $|l| \geq |r|$ in Algorithm 3 rather than seting $w_j = -w_j$ when $|w_j| = 0.5$ for all j in Algorithm 2; Otherwise, same to Algorithm 2, follow Babai's rounding procedure without any change. Hence,

$$\begin{aligned}x_i^\ddagger &= \sum_{j=1}^{j=n} \lambda_{ij} \cdot y_j^\ddagger = \sum_{j=i}^{j=n} \lambda_{ij} \cdot y_j^\ddagger \leq |\sum_{j=i}^{j=n} \lambda_{ij} \cdot y_j^\ddagger| \\ &\leq \sum_{j=i}^{j=n} |\lambda_{ij} \cdot y_j^\ddagger| = \sum_{j=i}^{j=n} |\lambda_{ij} \cdot y_j| = \sum_{j=1}^{j=n} |\lambda_{ij} \cdot y_j| = x_i.\end{aligned}$$

Therefore,

$$\begin{aligned}\left\|t - v^\ddagger\right\|^2 &= \left\|B^*\Lambda y^\ddagger\right\|^2 = \left\|B^* x^\ddagger\right\|^2 \\ &= |x_1^\ddagger| \left\|b_1^*\right\|^2 + |x_2^\ddagger| \left\|b_2^*\right\|^2 + \cdots + |x_n^\ddagger| \left\|b_n^*\right\|^2 \\ &\leq |x_1| \left\|b_1^*\right\|^2 + |x_2| \left\|b_2^*\right\|^2 + \cdots + |x_n| \left\|b_n^*\right\|^2 \\ &= \left\|B^* x\right\|^2 = \left\|B^*\Lambda y\right\|^2 = \left\|t - v\right\|^2\end{aligned}$$

where the third and fourth equalities follow from the fact that B^* is the Gram-Schmidt orthogonalization of basis B. In particular, if there is at least one coordinate of w appeared in Algorithm 2 whose absolute value is exactly the value 0.5, the step 16 in algrithm 2 will be performed. Thus, there is at least one coordinate $x_i^\ddagger$ which would be strictly smaller than it's corresponding element x_i, i.e. $x_i^\ddagger < x_i$. Then

$$\begin{aligned}\left\|t - v^\ddagger\right\|^2 &= \left\|B^*\Lambda y^\ddagger\right\|^2 = \left\|B^* x^\ddagger\right\|^2 \\ &= |x_1^\ddagger| \left\|b_1^*\right\|^2 + |x_2^\ddagger| \left\|b_2^*\right\|^2 + \cdots + |x_n^\ddagger| \left\|b_n^*\right\|^2 \\ &< |x_1| \left\|b_1^*\right\|^2 + |x_2| \left\|b_2^*\right\|^2 + \cdots + |x_n| \left\|b_n^*\right\|^2 \\ &= \left\|B^* x\right\|^2 = \left\|B^*\Lambda y\right\|^2 = \left\|t - v\right\|^2.\end{aligned}$$

□

4.2 Special Version for HNF

An important fact in lattice theory is that every integer lattice can be represented in its unique Hermite Normal Form basis, and the HNF basis can be efficiently computed from any lattice basis. In this section, we will modify Algorithm 2 and Algorithm 3 according to the characteristics of HNF basis and present their special editions for HNF basis.

The Hermite Normal Form basis of an integer lattice is defined as either row-style or column-style. In this paper, we present the HNF basis as the column-style one which is defined as follows.

Definition 5. *Let L be a m dimentional integer lattice with rank n. A basis $H \in \mathbb{Z}^{m \times n}$ is in Hermite Normal Form if*

- *H is upper triangular, that is, there exists a sequence of integers $j_1 < \cdots < j_l$ such that for all $1 \leq i \leq l$ we have $h_{i,j} = 0$ for all $j < j_i$.*
- *For $1 \leq k < i \leq l$, we have $0 \leq h_{j_i,k} < h_{j_i,i}$, that is, the pivot element is the greatest along its row and the coefficients right are non-negative.*
- *For $i > l$, we have $h_{i,j} = 0$ for all $j = 1, \cdots, m$.*

Lemma 1. *For every integer lattice L, there exists a unique basis H that is in Hermite Normal Form.*

Lemma 2. *For any m dimentional integer lattice with rank n, there exists an polymomial time algorithm to compute it's HNF basis with $O(n^2 log M)$ space complexity and $O(mn^4 log^2 M)$ running time, where M is a bound on its entries.*

Given HNF basis B of an arbitrary integer lattice and a target t, in order to find a lattice point close to a target t we conduct the following steps (Algorithm 4).

1) First compute the Gram-Schmidt orthogonalization B^* of basis B and the corresponding upper-triangular matrix Λ such that $B = B^* \cdot \Lambda$;
2) Apply the inverse transformation B^{-1} to get $B^{-1}t$;
3) Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
4) Compute the vector $w = u - B^{-1}t$;
5) If $w_j \geq 0$ for all $j > i$, goto step 6); Otherwise, goto step 7);
6) For j from 1 to $n-1$, if $|w_j| = 0.5$, set $w_j = -w_j$. Then, goto step 7) ;
7) Map the resulting integer vector to the lattice point $v = B^{-1}t - w$.

Algorithm 4 Special Edition for HNF I

Input:
Lattice HNF basis B, and vector $t \in \mathbb{R}^n$
Output:

```
Compute the GS-basis B* of B and Λ such that B = B*Λ;
Compute B^{-1} and get B^{-1}t;
Round B^{-1}t to the closest integer vector u = ⌊B^{-1}t⌉ ∈ Z^n;
Compute the vector w = u − B^{-1}t;
k = 0;
for i = 1 to n do
    if w_i ≥ 0 then
        k++;
    end if
end for
if k = n then
    for j = 1 to n − 1 do
        if w_j = 0.5 then
            w_j = −w_j;
        end if
    end for
end if
return the resulting integer vector to the lattice point v = B(B^{-1}t + w)
```

Corollary 1. *Given an arbitrary lattice presented in its Hermite normal form(HNF) basis B and a target vector t. Let v and v' be the outputs of Algorithm 1 and Algorithm 4, respectively. Then the distance of t from v' is not greater than that of t from v. In particular, if there is at least one coordinate of w appeared in Algorithm 2 which is exactly the value 0.5, then the distance of t from v' will be strictly smaller than that of t from v.*

Proof. The main differences between Algorithm 2 and Algorithm 4 are several steps which are used to finish some checks. The function of Steps 6 through 12 in Algorithm 2 is to check whether $\lambda_{ij} \cdot w_j \geq 0$ for all $j > i$, while the function of steps 6 through 10 in Algorithm 4 is to check whether $w_i \geq 0$ for all i from 1 to n.

Since the input of Algorithm 4 is a HNF basis H, H is upper triangular and each element is non-negative. Therefore, the Gram-Schimidt orthogonalization can be represented in the following form

$$
\begin{aligned}
H &= \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ 0 & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{nn} \end{bmatrix} \\
&= \begin{bmatrix} h_{11} & 0 & \cdots & 0 \\ 0 & h_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{nn} \end{bmatrix} \cdot \begin{bmatrix} 1 & \lambda_{12} & \cdots & \lambda_{1n} \\ 0 & 1 & \cdots & \lambda_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \\
&= H^* \Lambda
\end{aligned}
$$

where $\lambda_{ij} = \frac{h_{ij}}{h_{ii}}$. Hence, $\lambda_{ij} \geq 0$. Once the input HNF basis passes the check steps 6 through 10 in Algorithm 4, it means that the HNF basis can also pass the check Steps 6 through 12 in Algorithm 2 since λ_{ij} and w_j are both non-negative for all i and j. Therefore, the outputs $v^\dagger$ of Algorithm 2 and v' of Algorithm 4 are actually the same vector if the input of these two algorithms is the same. According to Theorem 3, the conclusion of this corollary holds. □

Corollary 2. *Given an arbitrary lattice presented on its Hermite normal form(HNF) basis B and a target vector t. Let v and v'' be the outputs of Algorithm 1 and Algorithm 5, respectively. Then the distance of t from v'' is not greater than that of t from v. In particular, if there is at least one coordinate of w appeared in Algorithm 2 whose absolute value is exactly the value 0.5, then the distance of t from $v^\ddagger$ will be strictly smaller than that of t from v.*

Proof. This proof is similar to the one above except that it is based on Theorem 4 instead of Theorem 3, omitted here. □

5 Examples and Experiments

In this section, we analyse our proposed algorithms through examples and experiments, and show that these

Algorithm 5 Special Edition for HNF II

Input:
Lattice HNF basis B, and vector $t \in \mathbb{R}^n$
Output:
Compute the GS-basis B^* of B and Λ such that $B = B^*\Lambda$;
Compute B^{-1} and get $B^{-1}t$;
Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
Compute the vector $w = u - B^{-1}t$;
$l = 0; r = 0$
$k = 0$;
for $i = 1$ to n **do**
  if $w_i \geq 0$ **then**
    k++;
  end if
end for
if k=n **then**
  for $j = n - 1$ to 1 **do**
    if $|w_j| = 0.5$ **then**
      $l+ = w_j$;
      $r+ = -w_j$;
      for $i = j + 1$ to n **do**
        $l+ = \lambda_{ji} \cdot w_i$;
        $r+ = \lambda_{ji} \cdot w_i$;
      end for
      if $\mid l \mid \geq \mid r \mid$ **then**
        $w_j = -w_j$;
      end if
    end if
  end for
end if
return the resulting integer vector to the lattice point $v = B(B^{-1}t - w)$

algorithms outperform the classic Babai's algorithm with respect to their corresponding errors.

5.1 Examples

Example 1. Given a 6-rank lattice basis

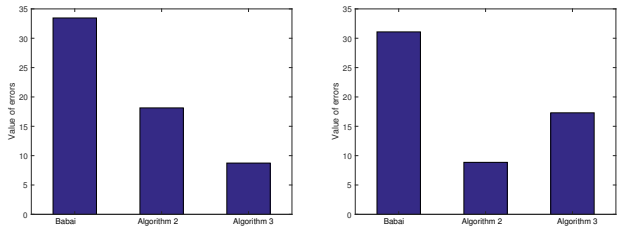
$$B = \begin{bmatrix} 10 & 10 & 2 & 3 & 5 & 6 \\ 9 & 2 & 6 & 5 & 1 & 7 \\ 5 & 0 & 5 & 8 & 10 & 6 \\ 1 & 6 & 6 & 6 & 9 & 3 \\ 5 & 3 & 8 & 2 & 1 & 4 \\ 4 & 9 & 6 & 9 & 1 & 4 \end{bmatrix}$$

and a target vector

$$t = [189.1, 157.6, 133.6, 129, 122.9, 175.6]$$

as input of Algorithms 1, 2, and 3, it is not difficult to obtain the outputs of these three algorithms as

$$v_1 = [205, 170, 148, 143, 133, 190],$$
$$v_2 = [178, 152, 128, 121, 116, 170],$$
$$v_3 = [188, 154, 128, 127, 119, 179],$$



(a) The case of Example 1 (b) The case of Example 2

Figure 1. The error comparison of three algorithms

respectively. The errors between these outputs and the target vector are as follows,

$$error_1 = \|v_1 - t\| = 33.4559,$$
$$error_2 = \|v_2 - t\| = 18.1356,$$
$$error_3 = \|v_3 - t\| = 8.7350,$$

Figure 1(a) shows a comparison of these errors. Both the proposed Algorithm 2 and Algorithm 3 output vectors that are closer to the target vector t than Algorithm 1 (i.e. Babai's algorithm) . In addition, $error_3$ is smaller than $error_2$, which means that Algorithm 3 outperforms Algorithm 2 for this input. In fact, the following experiments show that this phenomenon is common in the vast majority of inputs. However, there are exceptions, such as the input given in example 2.

Example 2. Given a 6-rank lattice basis

$$B = \begin{bmatrix} 3 & 4 & 7 & 4 & 3 & 6 \\ 1 & 4 & 2 & 9 & 1 & 9 \\ 1 & 7 & 7 & 5 & 1 & 8 \\ 6 & 6 & 3 & 6 & 1 & 8 \\ 3 & 2 & 4 & 8 & 7 & 6 \\ 10 & 4 & 4 & 9 & 0 & 2 \end{bmatrix}$$

and a target vector

$$t = [125.7, 110.7, 137.4, 150.8, 120.2, 170.2]$$

as input of Algorithms 1, 2 and 3. It is not difficult to obtain the outputs of these three algorithms as

$$v_1 = [138, 122, 150, 164, 134, 183],$$
$$v_2 = [125, 118, 141, 154, 120, 169],$$
$$v_3 = [132, 120, 148, 157, 124, 173],$$

respectively. The errors between these outputs and the target vector are as follows,

$$error_1 = \|v_1 - t\| = 31.0847,$$
$$error_2 = \|v_2 - t\| = 8.8578,$$
$$error_3 = \|v_3 - t\| = 17.2991,$$

Figure 1(b) shows a comparison of these errors. Both the proposed Algorithm 2 and Algorithm 3 also output vectors that are closer to the target vector t than Algorithm 1

(i.e. Babai's algorithm) . In this case, $error_2$ is smaller than $error_3$, which means that Algorithm 2 outperforms Algorithm 3 for this input.

5.2 Experiments

Our experiments are done on a Windows desktop PC with an Intel Core i5-7200U CPU running at 2.50 GHz. The algorithms are implemented in Matlab R2016b.

For comparison, we randomly generate 500 lattice bases with full rank 8 that can trigger the adjustment strategies in Algorithm 2 and Algorithm 3. Figure 2 shows a comparison of the errors generated by Algorithms 1, 2 and 3 with these randomly chosen lattice bases as inputs. It can be seen easily that our proposed Algorithm 2 and Algorithm 3 both outperform Babai's algorithm with respect to the corresponding errors. Figure 2 also shows Algorithm 3 outperforms Algorithm 2 for the vast majority of inputs. This is in line with our algorithm design expectations. After all, Algorithm 3 has a more accurate adjustment strategy than Algorithm 2.

Figure 3 shows the time overhead of these three algorithms. Our algorithms are slower than Babai's due to the time-consuming Gram-Schimdt procedure. However, this is not a big problem for cryptographic applications, since the implementation of a cryptographic algorithm involves only one selected lattice basis, and attacking the cryptographic algorithm only needs to solve the specific CVP problem of the lattice basis.

In addition to the above general case, we have also done experiments for the special case of HNF. For comparison, we randomly generate 100 lattice bases in HNF with full rank 10. Figure 4 shows a comparison of errors generated by all the five algorithms appeared in the paper. A strange thing is that there are only three curves in Figure 4. This is because Algorithms 4 and 5 are special cases of Algorithm 2 and Algorithm 3 respectively, so the errors they generate are exactly the same.

Although the errors generated are the same, it can be easily seen from Figure 5 that the computational cost of Algorithm 4 and Algorithm 5 is smaller than that of Algorithm 2 and Algorithm 3. In particular, Algorithm 4, in which there is no Gram-Schimdt process, has almost the same computational cost as Babai's algorithm.

6 Conclusions

In this paper, we propose four improved rounding procedures for CVP problem based on Babai's rouding procedure. The first two procedures are general for any type of lattice basis, while the latter two algorithms are special versions of the first two for which the input basis is in HNF form. We also show that all four algorithms perform better than Babai's procedure with respect to approximation factor, although they lose some efficiency as the cost.

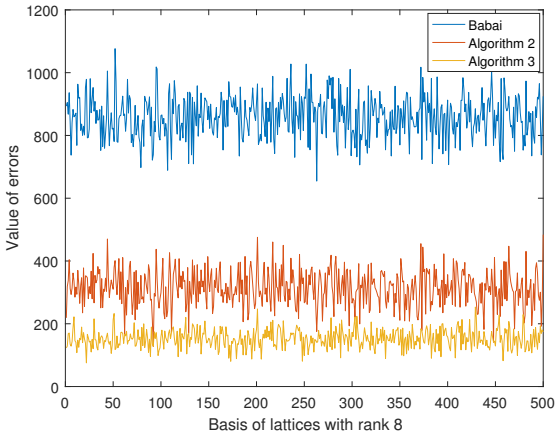


Figure 2. The error comparison of three algorithms

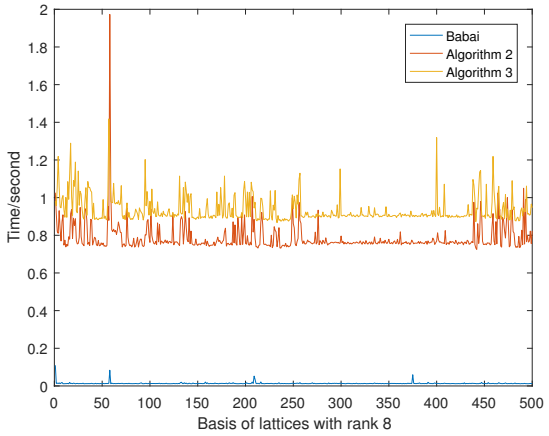


Figure 3. The efficience comparison of three algorithms

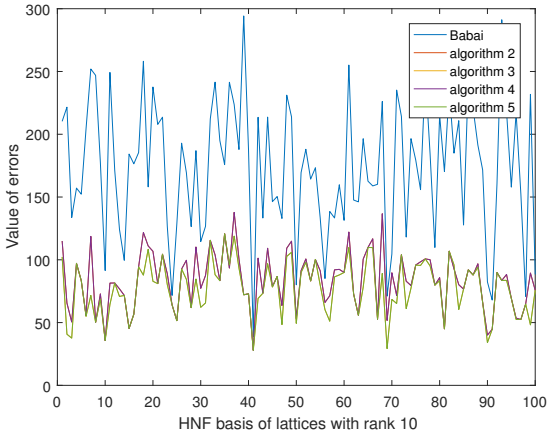


Figure 4. The efficience comparison of five algorithms

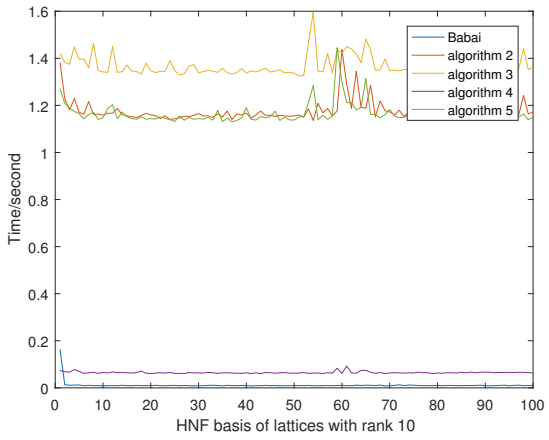


Figure 5. The efficience comparison of five algorithms

Acknowledgments

This work was supported by the Doctoral Fund of University of Jinan (Granted No. XBS1455), and National Science Foundation of Shandong Province (No. ZR2018LF006).

References

[1] D. Aggarwal, H. Bennett, A. Golovnev, and N. Stephens-Davidowitz, "Fine-grained hardness of cvp(p): everything that we can prove (and nothing else)," in *Symposium on Discrete Algorithms*, 2021.

[2] D. Aggarwal and Eldon Chung, "A note on the concrete hardness of the shortest independent vector in lattices," *Information Processing Letters*, vol. 167, no. 106065, pp. 1–5, 2021.

[3] S. Agrawal, E. Kirshanova, D. Stehle, and A. Yadav, "Practical, round-optimal lattice-based blind signatures," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2022.

[4] S. Agrawal, A. Yadav, and S. Yamada, "Multi-input attribute based encryption and predicate encryption," in *Proceedings of Crypto*, 2022.

[5] M. Ajtai, "Generating hard instances of lattice problems (extended abstract)," in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pp. 99–108, ACM, 1996.

[6] M. Ajtai, R. Kumar, and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem," in *STOC*, pp. 601–610, 2001.

[7] L. Babai, "On lovász lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[8] J. Bajard, J. Eynard, N. Merkiche, and T. Plantard, "Babai round-off cvp method in rns: Application to lattice based cryptographic protocols," *International Symposium on Integrated Circuits (ISIC)*, pp. 440–443, 2014.

[9] H. Bennett, C. Peikert, and Y. Tang, "Improved hardness of bdd and svp under gap-(s)eth," in *Proceedings of Innovations in Theoretical Computer Science (ITCS)*, 2022.

[10] V. E. Boas. "Another np-complete problem and the complexity of computing short vectors in a lattice,". Tech. Rep. Technical Report 81-04, University of Amsterdam, 1981.

[11] Z. Brakerski, R. Tsabary, V. Vaikuntanathan, and H. Wee, "Private constrained prfs (and more) from lwe," in *TCC 2017*, pp. 264–302, Baltimore, USA, November 2017.

[12] Z. Brakerski and V. Vaikuntanathan, "Lattice-inspired broadcast encryption and succinct ciphertext policy abe," in *ITCS*, 2022.

[13] R. Challa and G. V. Kumari, "Additively lwe based homomorphic encryption for compact devices with enhanced security," *International Journal of Network Security*, vol. 21, no. 3, pp. 378–383, 2019.

[14] O. Goldreich and S. Goldwasser, "On the limits of nonapproximability of lattice problems," *Journal of Computer and System Sciences*, vol. 60, no. 3, pp. 540–563, 2000.

[15] M. Jiang, Q. Chen, Y. Guo, and D. Zhang, "Multi-bit functional encryption for inner product predicate over lattice," *International Journal of Network Security*, vol. 24, no. 6, pp. 1106–1113, 2022.

[16] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.

[17] A. Mandangan, H. Kamarulhaili, and M. A. Asbullah, "Good basis vs bad basis: On the ability of babai's round-off method for solving the closest vector problem," *Journal of Physics: Conference Series*, vol. 1366, no. 1, p. 012016, 2019.

[18] P. Martins, J. Eynard, J. Bajard, and L. Sousa, "Arithmetical improvement of the round-off for cryptosystems in high-dimensional lattices," *IEEE Trans. on Computers*, vol. 66, no. 12, pp. 2005–2018, 2017.

[19] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective*. Netherlands: Kluwer Academic Publisher, 2002.

[20] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *EUROCRYPT 2012*, pp. 700–718, Cambridge, United Kingdom, April 2012.

[21] S. Zhang, "A lwe-based oblivious transfer protocol from indistinguishability obfuscation," *International Journal of Network Security*, vol. 22, no. 5, pp. 801–808, 2020.

Biography

Shuying Yang received the B.S. and M.S. degree in Mathematics from Shandong Normal University, China, in 2004 and 2007, respectively. Currently, She is an associate professor in department of data and computer

science at Shandong Women's University. Her research interests include information security and cryptology. Prof. Yang may be reached at ysystudy2005@163.com.

Multi-level Program Analysis Method Based on Formal Method

Huaxu Li[1], Weidong Tang[2], and Meiling Liu[2]
(Corresponding author: Weidong Tang)
School of electronic information, Guangxi Minzu University[1]
School of artificial intelligence, Guangxi Minzu University[2]
Nanning, Guangxi 530006, China
Email: china5161@126.com
(Received Sept. 13, 2022; Revised and Accepted Feb. 3, 2023; First Online Feb. 18, 2023)

Abstract

The rapid development of computer technology, parallel computer systems, and various highly concurrent programs are all over every corner of people's daily lives. At the same time, due to the complexity of program design, the system program with high concurrency will also have some unexpected problems. At present, formal methods are widely used in verifying programs and systems, and Petri nets are one of the commonly used methods. However, Petri net system is different from general programs with global control. There is no global control flow in Petri net. Instead, it controls the transition of resources based on the principle of local certainty. Therefore, when analyzing the concurrency problem through Petri net, it is not necessary to establish the global control between processes by exhausting the concurrency of processes, but only to set transition conditions for processes to meet the system's functional requirements. This paper will first introduce the theoretical knowledge of the Petri net system, then illustrate how to use Petri net system to simplify the program model, and finally analyze and summarize the simplification effect of the program model.

Keywords: Concurrent System; Formal Method; Petri Net System; Program Analysis; Structure Reduction

1 Introduction

In the early stage of computer development, the program needs to be guided by the global control flow to execute in sequence, because the main structure of computer program is sequence structure. However, with the development of parallel computer systems [20], the global control flow, which is widely used in sequential systems, is no longer applicable, because parallel systems cannot be controlled in global order like sequential systems. Today's computer programs are no longer composed of simple sequential structures. There are often many synchronous processes in the program, and these processes cooperate with each other after completing their respective work, and finally form a complete program. Therefore, how to analyze programs with more complex subprocesses [3] has become the focus of program analysis researchers. Opara *et al.* [14] analyzed the network system structure for the security of the enterprise network and determined a feasible solution. In terms of security protocols, Liu *et al.* [11] analyzed and calculated the existing authentication protocols, and improved the protocol structure to improve its security.

At present, formal method are widely used in the verification of programs and systems, and Petri nets are one of the commonly used methods. Petri net [7,10] appeared in 1962 and was proposed by German scholar Carl Adam Petri in his doctoral thesis. After more than half a century of development, Petri net has become an important part of the field of information science. Every year, many papers on Petri net are published in various academic journals and conferences.

Gan *et al.* [5] described the propagation relationship of product attribute design changes by constructing an extended Petri net (EPN) model. Rui *et al.* [16] used Petri nets as the basis of service composition modeling, and designed a reliability evaluation algorithm for complex networks. At the same time, Petri nets are also widely used in the research of control fields [9,18,21]. Liu *et al.* [23] proposed a modeling method based on timed colored Petri nets for the lack of attack trees. Taleb Berrouane *et al.* [19] proposed the Bayesian Stochastic Petri Nets(BSPN) and dynamically evaluated security by capturing additional data trend sets.

There is no global control flow and central control in Petri net. Petri net can determine the local conditions of transitions [13] in things through locally determined transition rules and describe the dependencies between things. In Petri net, the definition of the nature of resources depends on the perspective of observing resources. For example, for an electronic product, it is goods for the person who produces the product, and it is daily necessi-

ties for the person who needs to use it. The two purposes are different and the nature is not the same. This qualitative difference is reflected in the relationship between resources and transitions.

It is usually very difficult to analyze a program directly, because the interior of the program usually contains many sequence, cycle and selected substructures. For some concurrent processes [2, 17] in the program, if program researchers want to detect the program by traversing all possible process steps, it will produce explosive state space, because the running order between these processes is not fixed. By using Petri net system [1, 6] to analyze the program structure, we can describe the dependencies between different processes in the program through locally determined transition rules and simplify the program structure, so as to reduce the complexity of program analysis. This paper will first explain the principle of Petri net, then illustrate how to use Petri net to model the program structure and how to reduce the model structure through examples, and finally make a summary.

2 Introduction to the System Structure of Petri Net

2.1 Directed Net

The objects described in Petri net [15] are called resources, and the resources with the same properties are classified into the same class, which is represented by the state element (S). It is specified that the state of the state element (S) is an integer greater than or equal to 0, which is used to represent the number of such resources. Petri nets call the behavior that can change the nature or quantity of state elements transition (T). When describing the net structure graphically, circles are usually used to represent state elements (S), rectangles are used to represent transitions (T), arrows pointing from circles to rectangles are used to represent inputs, and arrows pointing from rectangles to circles are used to represent outputs. We can mark a number on the arrow to indicate the quantity of input or output resources. If no number is marked, the default quantity is one. In this way, we can clearly describe the changing relationship between resources and transitions. This net structure is called directed net. As shown in Figure 1.

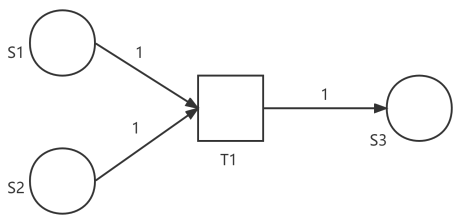


Figure 1: Directed net structure diagram

Definition 1. *Directed net:* $N = (S, T; F)$ *is a directed net, if the following conditions are met:*

1) $S \cup T \neq \emptyset \wedge S \cap T \neq \emptyset$;

2) $F \subseteq S \times T \cup T \times S$;

3) $dom(F) \cup cod(F) = S \cup T$.

In the above, S represents the place, T represents the transition, and F represents the flow relationship from place to transition or transition to place.

2.2 Hierarchy of Petri Net System

Petri net system [] is divided into many different levels, in which each upper structure is constructed from the information extracted from the lower layer. The first layer is the Elementary Net system, which describes the model composed of the activity trajectories of individual objects. The second layer is the Place/Transition system [4, 12]. The difference between the P/T system and the EN system is that the P/T system describes the same kind of individuals together. Therefore, there may be different numbers of individuals of the same kind in the place where the same kind of objects are stored, and such places are called places. The third layer net system is the advanced net system, which takes a step further on the P/T system, and it can describe different types of objects. In the advanced net system, although objects of different classes have different properties, they are in the same state, which is called a predicate. Common advanced net systems include Predicate/Transition system and Colored net system [22].

2.2.1 Elementary Net System

Elementary Net system is the most basic Petri net system model, which describes the transition of individual state, including the combination between individuals and the separation of individuals. In the EN system, each individual has only two different states, and the different states of individuals cannot overlap each other before, that is, individuals have only one state at any time. The state of an individual is also called a condition, and its condition can only be true or not. The Elementary Net system uses token and no token to indicate the true or false of the condition, and there can only be one token in each condition at most, that is, the maximum capacity is 1.

Definition 2. *Elementary Net system:* $\sum = (B, E; F, c_{in})$ *is a Elementary Net, if* $(B, E; F)$ *is a directed net, and* $c_{in} \subseteq 2^B$. *Where* B *is called place, and there are all two states of the place in* B*, namely, with or without token, and the power set of* B *can be represented by* 2^B. *Transitions in* E *are called events, which are only related to conditions.* F *is the same as the definition in directed net, which indicates the flow relationship.* c_{in} *is the initial state of the directed net.*

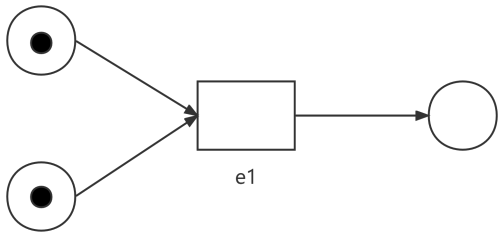


Figure 2: Structure diagram of multi input Elementary Net

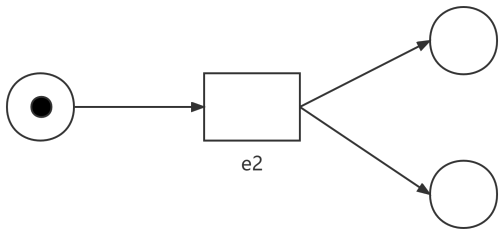


Figure 3: Structure diagram of multi output Elementary Net

In Figure 2 and Figure 3, if there is a black dot in the place, it means there is token; if there is no black dot, it means there is no token. $e1$'s precursor place has token, and then there is no token in the successor place, so the event can occur, and $e2$ is the same.

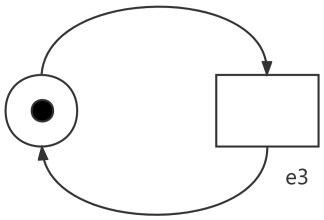


Figure 4: Self ring structure diagram with token

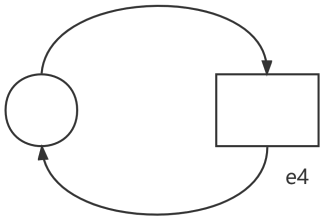


Figure 5: Self ring structure diagram without token

Figure 4 and Figure 5 show two non simple structures, namely, self loop structure, in which the place is both input and output. For $e3$, because there is a token in the precursor and successor of $e3$, this forms a collision and prevents the occurrence of $e3$. For $e4$, because there is no token in the precursor of $e4$, $e4$ cannot occur. This kind of self circulation and non consumption structure is not suitable to be described by the EN system.

Since the EN system describes the individual state and state changes of all objects, there will be many elements in the EN system, which is the disadvantage of the EN system.

2.2.2 Place/Transition System

Because the EN system describes the changing relationship between individuals, it will make the net structure very complicated when it is used to describe the model with many individuals. The Place/Transition system describes the system by grasping the commonness between individuals, and introduces the concept of class in the net structure, stacking the individuals of the same kind in the system, so that it greatly reduces the complexity of the system.

The P/T system is different from the EN system. There may be more than one similar resource in the place in the P/T system. Therefore, the place capacity and weight function are added to the formal definition of the P/T system to specify the number of resources that the place can load and the number of resources that will be consumed or generated in the transition.

Definition 3. *P/T system:* $\sum = (S, T; F, K, W, M_0)$ *is a Place/Transition system, if* $(S, T; F)$ *is a directed net, and the following conditions are met:*

1) $K : S \to \{1, 2, 3...\} \cup \{\infty\}$;

2) $W : F \to \{1, 2, 3...\}$;

3) $M_0 : S \to \{0, 1, 2...\}$;

4) $\forall s \in S, M_0(s) \leq K(s)$.

Where K is the capacity of the place, and the range is $1 \sim \infty$. W represents the number of resources to be consumed or generated by the transition, and the number is limited, which is called weight. M_0 represents the initial state of each S element, and the number of initial elements of each place cannot be greater than the maximum capacity of the place.

2.2.3 Advanced Net System

The advanced net system integrates different types of individuals with the same transition law on the basis of the P/T system, and the advanced net system also belongs to the linear net system. Common advanced net system [24], [8] include Predicate/Transition system and Colored net system. Predicate/Transition system is suitable for predicate logic, while Colored net system is often

used to describe physical system. This paper will use the Colored net system as the basis for model simplification, therefore, the following article will focus on the Colored net system in the advanced net system.

Colored net system distinguishes resources by giving different colors to each type of token, so that different types of token can transition in the same net structure. The P/T system merges resources of the same kind, but for some resources, although they belong to different resource places, they may also have the same transition law with each other. By using different colors, it is possible to combine and stack these different types of resources with the same transition path, so as to further reduce the number of places and transitions in the net system.

Definition 4. *Colored net system:* $\sum = (P, T; F, C, I_-, I_+, M_0)$ *is a Colored net system, if* $(P, T; F)$ *is a directed net, and the following conditions are met:*

1) $C : P \cup T \to \psi(D)$, *where* $\psi(D)$ *is the power set of color set* D. *For* $\forall p \in P$, $C(p)$ *is the token color set of* p, *and for* $\forall t \in T$, $C(t)$ *is the generating color set of* t.

2) I_- *and* I_+ *are negative and positive functions on* $P \times T$ *respectively, for* $\forall p, t \in P \times T$:

$$I_-(p,t) \in [C(t)_{MS} \to C(p)_{MS}]_L$$

$$I_+(p,t) \in [C(t)_{MS} \to C(p)_{MS}]_L$$

Where the necessary and sufficient condition of $I_-(p,t) = 0$ *is* $(p,t) \notin F$, *and the necessary and sufficient condition of* $I_+(p,t) = 0$ *is* $(t,p) \notin F$. *The function from a non empty set* S *to a non negative integer is called the multiset of* S, *and* S_{MS} *is the set of all finite multisets on set* S. $[A \to B]_L$ *represents the set of all linear functions from set* A *to set* B.

3) $M_0 : P \to D_{MS}$, M_0 *is the initial mark, for* $\forall p \in P$, $M_0(p) \in C(p)_{MS}$.

3 program Structure Simplification Based on Petri Net System

3.1 Model Construction of Elementary Net System

There are often many processes competing for the same resources in the program, and because of the concurrency between different processes, it is very difficult to directly analyze the structure of the program. Reisig [15] pointed out that Petri nets can well analyze the regulated flows of objects and information in the system. At present, Petri nets have been widely used in modeling hardware, communication protocols, and parallel programs.

Using Petri net system to build the net structure of the program can help people analyze the flow of each resource in the program, thus reducing the workload of program analysts. The following will illustrate how to use the EN system to construct the Petri net model of the program.

Example: Suppose a program is composed of three concurrent processes, which compete for a common resource in the system in a cycle, and it is required that the next cycle can be carried out only after all three processes are completed. The program structure is as follows:

```
Program{
    while(true){
        Process1 || Process2 || Process3;
        wait Process1&Process2&Process3 done;
        continue;
    }
}
Process1{
    use Resource;
    release Resource;
}
Process2{
    use Resource;
    release Resource;
}
Process3{
    use Resource;
    release Resource;
}
Resource{
    be used;
    reset;
}
```

First, taking Process1 as an example, the EN system is used to model the process. The EN system is to analyze the individual level, and it can only have one token at most in each place. Here, the different states of Process1 can be regarded as places, and each statement in Process1 can be represented by different transitions, and token is used to represent the occurrence right of the running state of Process1. The structure of the statements in the Process1 corresponding to the transition and place is as follows:

```
Process1{
    //place1
    use Resource; //transition1
    //place2
    release Resource; //trasition2
    //palce3
}
```

Figure 6 shows the net structure corresponding to Process1. Because Process1 needs to interact with external resources, there are two arrows(inward&outward) marked in the figure.

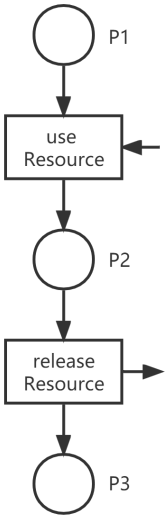


Figure 6: Net structure of Process1

For the Resource, when Resource is used by a Process until it is released by the Process, it is in the used state. Therefore, the program statement(be used;) in the Resource correspond to the two sub statements in the Process. Since the Resource will be reset after being used by the Process, there will be no new place after the program statement(reset;), but it will return to the first place. The structure of transition and place in Resource is as follows:

```
Resource{
    //place1
    be used; //trasition1
    //place2
    reset; //trasition2;
}
```

Figure 7 describes the net structure of the Resource, in which the additional two arrows are used to represent the external input and output.

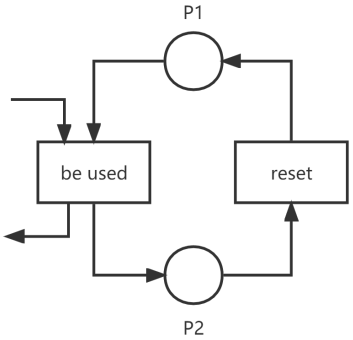


Figure 7: Net structure of Resource

For the Program, we only need to analyze places and transitions in the loop. Similarly, there is no new place after the statement(continue;), but back to the first location of the loop structure. The corresponding structure is as follows:

```
Program{
    while(true){
        //place1
        Process1 || Process2 || Process3;
            //trasition1
        //place2
        wait Process1&Process2&Process3 done;
            //trasition2
        //place3
        continue; //trasition3
    }
}
```

Figure 8 shows the net structure corresponding to the Program.

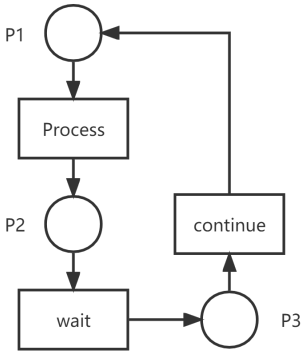


Figure 8: Net structure of Program

The EN system model of the whole program can be obtained by integrating the above three parts, as shown in Figure 9. In Figure 9, the black dots in the P1, P2, P3 and Resource indicate that there is a token in the initial state of the place, and the symbols on each line indicate the resources required for the transition or the resources generated by the transition. There are four resources in total, namely p1, p2, p3 and r, which represent the token corresponding to the three processes and resources respectively. It should be noted that for transition, transition can occur only when its predecessor and successor can be satisfied.

3.2 Using the P/T System to Simplify the Model

The previous section explained how to build the EN system model of the program, but the EN system model only describes the trasition of individuals, so the net structure is very complex. Next, we will further simplify the EN system model by merging similar resources through the P/T system.

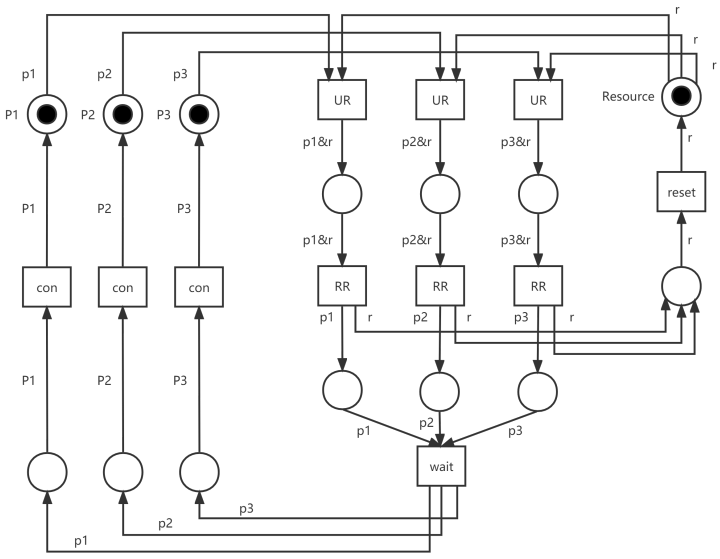


Figure 9: EN system model of program

In the EN system model, Process1, Process2, and Process3 belong to the same place, and their structures are also identical. Therefore, the net structures of these three individuals can be combined, so that the number of place and transition nodes in the net structure can be reduced. In the P/T system, similar resources have the same net structure, so when building the P/T system model, we only need to consider the net structure of one process, which can represent all similar processes. At the same time, we should place three black dots in the original initial place to indicate that there are three similar resources. The P/T system model is in Figure 10.

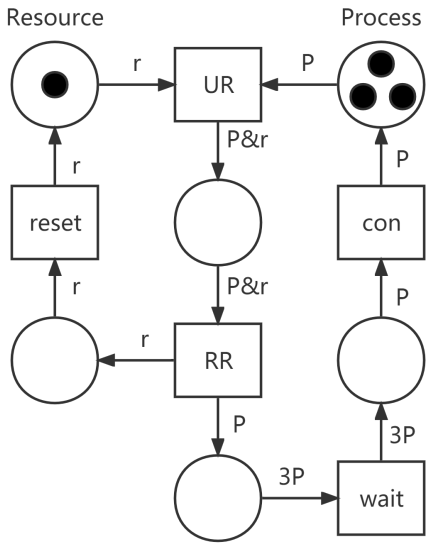


Figure 10: P/T system model of program

The P/T structure in Figure 10 is significantly simplified compared with the EN net structure in Figure 9, because the merging of similar resources greatly reduces the number of nodes in the net structure, while retaining the nature of the original program structure.

3.3 Using Colored Net System to Optimize the Model

After simplifying the EN system model by using the P/T system, multiple transition paths of similar resources in the original net model are combined into a single path, which carries three process resources. The P/T system divides different types of resources, and in some cases, even different types of resources have some similar functions and structures. For example, transition *reset* and transition *con* in Figure 10 have similar functions and structures. Their functions are to complete this loop and return to the beginning of the next loop. Their differences lie in the different types of resources and the number of resources.

The Colored net system distinguishes different types of resources by color, which enables different types of resources to use the same path in the net structure. Yu *et al.* [22] analyzed and modeled multi resources and multi activities by using colored Petri nets, and pointed out that it was easier to observe the overall system resource allocation process and understand the resource allocation principles. By optimizing the P/T net structure in Figure 10 through the Colored net system, the net structure shown in Figure 11 can be obtained.

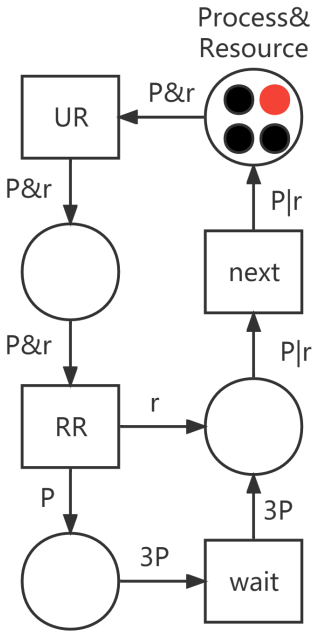


Figure 11: Colored net system model of Program

The initial state in Figure 11 has four token, with three black dots representing the three processes of the program, and the red dots representing the resources required by the process. Compared with the P/T net structure, the Colored net structure incorporates resources into part of the program structure, and renames the original transition *con* to transition *next* to indicate that it has new significance in the Colored net structure. It can be seen from the net structure in Figure 11 that by using Petri net system to simplify the program, the complexity of the original program can be greatly reduced, so as to facilitate people to analyze and study the program structure.

4 State Space Analysis of Different Net Structures

After constructing the Petri net system of different levels in the previous program, three different net structures are obtained. Next, the three net structures will be modeled by CPN tools and their state space will be analyzed. Since CPN tools are mainly used to build Colored net models, the following programming will use the color of only one single resource to represent the individuals in the EN system, and the construction method of the P/T net system is the same.

The code and net structure model of the EN system are as follows:

```
colset Pro1 = with pro1;
colset Pro2 = with pro2;
colset Pro3 = with pro3;
colset Resource = with res1;
var res : Resource;
colset prod1 = product Pro1*Resource;
colset prod2 = product Pro2*Resource;
colset prod3 = product Pro3*Resource;
```

The code and net structure model of the P/T system are as follows:

```
colset Resource= with res1;
var res: Resource;
colset Process = with pro1 | pro2 | pro3;
var pro: Process;
colset prod = product Process*Resource;
```

The code and net structure model of Colored net system are as follows:

```
colset Work = with pro | res;
var work: Work;
```

The state space and integer bounds corresponding to the net system structure models in Figure 12, Figure 13 and Figure 14 are shown.

Table 1: State space of net system structures

State space		
EN system	P/T system	Colored net system
Nodes: 81	Nodes: 81	Nodes: 26
Arcs: 155	Arcs: 155	Arcs: 39
Secs: 0	Secs: 0	Secs: 0
Status: Full	Status: Full	Status: Full

Table 2: Integer bounds of net system structures

Integer bounds(Upper&Lower)		
EN system	P/T system	Colored net system
P1: 1,0	P1: 1,0	P1: 4,0
P2: 1,0	P2: 3,0	P1: 2,0
P3: 1,0	P3: 1,0	P1: 4,0
P4: 1,0	P4: 3,0	P1: 3,0
P5: 1,0	P5: 3,0	
P6: 1,0	P6: 1,0	
P7: 1,0		
P8: 1,0		
P9: 1,0		
P10: 1,0		
P11: 1,0		
P12: 1,0		
P13: 1,0		
P14: 1,0		

It can be seen from Table 1 and Table 2 that the simplified net system structure of the P/T system is no different from the original EN system structure in the state space. However, the bounds number of P/T network system has decreased significantly. Because the P/T network system combines similar resources, and the number of nodes in the network structure model is reduced. But the P/T network system does not change the complexity of the overall state space, but simplifies the model structure.

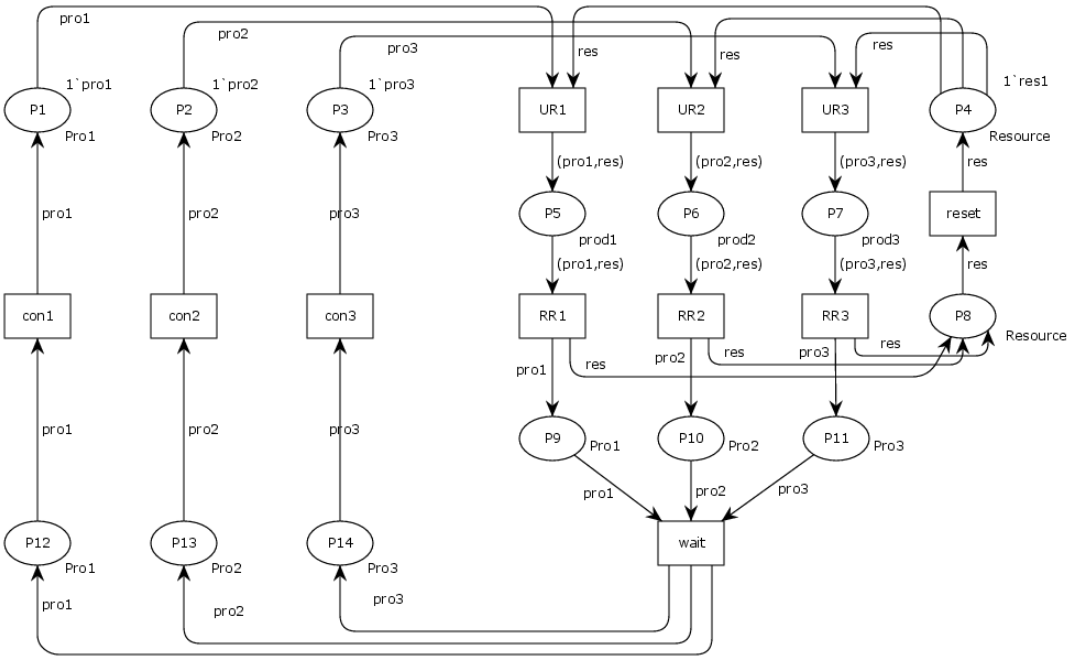


Figure 12: EN system structure model

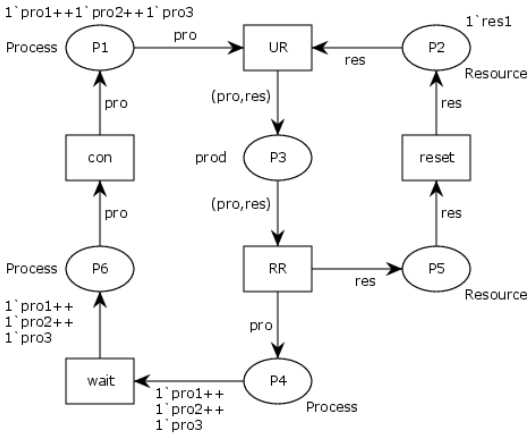


Figure 13: P/T system model of Program

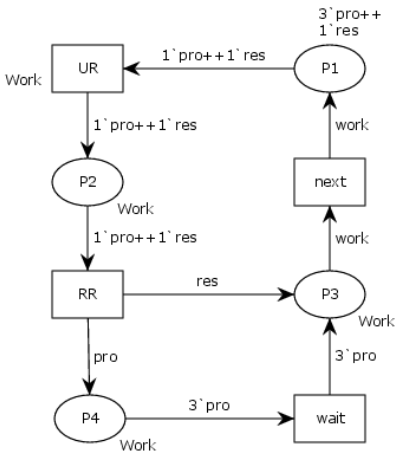


Figure 14: Colored net system model of Program

The structural optimization of Colored net system is a step further. This optimization method combines different types of resources with the same function and structure. Thus Colored net system can reduce the transition path in the net structure, and reduce the state space of the entire model.

5 Conclusions

This paper takes how to simplify the structure of program as the starting point. Firstly, by introducing the relevant knowledge of Petri net system, the corresponding net structure of the program is modeled, and then the net structure of the program is simplified layer by layer according to the different levels of Petri net system theory. Finally, the simplification of simplified program structure is realized. In order to apply Petri net theory to program simplification, this paper explains how to create places and transitions according to program statements through examples, takes the occurrence right of program statements as a token, and models programs based on different net system levels through Petri net tools. Finally, the net structures of different levels are analyzed.

Although this paper proposes how to establish transition and place according to program statements, it is not easy to model more complex software system structures. Therefore, in the next work, we will consider how to build the corresponding Petri net model for large-scale software programs with a large number of subroutines and complex structures, study how to design an algorithm that can build the net system model, and then expand the application of Petri Net Theory in computer program analysis

and detection.

Acknowledgments

This research was partly supported by the National Natural Science Foundation of China under Grant No. 62062011, Guangxi Natural Science Foundation under Grant No. 2017GXNSFAA198008 and Open Fund Grant No. GXIC20-06 of Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis.

References

[1] E. Best, R. Devillers, and M. Koutny, *Petri Net Algebra*, Springer Science & Business Media, 2013.

[2] J. Bicevskis, G. Karnitis, Z. Bicevska, and I. Oditis, "Analysis of concurrent processes in internet of things solutions," in *Special Sessions in the Advances in Information Systems and Technologies Track of the Conference on Computer Science and Intelligence Systems, Conference on Information Systems Management*, Springer, pp. 26–41, 2022.

[3] R. Boutonnet and N. Halbwachs, "Improving the results of program analysis by abstract interpretation beyond the decreasing sequence," *Formal Methods in System Design*, vol. 53, no. 3, pp. 384–406, 2018.

[4] R. Bruni, H. Melgratti, and U. Montanari, "A connector algebra for p/t nets interactions," in *International Conference on Concurrency Theory*. Springer, pp. 312–326, 2011.

[5] Y. Gan, Y. He, L. Gao, and W. He, "Propagation path optimization of product attribute design changes based on petri net fusion ant colony algorithm," *Expert Systems with Applications*, vol. 173, p. 114664, 2021.

[6] A. Giua and M. Silva, "Petri nets and automatic control: A historical perspective," *Annual Reviews in Control*, vol. 45, pp. 223–239, 2018.

[7] K. Jensen and G. Rozenberg, *High-level Petri nets: theory and application*, Springer Science & Business Media, 2012.

[8] S. Kabir and Y. Papadopoulos, "Applications of bayesian networks and petri nets in safety, reliability, and risk assessments: A review," *Safety science*, vol. 115, pp. 154–175, 2019.

[9] H. Kaid, A. Al-Ahmari, Z. Li, and R. Davidrajuh, "Single controller-based colored petri nets for deadlock control in automated manufacturing systems," *Processes*, vol. 8, no. 1, p. 21, 2019.

[10] I. Koch, W. Reisig, and F. Schreiber, *Modeling in systems biology: the Petri net approach*, Springer science & business media, vol. 16, 2010.

[11] W.-R. Liu, X. He, and Z.-Y. Ji, "An improved authentication protocol for telecare medical information system," *International Journal of Electronics and Information Engineering*, vol. 12, no. 4, pp. 170–181, 2020.

[12] Z. Ma, X. Yin, and Z. Li, "Marking diagnosability verification in labeled petri nets," *Automatica*, vol. 131, p. 109713, 2021.

[13] B. Muminov and E. Kh, "Modelling asynchronous parallel process with petri net," *International Journal of Engineering and Advanced Technology*, vol. 8, pp. 400–405, 2019.

[14] E. U. Opara and O. J. Dieli, "Enterprise cyber security challenges to medium and large firms: An analysis," *International Journal of Electronics and Information Engineering*, vol. 13, no. 2, pp. 77–85, 2021.

[15] W. Reisig, *A primer in Petri net design*, Springer Science & Business Media, 2012.

[16] L. Rui, X. Chen, X. Wang, Z. Gao, X. Qiu, and S. Wang, "Multiservice reliability evaluation algorithm considering network congestion and regional failure based on petri net," *IEEE Transactions on Services Computing*, 2019.

[17] A. Schumann, "Towards context-based concurrent formal theories," *Parallel Processing Letters*, vol. 25, no. 01, p. 1540008, 2015.

[18] A. Shahidinejad, M. Ghobaei-Arani, and L. Esmaeili, "An elastic controller using colored petri nets in cloud computing environment," *Cluster Computing*, vol. 23, no. 2, pp. 1045–1071, 2020.

[19] M. Taleb-Berrouane, F. Khan, and P. Amyotte, "Bayesian stochastic petri nets (bspn)-a new modelling tool for dynamic safety and reliability analysis," *Reliability Engineering & System Safety*, vol. 193, p. 106587, 2020.

[20] X. Wang, L. Feng, and H. Zhao, "Fast image encryption algorithm based on parallel computing system," *Information Sciences*, vol. 486, pp. 340–358, 2019.

[21] X. Wu, S. Tian, and L. Zhang, "The internet of things enabled shop floor scheduling and process control method based on petri nets," *IEEE Access*, vol. 7, pp. 27 432–27 442, 2019.

[22] W. Yu, M. Jia, X. Fang, Y. Lu, and J. Xu, "Modeling and analysis of medical resource allocation based on timed colored petri net," *Future Generation Computer Systems*, vol. 111, pp. 368–374, 2020.

[23] J. Zhou, G. Reniers, and L. Zhang, "Petri-net based attack time analysis in the context of chemical process security," *Computers & Chemical Engineering*, vol. 130, p. 106546, 2019.

[24] M. Zhou and N. Wu, *System modeling and control with resource-oriented Petri nets*, Crc Press, 2018.

Biography

Huaxu Li, born in 1997. Master. His main research interests include formal methods and model checking.

WeidongTang, born in 1968. phD, associate professor. His main research interests include formal methods, symbolic computing and rough data reasoning.

Meiling Liu, born in 1979. Associate professor. Her main research interests include Data Mining, formal verification.

An Illegal Image Classification System Based on Deep Residual Network and Convolutional Block Attention Module

Zengyu Cai[1], Xinhua Hu[2], Zhi Geng[1], Jianwei Zhang[2], and Yuan Feng[1]
(Corresponding author: Yuan Feng)
College of Computer and Communication Engineering, Zhengzhou University of Light Industry[1]
Zhengzhou, Henan 450000,China
Email: mailfengy@163.com
College of Software Engineering, Zhengzhou University of Light Industry[2]
Zhengzhou, Henan 450000, China
(Received Sept. 28, 2022; Revised and Accepted Feb. 3, 2023; First Online Feb. 18, 2023)

Abstract

As the process of the Internet keeps advancing, various kinds of lousy information appear on the network, and more and more illegal images start to show in front of people. Illegal images pollute the network environment and affect people's physical and mental health. Hence, we design and implement an illegal image classification system based on a deep residual network and Convolutional Block Attention Module (CBAM) to solve such problems. In this paper, ResNet101 is used as the backbone network to build the model, and CBAM is embedded in the residual neural network. During the training process of the network, the training parameters and optimization algorithm are continuously adjusted to improve the accuracy of the system's recognition. The network's excellent performance is verified by comparing different residual structures and attention mechanisms. The experimental results show that the model achieves 93.2% classification accuracy and can effectively classify illegal images, which helps to create an exemplary network environment.

Keywords: CBAM; Deep Residual Network; Illegal Image

1 Introduction

Due to the rapid development of the Internet and its open and free characteristics, illegal information began to appear on the Internet and gradually spread. These illegal images greatly affect people's online experience. Pornographic and bloody images are the most common carriers of illegal content on the Internet. Pornographic images are images that often contain nude sensitive areas of the body or sexual acts. Its manifestations are relatively rich, and some pornographic pictures also have a certain degree of concealment, such as pictures with strong sexual hints [6].

Obsessed with online pornography can lead people to neglect their studies or jobs. Bloody pictures tend to leave psychological shadows on teenagers and damage their physical and mental health. It may even distort their psychology and make them commit crimes [1]. Detection of illegal images has always been a topic of concern in the field of network information security [4]. In order to remove illegal images on the network, we develop an illegal image classification system to filter out illegal images and eliminate them, which is of great significance for purifying the network environment.

With the development of artificial intelligence technology, more and more researchers have applied machine learning to illegal image recognition, and achieved excellent results. However, faced with the explosive growth of image data, the limitations of traditional machine learning algorithms have been unable to meet the learning ability of illegal image recognition. The deep learning method mines the potential rules of data and uses multi-layer neural network to adapt to high-dimensional learning [3], which shines brightly in the field of image recognition.

This paper implements an illegal picture classification system based on deep residual neural network and CBAM. The system can identify and divide the pictures in the folder into three categories: pornographic, bloody and normal through deep learning algorithm. The classifier of this system uses a model based on deep residual neural network. In order to make the neural network pay more attention to the illegal information objects and reduce the influence of uncertainty in the classification task, we introduce the visual attention mechanism into the model. FSinally, the classification accuracy of the model is up to 93.2%, which is 3.8% higher than that of the network without attention mechanism.

2 Related Work

2.1 Machine Learning-based Illegal Image Recognition

There are two main categories of traditional machine learning methods for illegal image classification. The first category is the method of modeling based on skin color and texture features. Yin et al. established a triple filtering model of color filtering, texture filtering and geometric filtering for skin area detection in pornographic image recognition [21]. This method is not completely effective. For example, face images have many skin pixels, but It's not a porn image. Balamurali and other researchers have proposed a two-stage multi-parameter statistical algorithm to identify adult images [2]. The first stage is to convert the RGB image into a YCbCr image and count the skin pixel values from it. The second stage is to use the Viola-Jones algorithm to detect the face data in the image. Finally, they used the data from the first two stages to classify pornographic and non- pornographic images. However, this kind of method relies too much on the threshold of skin pixels, and the recognition performance of the model is not good under the influence of factors such as illumination [6]. The second category is pattern recognition methods based on handcrafted features. Dong et al. proposed a model based on Bag-of-Visual-Words (BoVW), which combined text information such as file names, file headers, and web page content with image features such as textures and shapes, and used support vector machines to classify them [8]. These methods focus more on the selection and extraction of image features, and the models built by relying on skin alone are not very applicable and reliable in many scenarios [4].

2.2 Illegal Image Recognition Based on Deep Learning

Deep learning is widely used in the field of image recognition, and a large number of researchers have applied it in the field of illegal image recognition. Chen [5] compared the global view feature bloody picture method of support vector machine and the regional bloody detection method of convolutional neural network, and proved that the classification of convolutional neural network is better [16]. Ou [17] designed a deep multi-context network (DMCNet), which fuses the results of global classification and local detection, and makes the final decision through weighted voting. Wang proposed an illegal image recognition algorithm with coarse classification and multiple networks [18]. The algorithm idea is to input images of different target sizes into different neural network models in advance, so that different networks focus on image feature extraction and learning of the same target size, which improved the accuracy of recognition. Gupta [10] used a proposed Aquila Coyote (AqCO) optimization algorithm and AqCO-based deep convolutional neural network for feature extraction from the features collected from input images to collect important features to assist effective classification. The method has high classification accuracy. J. CHEN proposed a method of image compression and reconstruction to reduce the distortion caused by image compression. The authors of [8, 14] use a deep convolutional neural network based on attention mechanism to recognize pornographic images.

2.3 Residual Neural Network

The convolutional neural network (CNN) extracts the features of the image through the strategies of weight sharing, local receptor field and spatial sampling, which not only reduces the calculation parameters, but also retains the spatial features of the image [20]. Deep CNNs reduce the possibility of overfitting. However, in the training process of the deep CNN, there will be a gradient instability problem, resulting in weight degradation. And as the number of layers deepens, this phenomenon becomes more and more obvious. In order to solve the problem of weight degradation in the training process of CNN, some layers of the neural network can artificially skip the connection of neurons in the next layer, which is called Residual Network (ResNet). In ResNet, a unit that is directly connected across layers is called a residual block. The specific construction details of the residual block are shown in Figure 1(a). If the input of the stacked layer is expressed as x, and the expected mapping output is expressed as $H(x)$ then the learned residual can be $F(x) = H(x) - x$, so the original learned feature is $F(x)+x$. The reason for this is that residual learning is easier than direct learning from raw features. If the residual is 0, then the stacking layer only does the identity mapping, at least the network performance will not be degraded [11].

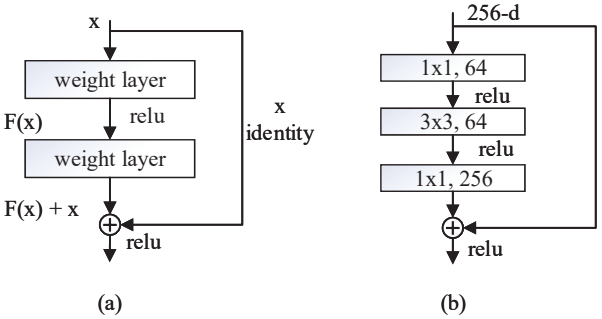


Figure 1: Residual block

In deeper networks, the residual blocks can use bottleneck design in order to make the training time not too long. The three-layer convolution as shown in the Figure 1(b): the 1×1 convolution is to reduce or increase the dimensionality, making the 3×3 layer a bottleneck with smaller input/output. The ResNet101 used in this article uses a residual block with bottleneck design [11].

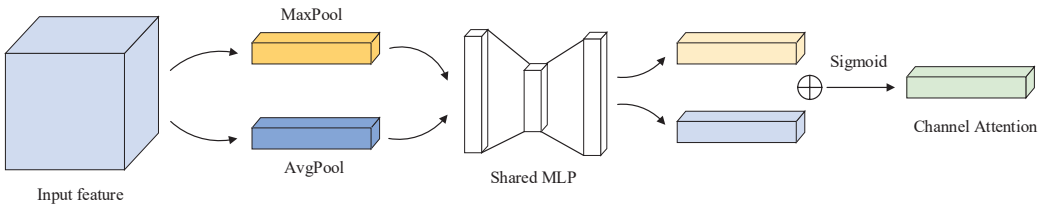


Figure 2: Structure of channel attention module

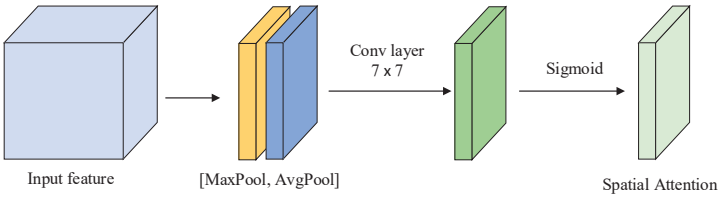


Figure 3: Structure of spatial attention module

2.4 Attention Mechanism

The essence of attention mechanism is to locate interesting information from numerous information and suppress useless information. In the illegal image classification task, the attention mechanism increases the weight ratio of key information by finding key regions, and improves the classification accuracy of the model.

2.4.1 Channel Attention Module

The Channel Attention Module (CAM) corrects the feature representation ability of different channels by extracting the importance of different channels to key information in the feature map. Its structure is shown in Figure 2. First, the spatial dimension of the input feature map is compressed by average pooling and max pooling to generate two spatial context descriptions. Then input the two descriptors into the shared network to get the attention weight matrix. Channel attention focuses on "what" is meaningful given an input image [19].

2.4.2 Spatial Attention Module

The role of the Spatial Attention Module (SAM) is to find the key information of the spatial location and improve the weight of the key spatial regions. Its structure is shown in Figure 3. First, the channel dimension of the input feature map is compressed by average pooling and max pooling, and the generated feature descriptors are connected. A 7×7 convolution kernel is then used to generate the spatial attention weight matrix. The "where" that spatial attention focuses on is an informative part. It is complementary to channel attention [19].

3 Design of Illegal Image Classification System

3.1 System Function Design

This system mainly has three functions: data import, illegal content detection, classification results show.

1) Data import. Users can directly select the pictures they want to detect through the folder selection function, which simplifies the tedious process for users to input folder paths.

2) Illegal content detection. After receiving the file path selected by the user, the system will find the image to be classified, and then input the preprocessed image into the illegal image classifier. The deep residual network calculates the input image.

3) Classification result display. During the detection process of the deep residual network, the system will display the classification results of each picture one by one.

4) File classification. After the deep residual network detects all the pictures in the folder, the system will generate different categories of folders under the folder, and transfer the pictures to the corresponding category folder.

3.2 System Process Design

As shown in Figure 4. After the user enters the main interface, he selects the image folder to be classified through a module for selecting folders. Then the system will crop and tensor the images in the folder, and then input the preprocessed data into the illegal image classifier. The

classifier obtains the classification result through the calculation of the deep residual network and displays it on the system interface. After the detection, the system divides the pictures into different categories of folders according to the classification results, so as to realize the classification of illegal pictures.

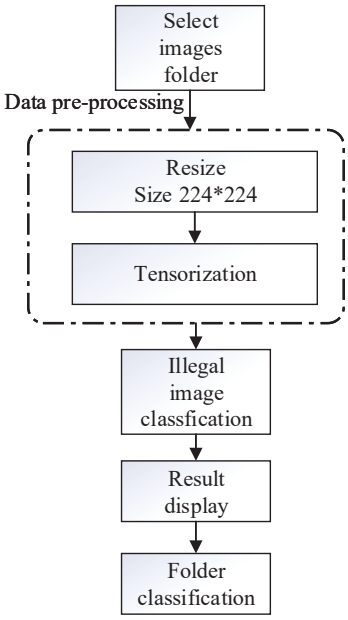


Figure 4: System flow

3.3 Design of Illegal Image Classification Model

Deep convolutional neural networks have brought a series of breakthroughs in image classification [11]. As the network deepens, the features extracted by the convolutional neural network become more detailed, so that the performance of the network gets better and better. However, deep networks are prone to problems such as vanishing gradients or exploding gradients. ResNet can effectively avoid this situation by a shortcut connection. Therefore, we use a deep ResNet for illegal image classification and insert a convolutional block attention module in the middle of the network. This module has two sequential submodules: CAM and SAM. The structure diagram is shown in Figure 5. The input feature map first passes through the CAM and then through the SAM [19]. CBAM is a lightweight general module that can be seamlessly embedded into ResNet and can be trained with the network.

The deep residual network classifier in this paper uses ResNet101 as the backbone network, and two CBAMs are embedded in the network. As shown in the network structure diagram shown in Figure 6, the initial layer is an ordinary convolution structure and a maximum pooling layer, and CBAM is embedded in the middle. Then there are four layers in the residual neural network, layer1 contains three residual blocks, layer2 contains four residual blocks, layer3 contains 23 residual blocks, and layer4 contains three residual blocks. Finally, there is an average pooling layer and a fully connected layer, and CBAM is embedded before the average pooling layer.

4 Experiment

4.1 Data Preprocessing

The data used in this paper for pornographic image detection comes from the public dataset project published on GitHub by data scientist Alexander Kim. We selected 1300 pornographic images and 900 normal images from this project. In addition, about 800 bloody pictures were downloaded from the Internet for bloody detection. In deep learning, the purpose of expanding the dataset can be achieved through data augmentation. In this paper, operations such as changing brightness, chroma, contrast, sharpness and mirror transformation are performed on the image dataset. On the premise of ensuring image features, the number of image datasets is increased. Finally, there are 11,476 pornographic pictures, 8,415 bloody pictures, and 9,290 normal pictures. The proportion of categories in the dataset is shown in Figure 7.

The image itself must be the same size as the model input to be learned by the neural network. When loading the dataset, we first crop the image into a 224×224×3 input feature map and convert it to a tensor type. Then split the data set, the training set accounts for 80%, and the test set accounts for 20%. Shuffle the order of the pictures and load them into the model in batches for training.

4.2 Model Training

In the process of data training, there are gradient explosion and gradient disappearance problems. Using residual struc-ture and Rectified Linear Unit (Relu) can reduce the influ-ence of gradient explosion or disappearance [9]. In addition, the network parameters of the convolutional layer are initialized to satisfy the normal distribution when building the model. Weight initialization makes the gradient of forward or backward propagation more stable during the training pro-cess of the network. A reasonable variance not only guaran-tees a certain difference in the values, but also ensures a cer-tain stability of the values. That is, through the reasonable initialization of the convolution weights, the numerical dis-tribution in the calculation process is stabilized.

The main idea of the Momentum gradient descent algorithm is to apply the method of moving exponentially weighted average:

$$v_{dw} = \beta_1 v_{d\omega} + (1 - \beta_1)\, dw \tag{1}$$

$$v_{db} = \beta_1 v_{db} + (1 - \beta_1)\, db \tag{2}$$

The parameters v_{dw} and v_{db} are the gradient momentum of the loss function during the first $t-1$ iterations, and β_1

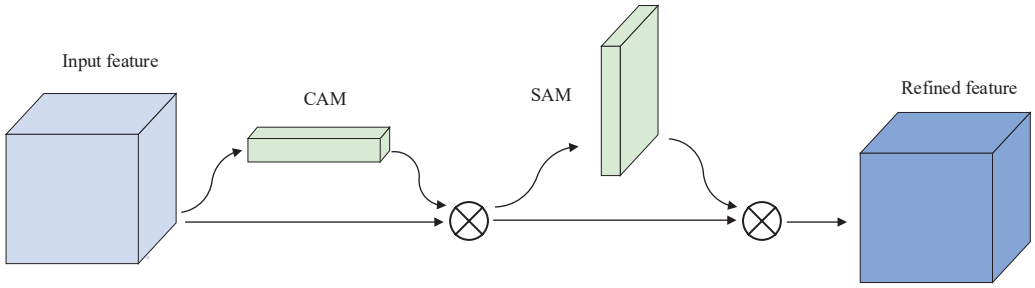


Figure 5: CBAM

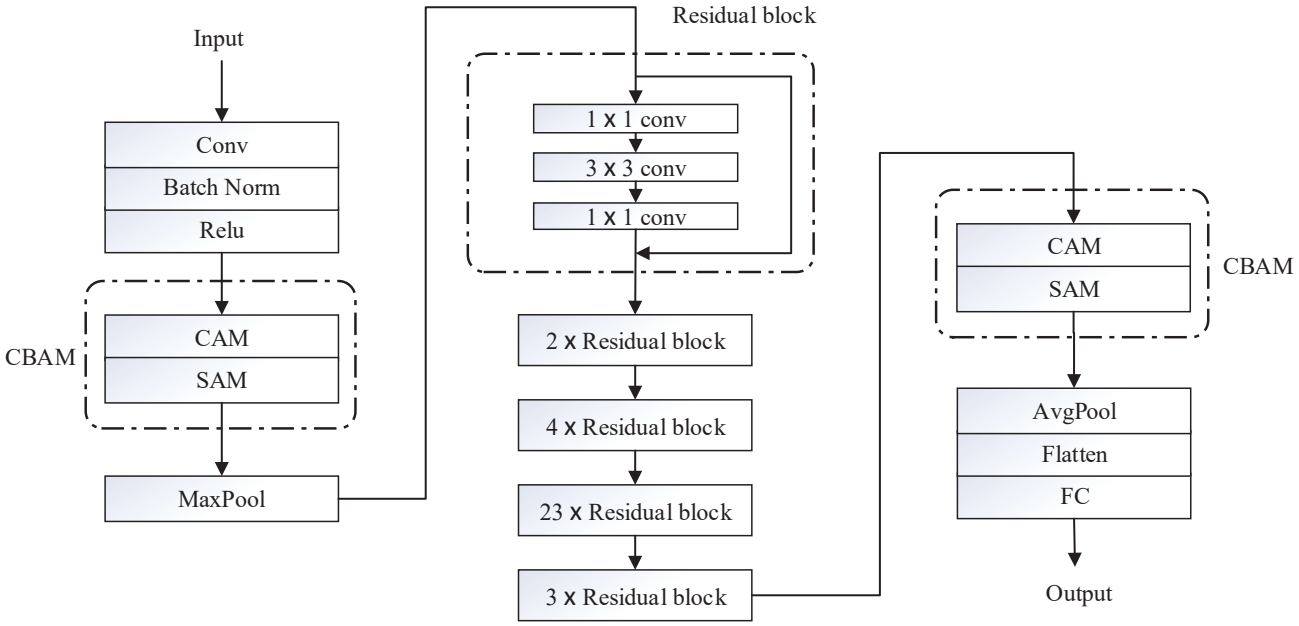


Figure 6: Illegal image classification model based on deep residual network and CBAM

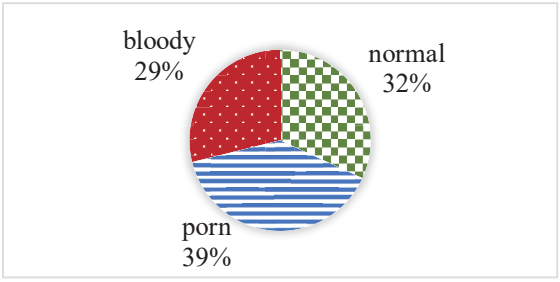


Figure 7: Dataset distribution

is an index of gradient accumulation. To a certain extent, in the multi-weight environment, the change direction of the parameters is more oriented towards the center, which reduces the swing amplitude of the gradient.

The Root Mean Square Prop (RMSProp) algorithm uses the differential square weighted average for the weights and biases, which not only corrects the amplitude of the swing, but also accelerates the convergence speed of the network. The formula for updating the parameters is:

$$s_{dw} = \beta_2 s_{dw} + (1-\beta_2)\, dw^2 \tag{3}$$

$$s_{\mathrm{db}} = \beta_2 s_{\mathrm{db}} + (1-\beta_2)\, \mathrm{d}b^2 \tag{4}$$

The Adaptive moment estimation (Adam) algorithm is an algorithm that combines Monentum and RMSprop [14]. It calculates the gradient momentum accumulated by the loss functions of the two algorithms in the iterative process, and corrects the deviation. The formula for calculating the gradient momentum of the weight is:

$$v_{dw}^{c} = \frac{v_{dw}}{1-\beta_1^t} \tag{5}$$

$$s_{dw}^{c} = \frac{s_{dw}}{1-\beta_2^t} \tag{6}$$

In order to avoid the excessive deviation generated during the update, the variable ε needs to be introduced.

Table 1: The influence of different hyperparameters on the accuracy of different classifications (Porn-P, Normal-N, Bloody-B)

Optimizer	LR Scheduler	Batch Size	Precision(%)			Recall(%)			F1-score(%)			Accuracy(%)
			P	N	B	P	N	B	P	N	B	
RMSprop	0.4	16	91.2	84.1	92.4	89.2	82.1	97.2	90.2	83.1	94.7	89.4
RMSprop	0.4	32	88.6	86.2	93.1	90.2	80.2	97.6	89.4	83.1	95.3	89.3
RMSprop	0.5	16	94.3	86.6	92.0	91.4	85.6	97.8	92.9	86.1	94.3	91.2
RMSprop	0.5	32	88.6	86.3	92.2	92.3	78.3	96.0	90.5	82.1	94.0	89.1
RMSprop	0.6	16	90.7	88.2	94.9	91.5	83.8	98.8	91.1	85.9	96.8	91.2
RMSprop	0.6	32	90.0	86.9	89.3	91.1	76.8	98.8	90.6	81.6	93.4	88.9
Adam	0.4	16	92.3	86.4	88.9	92.1	79.3	96.8	92.2	82.8	92.7	89.5
Adam	0.4	32	88.6	86.2	93.0	90.2	80.2	97.6	89.4	83.1	95.3	89.3
Adam	0.5	16	95.5	88.3	94.9	90.8	90.4	98.8	93.1	89.3	96.8	93.0
Adam	0.5	32	93.5	84.5	95.7	88.6	88.4	98.0	91.1	86.4	96.8	91.3
Adam	0.6	16	92.0	86.7	92.0	90.8	82.4	97.2	91.4	84.1	94.5	90.14
Adam	**0.6**	**32**	**94.4**	**91.1**	**93.9**	**93.9**	**86.9**	**99.2**	**94.2**	**88.9**	**96.5**	**93.2**

Update after bias correction:

$$w = w - \alpha \frac{v_{dw}^c}{\sqrt{s_{dw}^c} + \varepsilon} \tag{7}$$

$$b = b - \alpha \frac{v_{db}^c}{\sqrt{s_{db}^c} + \varepsilon} \tag{8}$$

Adam is essentially RMSprop with momentum, which has excellent performance in image classification tasks. This paper adopts the Adam algorithm, and the value of β_1 is 0.9, the value of β_2 is 0.999, and the value of ε is $1e-8$.

4.3 Hyperparameter Optimization

Too large learning rate may make the loss value of the late training unable to converge, too small learning rate will reduce the efficiency of the loss value of the early training. Gradually attenuating the learning rate can make the training drop quickly in the early stage, and the lowest point can be found well in the later stage. Exponential decay is a commonly used learning rate decay method during training. The principle is that after each round of training, the learning rate is multiplied by a fixed value less than 1, so that it is continuously reduced in multiple rounds of training. In this paper, the exponential decay method is used in the training process. There are 16 rounds of training in total, and the initial learning rate is 0.01.

Overfitting and underfitting are often encountered during training. Overfitting means that the model has a strong learning ability and learns too many unnecessary features in the training samples, but the effect is poor in the test samples. Underfitting means that the learning ability of the model is weak. When the data complexity is high, the general rules of the data set cannot be learned, and the generalization ability is weak. In order to solve the problem of underfitting and overfitting, a regularization method needs to be introduced. Both L1 regularization and L2 regularization methods add a regularization term after the cost function, and punish the influence of eigenvalues in the gradient descent process by controlling the size of the hyperparameters of the regularization term. This paper mainly adopts the L2 regularization method of weight decay, and the weight decay value is set to 0.01.

Mini-batch is a method of dividing the sample set, which enables efficient training of neural networks. The method is to divide the sample set into multiple small sample sets, so that the gradient descent algorithm can be performed multiple times, thereby improving the training efficiency.

The grid search method was used to optimize the hyperparameters, as shown in Table 1. The optimal parameter selection was obtained by comparing the indexes of accuracy, precision, recall and f1-score. When the accuracy is highest, the Adam optimizer is used, the batch size is 32, and the multiplier factor of the exponential decay rate is 0.6.

4.4 Training Process Analysis

As shown in Figure 8, we compare the training process of the three models: ResNet101, ResNet101+SE and ResNet101+CBAM. The Squeeze-and-Excitation (SE) is an attention module that weights the channel by averaging the channel characteristics [12]. The recognition rate of ResNet101 and ResNet101+SE increased to more than 65% in the second round of training, while the ResNet101+CBAM proposed by us improved relatively gently in the first few rounds of training, but in the first round of training. After eight rounds of training, its accuracy is significantly higher than the previous two networks. Finally, after the 11th round of training, the recog-

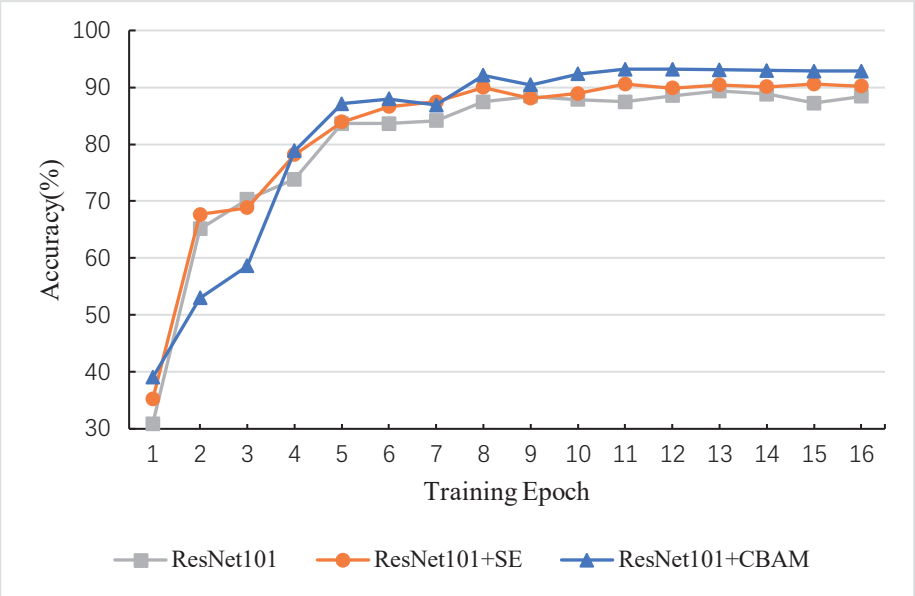


Figure 8: Recognition rate during network training

nition accuracy of the network reached 93.2%.

4.5 Analysis of Deep Residual Network Model

In this study, our proposed illegal image classification model based on deep residual network and CBAM is compared with the traditional machine learning model [7, 15] and CNN [13]. In addition, we trained three kinds of residual networks, Rsnet34, ResNet50 and ResNet101, as the backbone to compare the classification effect of networks with different residual structures. It can be seen from Table 2 that the ResNet101+CBAM model used in this paper has the highest recognition rate. Compared with the traditional machine learning model and CNN, the residual neural network is more competitive and has higher accuracy in illegal image recognition.

Table 2: Comparison of accuracy of different models

Method	Accuracy(%)
Skin	61.0
BoVM	84.9
MLP	84.7
CNN	86.9
ResNet34	89.5
ResNet50	88.9
ResNet101	89.4
ResNet101+SE	90.5
ResNet34+CBAM	90.3
ResNet50+CBAM	87.7
ResNet101+CBAM	93.2

By comparing the recognition accuracy of the three backbone networks, we find that the depth of the network is very important. The deeper the network, the more features are extracted, and the depth Residual networks can easily gain accuracy from depth increase [11], and using ResNet101 as the backbone network helps improve model accuracy.

The introduction of attention mechanism plays a significant role in our deep residual model. In order to verify the excellent performance of CBAM in illegal image classification tasks, we use the Squeeze-and-Excitation module to do a comparison test with the same parameters. The results show that in the illegal image classification task, the improvement effect of introducing the SE attention module is not as obvious as CBAM. Because SE only utilizes average pooling features, ignoring the importance of max pooling features. While CBAM uses max pooling features and average pooling features, it produces better attention effects than SE. At the same time, CBAM also combines the spatial attention module, which effectively refines the intermediate features and greatly improves the performance of the deep residual network.

4.6 System Functionality Testing

In this test, 100 images were randomly selected from each of the three categories of the data set, totaling 300 images, and mixed into a folder.

The system starts the test by selecting the folder to be tested and clicking the "Sort" button. The system preprocesses the images in the folder and inputs them into the deep residual neural network in batches. After the network calculation, the system displays the classification results of the images and classifies the images into

different categories of folders. The classification results are shown in Figure 9. The results of this test are 106 pornographic pictures, 101 bloody pictures and 93 normal pictures, which have a good classification effect.

Figure 9: Classification results display

5 Conclusions

In order to provide Internet users with a healthy network environment, we designed and implemented an illegal image classification system based on deep residual network and CBAM to address the problem of illegal images spreading freely on the Internet. The system's classifier uses ResNet101 as the backbone network, and CBAM is embedded in the middle of the model to improve the recognition accuracy of the model, and optimizes settings such as weight initialization and Adam algorithm for the convolutional neural network. During the training process, the exponential decay method of learning rate, the regularization method of weight decay and the Adam algorithm are used to make the network converge better. The final trained deep residual network for illegal image classification tasks can achieve a classification accuracy of 93.2%. We compare traditional machine learning models, convolutional neural networks, and network approaches using different residual structures and attention mechanisms. The experimental results show that our designed deep residual network embedded in CBAM has higher accuracy than other methods.

The proliferation of bad information on the Internet has always been a concern. As a hot research field in the information age, illegal image classification technology has achieved many research results, but there are still many problems to be solved. For pornographic images, there are various forms of expression and kinds of content, and many of them are hidden to a certain extent. Our proposed model has a mediocre performance on the recognition effect of pornographic images with sexual suggestive information. Moreover, our proposed model is trained offline, once and continuously identified. With the emergence of new illegal image data, how to realize online training of real-time image data is a problem that should be considered in the follow-up work. In order to create a good network environment, the classification technology of illegal images still needs to be continuously explored and improved in the future.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (62072416), Zhongyuan Science and Technology Innovation Leadership Program (214200510026), and Key Technologies R&D Program of Henan Province (212102210429, 222102210170, 222102210322).

References

[1] Z. An and H. Yuana, "Violent and terrorist image classification based on improved residual network," *International Core Journal of Engineering*, vol. 7, no. 4, 2021.

[2] R. Balamurali and A. Chandrasekar, "Multiple parameter algorithm approach for adult image identification," *Cluster Computing*, vol. 22, pp. 11909–11917, 2019.

[3] Z. Y. Cai, J. C. Wang, J. W. Zhang, and Y. J. Si, "Intrusion detection algorithm based on residual neural network," *International Journal of Network Security*, vol. 24, no. 6, pp. 1135–1141, 2022.

[4] J. Chen, G. Liang, W. He, C. Xu, J. Yang, and R. Liu, "A pornographic images recognition model based on deep one-class classification with visual attention mechanism," *IEEE Access*, vol. 8, pp. 122709 – 122721, 2020.

[5] S. L. Chen, C. Yang, C. Zhu, and X. C. Yin, "Bloody image classification with global and local features," in *Proceedings International Conference on Natural Computation*, vol. 663, pp. 379–391, Chengdu, China, NOV 2016.

[6] F. Cheng, S. L. Wang, X. Z. Wang, W. C. Liew, and G. S. Liu, "A global and local context integration dcnn for adult image classification," *Pattern Recognition*, vol. 96, 2019.

[7] T. Deselaers, L. Pimenidis, and H. Ney, "Bag-of-visual-words models for adult image classification and filtering," in *2008 19th International Conference on Pattern Recognition*, 2008.

[8] K. Dong, G. Li, and Q. Fu, "An adult image detection algorithm based on bag-of-visual-words and text information," in *Proceedings International Conference on Natural Computation*, pp. 556–560, Xiamen, China, AUG 2014.

[9] A. Gangwar, V. González-Castro, E. Alegre, and E. Fidalgo, "Attm-cnn: Attention and metric learning based cnn for pornography, age and child sexual abuse (csa) detection in images," *Neurocomputing*, vol. 445, pp. 81–104, 2021.

[10] J. Gupta, S. Pathak, and G. Kumar, "Aquila coyote-tuned deep convolutional neural network for the classification of bare skinned images in websites," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 10, pp. 3239–3254, 2022.

[11] K. He, X. Zhang, S. Ren, and J Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[12] J. Hu, S. Li, and S. Gang, "Squeeze-and-excitation networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020.

[13] S. M. Kia, H. Rahmani, R. Mortezaei, M. E. Moghaddam, and A. Namazi. "A novel scheme for intelligent recognition of pornographic images,", 2014.

[14] D. Liang, F. Ma, and W. Li, "New gradient-weighted adaptive gradient methods with dynamic constraints," *IEEE Access*, vol. 8, pp. 110929–110942, 2020.

[15] J. A. Marcial-Basilio, G. Aguilar-Torres, G. Sanchez-Perez, L. K. Toscano-Medina, and H. M. Pérez-Meana, "Detection of pornographic digital images," *Int. J. Comput*, vol. 5, no. 2, p. 298–305, 2011.

[16] F. Nian, T. Li, Y. Wang, M. Xu, and J. Wu, "Pornographic image detection utilizing deep convolutional neural networks," *Neurocomputing*, vol. 210, pp. 283–293, 2016.

[17] X. Ou, H. Ling, H. Yu, P. Li, F. Zou, and S. Liu, "Adult image and video recognition by a deep multicontext network and fine-to-coarse strategy," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 5, 2017.

[18] Z. Wang, R. Guo, H. Wang, and X. Zhang, "A new model for small target adult image recognition," in *Procedia Computer Science*, pp. 557–562, Wuhan, China, APR 2020.

[19] S. Woo, J. Park, and J. Y. Lee, "Cbam: Convolutional block attention module," in *15th European Conference on Computer Vision (ECCV)*, pp. 3–19, Munich, GERMANY, SEP 2018.

[20] W. Xu, P. Hamid, and I. Hadi, "Deep learning neural network for unconventional images classification," *Neural Processing Letters*, vol. 52, no. 1, pp. 169–185, 2020.

[21] H. Yin, X. Xu, and L. Ye, "Big skin regions detection for adult image identification," in *2011 Workshop on Digital Media and Digital Content Management*, pp. 242–247, May 2011.

Biography

Zengyu Cai is an associate professor fellow of the College of Computer and Communication Engineering of Zhengzhou University of Light Industry. His research interests include network se-curity and artificial intelligence.

Xinhua Hu is a graduate student fellow of the College of Software Engineering of Zhengzhou University of Light Industry. His research interests include network security and artificial intelligence.

Zhi Geng is an undergraduate student of the College of Software Engineering of Zhengzhou University of Light Industry. His research interests include network security and artificial intelligence.

Jianwei Zhang is a professor fellow of the College of Software Engineering of Zhengzhou University of Light Industry. His research interests include next generation network and artificial intelligence.

Yuan Feng is an associate professor fellow of the College of Computer and Communication Engineering of Zhengzhou University of Light Industry. Her research interests include cyberspace security and artificial intelligence.

Detecting DDoS Attacks in Software Defined Networks Using Deep Learning Techniques: A Survey

Ntumpha P. Mwanza and Jugal Kalita
(Corresponding author: Ntumpha P. Mwanza)

Department of Computer Science, University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy, Colorado Springs, CO 80918, USA
Email: pmwanza@uccs.edu

(Received June 17, 2022; Revised and Accepted Feb. 3, 2023; First Online Feb. 28, 2023)

Abstract

Deep Learning (DL) is increasingly being used in Software Defined Networks (SDNs) to detect Distributed Denial of Service (DDoS) attacks because of high attack detection accuracy. This paper presents a survey on the types of deep learning techniques used to detect DDoS attacks in SDNs. Attack statistics show that DDoS attacks are on an increase. Some of the factors that have contributed to the increase in DDoS attacks is the inability of current techniques to detect unknown DDoS attacks, which can be referred to as zero-day attacks. In this work, we look at deep learning techniques and how they are used to detect DDoS attacks. The current techniques' weaknesses are discussed and recommendations are made.

keywords: Deep Learning; Machine Learning; Software Defined Network; Traffic Classification

1 Introduction

Recent years have seen a sharp increase in internet traffic and at the same time a significant decrease in the use of network physical infrastructure. Network physical infrastructure is rapidly being replaced by the increased use of smart technology and the cloud as infrastructure. Smart technology has led to an increase in the use of smart IoT devices, connected to the internet. Virtualization has also contributed to an increase in internet traffic [49, 81]. Traditional networks are hardware-based, and for them to be used with smart technologies, there needs to be more hardware infrastructure in place to function effectively. For this reason, Software Defined Networks were introduced.

Software Defined Networks (SDN) were introduced because of the enormous increase in network connectivity and exposure to vulnerabilities. Because SDN is software-based, network management as well as its performance usually improves [68]. The SDN is made up of three planes, the Application Plane, the Control Plane, and the Data Plane [34]. The control plane is responsible for all the activities of the SDN. It has a centralized operational architecture, which makes managing network resources easy. The centralized architecture has made the SDN vulnerable to security risks and threats of attacks [67]. One common attack it is exposed to is the Distributed Denial of Service (DDoS) attack [41]. DDoS attacks attack the controller of the SDN. The controller controls and manages how the SDN operates and functions. A DDoS attack sends continuous requests to the SDN controller overwhelms the controller and denies legitimate traffic requests, which do not get any response from the network resources [15].

SDNs are widely used today because of their ability to separate the control plane from the data plane. The control plane and the data plane were separated because of increasing network traffic and the need to have a reliable network with high performance. The control plane is responsible for routing and network management. It operates by making a single application program able to control multiple programs. The Data Plane is responsible for forwarding programmable packets such as OpenFlow [94]. OpenFlow protocols let a server tell network switches where to route packets. The SDN controller can collect network data because it has an overall view of the network, making it easy to facilitate applications such as machine learning algorithms to be implemented in the controller [75]. Machine learning algorithms can be used to perform traffic analysis and improve traffic classification [43, 55].

This paper summarizes recent developments in detecting DDoS attacks in SDN using deep learning techniques. The SDN architecture is presented in Section 2, related surveys are presented in Section 3. Section 4 presents DD0S attacks. Deep and other machine learning techniques are presented in Section 5. Section 6 presents detecting DDoS attacks in SDN using deep learning tech-

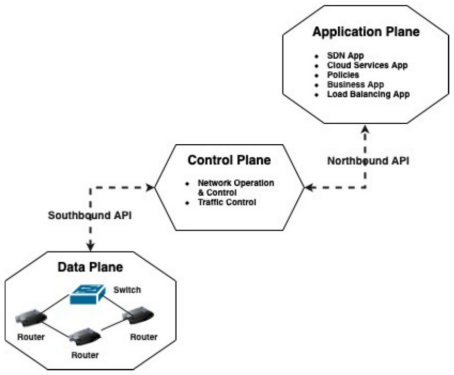


Figure 1: SDN Architecture

niques, Section 7 presents research issues and challenges and Section 8 is the conclusion.

2 SDN Architecture

Software Defined Networks break down the network into smaller disjoint parts. An SDN divides the functions of traditional networks into different parts and configures the parts according to functionality. Figure 1 shows the SDN architecture. Changes can be made instantly to the network functions. The SDN performs load balancing [11]. For example, if one area of the network is overwhelmed with data packets, the SDN routes the packets to where there is enough capacity. SDN uses one central point of operation, which is called the Control Plane to manage the entire network [1].

Each function of the SDN is programmed to operate automatically [8]. The SDN can allocate resources where they are needed the most on the network. The SDN has real-time centralized control of the network, which improves network performance and enhances the optimization of network function [70]. The other benefit of the SDN is virtualization [14]. Virtualization makes it possible to access both virtual and physical elements from one location. Virtualization allows multiple virtual networks to share resources from the same infrastructure and a virtual network has a simpler topology then the physical network.

2.1 Data Plane

Virtual switches and physical switches are found on the data plane [87]. Virtual switches communicate with other virtual machines by using software programs. Virtual switches are also referred to as software switches. Switches forward, drop, and modify packets received from the control plane [91]. The data and control planes communicate using network interfaces.

2.2 Control Plane

The control plane controls the entire operation of the network in a systematic and coordinated way [95]. It makes decisions and controls all operations through the SDN controller [7]. It provides control functionality using Application Programming Interfaces (APIs) to monitor the networks using an open interface. APIs are software interfaces that allow two or more applications to communicate with each other. The controller interacts with the data plane using the southbound API; the southbound APIs are used to communicate between the SDN controller and the switches and routers of the network [44]. The control plane updates forwarding rules that help with network management [93]. The southbound API facilitates the communication between the data plane and the control plane through the switches.

2.3 Application Plane

The responsibility of the application plane is to host applications that instruct the controller to perform changes depending on the requirements of its northbound APIs [47]. Applications such as end-user business applications, network virtualization, mobility management, and security applications are found on the application plane [24]. The application plane performs optimization and network management [76]. Depending on the network information and business requirements, the application plane can implement control logic, which is responsible for modifying network behavior.

3 Related Surveys

In this section, we present a review of published survey papers. SDN and Machine Learning (ML) techniques have improved the way DDoS attacks are detected and classified in SDN. The SDN controller provides centralized control and management, allocates resources, and directs network traffic. In a DDoS attack, the attacker attacks the controller so that network resources become inaccessible. Table 1 shows a summary of related survey papers on DDoS attacks in SDN. Table 1 is divided into two subsections; Subsection A is Traditional ML algorithms, and subsection B is SDN Techniques. Subsection A discusses published papers on how Traditional ML algorithms are applied in SDN to detect DDoS attacks. Subsection B discusses published papers on how SDN techniques are applied in SDN to detect DDoS attacks. SDN techniques protect the SDN by continuously monitoring network traffic. If malicious traffic is detected, the SDN controller takes action by applying the techniques to firewalls by updating the firewall rules to allow or block the traffic.

Table 1: Summary of related surveys on DDoS attacks in SDN

Publication	Year	Technique	Focused Area	Classification of DDoS	Research Recommendations
Zhao *et al.* [96]	2019	ML Algorithms	Network Security	No	✓
Yan *et al.* [92]	2018	ML Algorithms	Network Traffic Classification	No	✓
Nguyen *et al.* [60]	2018	ML Algorithms	Network Security Applications	No	×
Ahmad *et al.* [3]	2020	ML Algorithms	Network Security	No	✓
Sultana *et al.* [80]	2019	ML Algorithms	NIDS Security	No	✓
Gebremariam *et al.* [31]	2019	ML Algorithms	Network Security	No	×
Alamri *et al.* [5]	2021	ML Algorithms	Network Traffic Classification	No	✓
Sahoo *et al.* [72]	2018	ML Algorithms	Network Traffic Classification	No	×
Singh *et al.* [77]	2020	ML Algorithms	SDN Security	No	✓
Da Costa *et al.* [17]	2019	SDN Techniques	IoT Network Security	No	×
Dharmadhikari *et al.* [20]	2019	SDN Techniques	Network Security	No	×
Dantas *et al.* [18]	2020	SDN Techniques	IoT Security	No	×
Pajila *et al.* [12]	2019	SDN Techniques	IoT Security	No	×
Eliyan *et al.* [26]	2021	SDN Techniques	Network Security	No	×
Fajar *et al.* [28]	2018	SDN Techniques	Network Security	No	✓
Aladaileh *et al.* [4]	2020	SDN Techniques	Control plane Security	No	✓
Dong *et al.* [23]	2019	SDN Techniques	Cloud Security	No	✓
Ubale *et al.* [84]	2020	SDN Techniques	Network Security	No	×
Herrera *et al.* [9]	2019	SDN Techniques	Network Security	No	✓
Sahay *et al.* [71]	2019	SDN Techniques	Network Security	No	×
Our Survey	2022	Deep Learning	SDN Security	Yes	✓

3.1 Traditional Machine Learning Algorithms

Zhao *et al.* [96] discussed using ML algorithms in the context of SDNs in two ways. In the first approach, they used ML algorithms in SDN to classify network traffic. In the second approach, they used ML algorithms with network applications in SDN to classify network traffic. They compared the two approaches for performance and accuracy. They concluded that more has to be done to improve ML algorithms' ability to classify the network traffic in SDNs. Yan *et al.* [92] discussed new research on traffic classification technologies in SDNs. They analyzed challenges in traffic classification in SDNs and made recommendations.

Nguyen *et al.* [60] discussed the landscape of ML-enabled security intrusion detection in SDNs. They analyzed the vulnerabilities and attack methods and concluded by developing a new ML-based SDN security mechanism. Ahmad *et al.* [3] discussed evaluating traditional ML algorithms such as SVM and Logistic Regression to counter DoS and DDoS attacks in SDNs. Results showed that SVM produced the best results against all the other traditional machine learning algorithms used.

Sultana *et al.* [80] discussed traditional machine learning algorithms that leverage SDNs to implement Network Intrusion Detection Systems (NIDSs). They outlined various intrusion detection mechanisms using Deep Learning approaches. They used the SDN as the platform to carry out the analysis. They concluded that more needs to be done to be able to monitor real-time intrusion detection systems in high-speed networks. Gebremariam *et al.* [31] discussed traditional ML algorithms used for different applications such as network planning, management, and security in SDN and Network functions virtualization (NFV) environments. NFV is the replacement of network appliance hardware with virtual machines [13]. They concluded by laying out the challenges of detecting DDoS attacks in SDN and NFV using traditional ML algorithms.

Alamri *et al.* [5] reviewed and compared traditional ML algorithms to detect DDoS attacks in SDN. They evaluated NB, K-NN, SVM, DT, RF, and XGBoost algorithms based on accuracy, precision, recall, and f1-score. Results showed that XGBoost had the best overall performance. Sahoo *et al.* [72] discussed using traditional ML algorithms to detect DDoS attacks in SDN. They compared KNN, NB, SVM, RF, and LR algorithms for performance based on prediction and classification accuracy. Results showed that LR had the best prediction and classification accuracy compared to the other algorithms.

Singh *et al.* [77] discussed the SDN architecture and its ability to protect itself against DDoS attacks. The authors reviewed over 70 published publications in this area. Results from the review showed that 47% of the approaches are theory-based, 42% used traditional machine learning-based and 20% used artificial neural network-based.

3.2 Software Defined Network Techniques

Da Costa *et al.* [17] discussed providing security to network infrastructure using ML techniques. The ML techniques were used to enhance the Internet of Things (IoT) and an Intrusion Detection System (IDS). Results showed challenges in fully securing IoT and IDS systems. Dharmadhikari *et al.* [20] discussed and summarized DDoS attacks on SDNs and how they are detected and mitigated. The authors made a comparison using past and present studies and recommended the need to have strong mitigation techniques in place. They also acknowledged that DDoS attacks cannot be fully prevented.

Dantas *et al.* [18] discussed the need to come up with new virtual techniques to prevent DDoS attacks in SDNs in an IoT environment. The authors explained that when a technique is applied based on a scenario in the IoT environment, justification has to be given regarding its suit-

ability. A summary was provided based on the strengths and weaknesses of the techniques to detect DDoS attacks in an IoT environment.

Pajila *et al.* [12] discussed the growth and security of IoT systems. It is heterogeneous in design, and the way it operates through the internet exposes it to DDoS attacks. The authors concluded that there is a gap in modeling platforms, such as not having context-based security, and clients controlling access. Eliyan *et al.* [26] discussed two countermeasures that can be used for detecting, mitigating, and preventing DoS and DDoS attacks in SDNs. The two approaches are Intrinsic and Extrinsic approaches. The intrinsic approach is applied to SDN components and their functionalities. The extrinsic approach is applied to the network traffic flow and feature characteristics in SDNs. They concluded that more research had to be performed to improve the detection accuracy of DoS and DDoS attacks in SDNs.

Fajar *et al.* [28] discussed security vulnerabilities in the SDN controller. They concluded that the current defense mechanisms are not effective against DDoS attacks. Aladaileh *et al.* [4] discussed techniques used to detect DDoS attacks in SDNs. They explained the important role the SDN controller plays in protecting the SDNs. They further gave a detailed summary of the state-of-the-art and made future research recommendations such as combining different DDoS attack patterns, to create a more complex and effective defense technique.

Dong *et al.* [23] discussed DDoS attacks in both SDNs and the Cloud, along with a summary of DDoS attacks in SDNs and how they are detected and prevented. The authors recommended using traffic classification models to improve DDoS attack detection. Ubale *et al.* [84] discussed an SDN's ability to prevent DDoS attacks because of architectural design. The authors gave specific types of DDoS attacks that the SDN is unable to prevent and recommended future research to help thwart vulnerabilities against DDoS attacks.

Herrera *et al.* [9] discussed security concerns of SDNs by comparing different studies already concluded in this area by universities and the cybersecurity industry. The authors focused on security concerns such as the effectiveness of the current countermeasures used to detect DDoS attacks in SDN. They recommended more research to address the security concerns of the SDN.

Sahay *et al.* [71] discussed the benefits of SDNs in providing network security compared to traditional networks. The centralized architecture of the SDN controller makes it easy to dynamically configure the network, and also makes it easy to identify and mitigate DDoS attacks. The controller can analyze the entire network traffic in real time because of its global view of the network. The authors recommended more research on SDNs and applications situated in SDNs. They recommended that new SDN applications pay attention to network security.

4 DDoS Attacks

DDoS attacks attack the server or network with the intention of disrupting its normal function [38]. They continuously send malicious traffic requests to the system. The system is eventually unable to respond to requests and stops working and functioning normally. The system becomes unable to respond to requests coming from legitimate sources or users [73]. Legitimate users are trying to order books from Amazon.com website; as they browse through the website looking for specific books, the requested website is sending requests to the server database [74]. At the same time, the attackers are also sending multiple fake malicious requests to the server as well. The attackers' continuous requests coming from different sources overwhelm the system and use up its network bandwidth [10]. This causes the legitimate users' computers to be denied service because the server has been overwhelmed with fake malicious requests. Figure 2 presents a taxonomy of the types of DDoS attacks, and Figure 3 illustrates a DDoS attack.

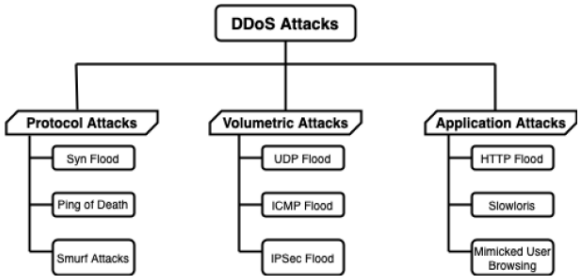


Figure 2: Types of DDoS attacks

4.1 Types of DDoS Attacks

4.1.1 Volume-Based Attacks

Volume-based attacks are DDoS attacks that aim to deplete the bandwidth of a network system [42]. The attackers use bots to amplify the attack by spreading malware which is transmitted through the bots and is spread through network traffic. Examples of volumetric-based attacks are UDP flood, ICMP flood, and IPSec flood attacks.

4.1.2 Protocol Attacks

Protocol attacks are DDoS attacks that prevent a legitimate user's computer from establishing a connection with the host computer [53]. This attack uses the 3-way handshake, SYN, SYN-ACK, and ACK, to carry out an attack. The legitimate user client computer sends a request (SYN), to the host computer. The host computer responds to the client computer accepting to establish a connection (SYN-ACK). Once the client computer receives the acknowledgment to establish a connection, it

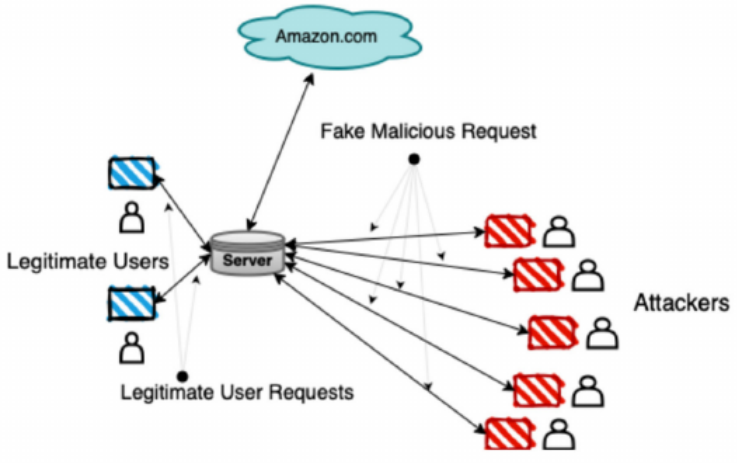


Figure 3: Distributed denial of service attack

sends back an acknowledgment (ACK), and then the connection is established. What the protocol attack does is when the client computer sends a request to the host computer to establish a connection, the attackers send continuous SYN requests, to the host computer at the same time the client computer sends the request to establish a connection. The aim is to exhaust the system so that it is unable to respond and establish a connection with the client's computer. Syn flood, Ping of Death, and Smurf DDoS attacks are examples of protocol attacks.

4.1.3 Application Plane Attacks

Application plane attacks attack the applications that make networks function properly [51]. They attack the applications by taking advantage of security flaws in the applications to carry out an attack. The application becomes unable to communicate with other applications and users. HTTP Flood, Slowloris, and Mimicked User Browsing attacks are an example of application plane attacks [32].

5 Deep and other Machine Learning Techniques

Attackers are taking advantage of the growing number of IoT devices connected to the internet, and the increasing amount of network traffic to the internet to launch DDoS attacks. Attackers are using more complex and sophisticated attack methods to carry out these DDoS attacks, which are difficult to detect. Because of the large amount of labeled data used to carry out these attacks, deep learning-based detection techniques may be the best techniques to use to detect DDoS attacks. DL techniques produce the best detection rate and classification accuracy when a large amount of labeled data is available. Compared with DL techniques, traditional ML techniques may not produce as high accuracy in detecting DDoS attacks.

DL techniques require a large amount of labeled data input to produce the desired accuracy. The data used for training need to be correctly labeled for the trained model to correctly classify unseen examples. If the input data are not correctly labeled, the model is unlikely to produce the correct classification. Most DDoS attacks are unknown attacks (zero-day attacks); this means the training dataset does not contain any similar labeled examples. DL techniques are the best techniques to detect DDoS attacks because:

- DL techniques are excellent, during training at discovering useful hidden features in the data. A trained DL network can extract features from previously unseen examples, and classify such examples well.
- Some DL techniques can learn long-term dependencies of temporal patterns.

Below we present the taxonomy of commonly used DL techniques for detecting DDoS attacks in SDN in Figure 4 and we discuss the techniques. We end the section with a short discussion of relevant traditional machine learning approaches also since many of the papers presented in this survey refer to such approaches in addition to DL methods.

5.1 Discriminative Learning Techniques

Discriminative Learning Techniques are techniques that learn the boundaries between classes in a dataset. The techniques use probability estimates and maximum likelihood to create new instances, in order to find the boundary separating one class from the other.

5.1.1 Multilayer Perceptron

Multilayer Perceptron (MLP) are neural networks that are made up of one or more densely connected hidden layers between the input and output layers. Figure 5 shows the architecture of MLP. MLPs are trained by adjusting

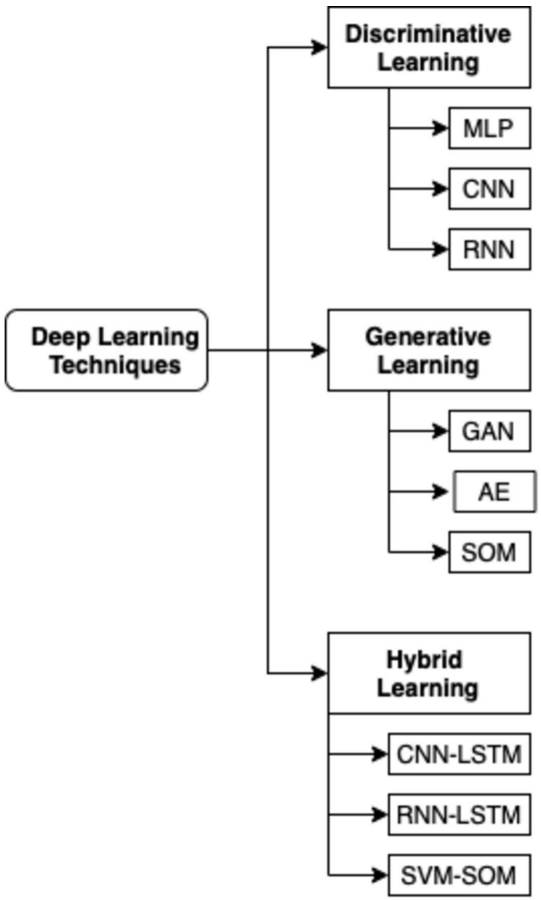


Figure 4: Taxonomy of detection methods for DDoS attacks in SDN using deep learning techniques

the weights of each connection after it is shown a dataset of labeled examples one by one [82]. The error or loss between the expected result and the output of the MLP determines the adjustments to be made to the weights. This process continues until the loss is reduced to a level that does not change the outcome.

5.1.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are primarily used for image classification and object detection [37]. CNNs have strong extraction capabilities that are used to automatically extract useful features from input data. The input data is passed through different layers of the CNN for feature extraction. As data move from one layer to the other, features are extracted at various levels of abstraction [30]. Figure 6 shows an example of CNN architecture.

5.1.3 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are used in Natural Language Processing (NLP) [57]. The input at a time step is processed and produces an output. Then the output of this step is processed together with the input to the next step. This allows the RNN to remember the inputs in the previous steps in a sequence [36]. The output of RNN at a certain step is dependent on previous input elements in a sequence as illustrated in Figure 7. In this figure, A is the input layer, B is the hidden layer with a recurrent loop, and C is the output layer. X, Y, and Z are network parameters used to improve the output of the model.

5.2 Generative Learning Techniques

Generative Learning Techniques are techniques that focus on the distribution of individual classes in a dataset. The technique uses likelihood and probability estimates to model data points and differentiates between class labels in a dataset. The technique uses joint probability, by creating instances where a given feature input (a) and the desired output (b) exist at the same time.

5.2.1 Generative Adversarial Networks

Generative Adversarial Networks (GANs) consist of two models, a Generator and a Discriminator [2]. The generator creates fake samples, which are then used to fool the discriminator [52]. The discriminator is trained on real as well as generated fake samples, and learns to distinguish between real and fake samples well. After the discriminator has determined whether a sample is real or fake, the result is sent back to the generator which is trained to generate better fake images. The generator is always trained with real samples. Figure 8 shows the GAN architecture.

5.2.2 Autoencoders

Autoencoders (AEs) are unsupervised neural networks [86]. They consist of two parts, the Encoder, and the Decoder. The encoder takes the input and learns how to compress and encode the data into a code. Then the decoder learns how to reconstruct the encoded data representation to create output [54]. The output is similar to the original input. The difference between the input and output is the input contains signals that have noise while the output has no noise, the signal has been denoised, as shown in Figure 9.

5.2.3 Self-Organizing Maps

Self-Organizing Maps (SOMs) are shallow neural networks that are trained on unlabelled data [35]. They are used for clustering high-dimensional inputs to easily visualize the two-dimensional output. A SOM has two layers, the input layer, and the output layer connected by edges with weights [90]. The weights determine the specific location of each neuron in the two-dimensional space. Weights are trained and updated to change the position of neurons into clusters that we can easily see. They can also be used to combine diverse datasets to find patterns. Figure 10 shows the SOM architecture.

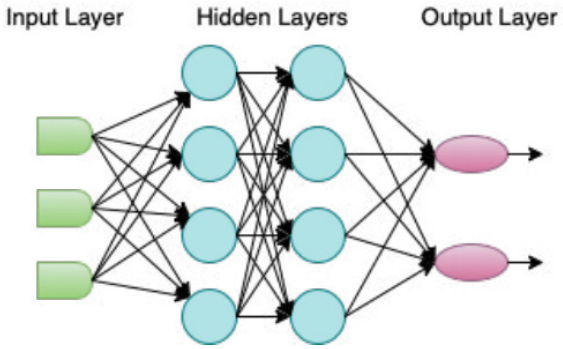


Figure 5: Multilayer perceptron

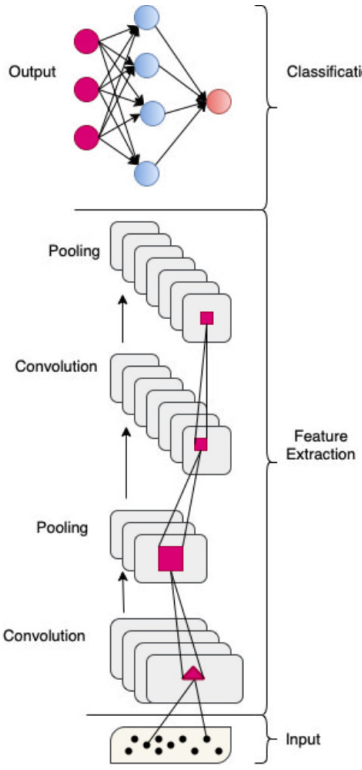


Figure 6: Convolutional neural network

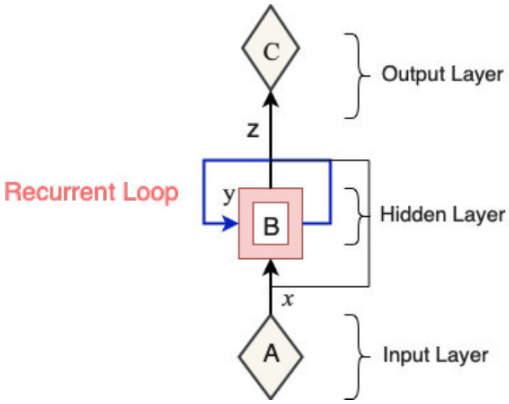


Figure 7: Recurrent neural networks

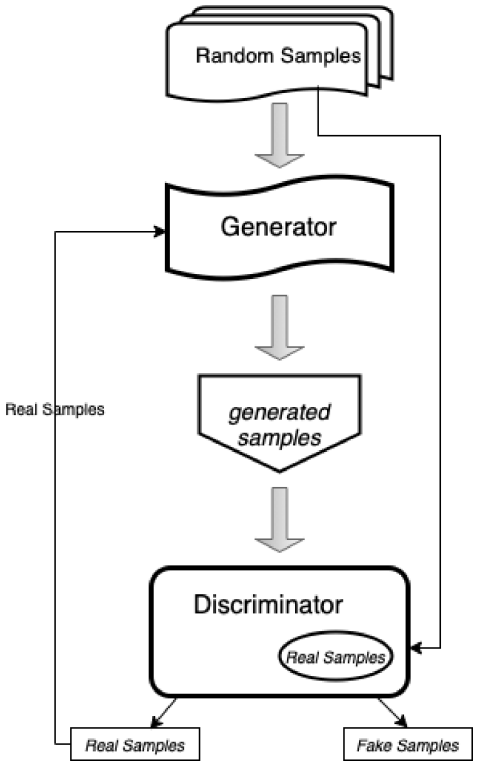


Figure 8: Generative adversarial networks

5.3 Hybrid Learning Techniques

Hybrid learning techniques are techniques that are comprised of a combination of two or more deep learning models, such as discriminative or generative deep learning models. The combination of these models can be used to extract more meaningful and robust features, depending on the target use.

5.3.1 Convolutional Neural Network-Long Short Term Memory

The Convolutional Neural Network-Long Short Term Memory (CNN-LSTM) model has been used for sequence prediction with spatial inputs. Figure 11 shows the CNN-

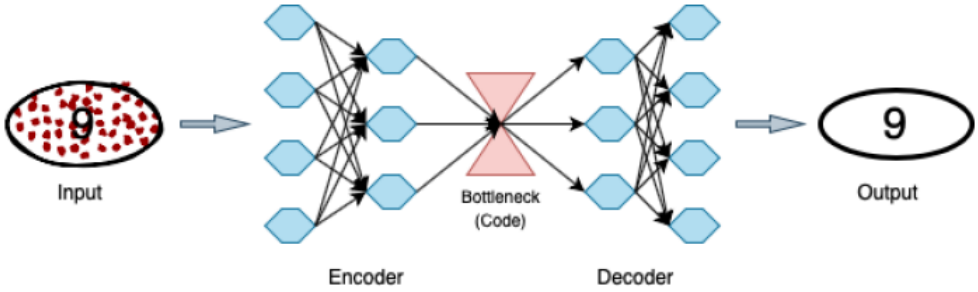


Figure 9: Autoencoders

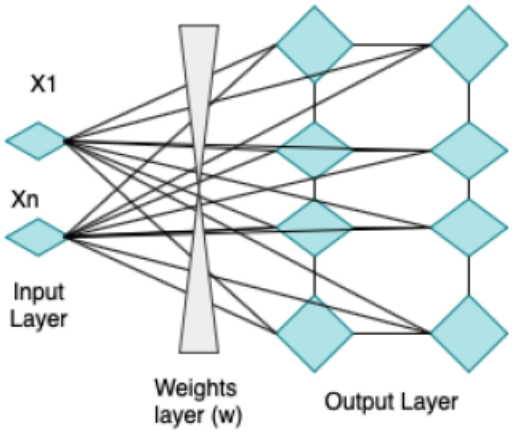


Figure 10: Self-organizing maps

LSTM architecture [46, 50]. The input are images that are fed into the CNN. The CNN extracts features from the image and feeds them to the [40]. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning long-term dependencies. Traditional Recurrent Neural Networks are incapable of learning to detect long-term dependencies, and LSTMs, were introduced to fix this problem.

5.4 Reinforcement Learning Technique

Reinforcement Learning is a machine learning technique that involves an agent learning how to interact with its environment by performing actions and seeing the results of actions [22]. The agent learns the best action to maximize its long-term reward. The agent gets positive feedback for each good action and negative feedback or a penalty for each bad action. The agent learns from the feedback and experience from its environment without depending on any labeled data [89].

5.5 Traditional Machine Learning Technique

Traditional machine learning methods use computational, statistical, and mathematical methods to deploy algorithms that extract patterns out of raw data on input. They automatically learn from the data and past experiences and be able to make a prediction. Traditional machine learning classification algorithms usually can perform well when trained with a small amount of data compared to DL. However, DL approaches usually produce better accuracy, assuming a large amount of labeled data is available.

5.5.1 Support Vector Machine

Support Vector Machines (SVMs) are supervised learning algorithms that have been widely used for solving complex classification and regression problems [65]. SVMs can perform data transformations that can be leveraged to determine boundaries between data classes when trained on examples from predefined classes. Given a set of high-dimensional data points or vectors in a vector space, it looks for the separating hyperplane that separates the vector space into subspaces containing sub-sets of vectors. Each sub-set corresponds to one class. Assuming binary classification, the separating hyperplane maximizes the margin between the two subspaces. Classification can be performed by finding the hyperplane that differentiates the two classes very well.

5.5.2 Decision Tree

A Decision Tree (DT) is a supervised machine learning technique that produces a tree-like structured classifier in which data is repeatedly divided at each row based on certain rules until the outcome is generated [64]. They are used for solving both classification and regression problems. A DT has two types of nodes, Decision Nodes, and Leaf Nodes. The decision nodes are used to make decisions and the leaf nodes are the output of those decisions and do not add any more branches.

5.5.3 Random Forest

Random Forest (RF) is a supervised machine learning technique that uses an ensemble of decision trees for both classification and regression. The forest consists of a number of decision trees created by sampling training instances and sampling attributes of training instances. Each tree individually classifies an unseen instance, and the classification with the most votes is selected [79].

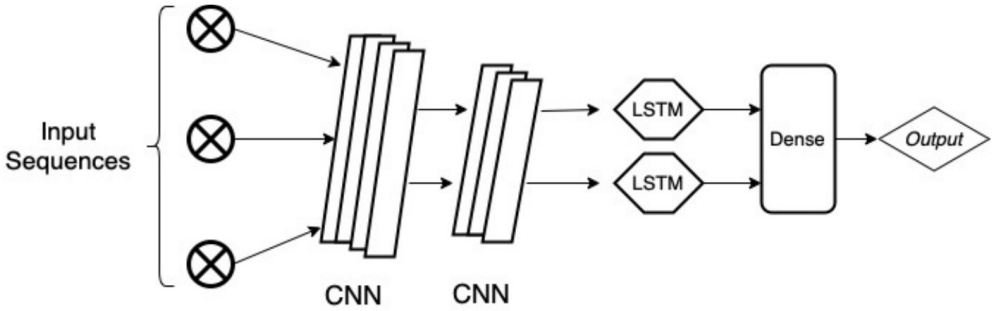


Figure 11: Convolutional neural network-long short-term memory

5.5.4 Naive Bayes

Naïve Bayes (NB) is a supervised learning technique used for classification and applies the Bayes theorem, with the assumption that all features of the data instances are independent of each other. The features independently contribute to making a probability of a label given the observed features [16].

5.5.5 Logistic Regression

Logistic Regression (LR) is a supervised learning technique that is used to predict the probability of the occurrence of a class with the help of independent variables or features. There are only two outcomes or classes, 1 or 0, true or false [98].

6 Detecting DDoS Attacks in SDN Using Deep Learning Techniques

An SDN uses the controller to control the entire network's functions by intelligently allocating and prioritizing network resources. The advantages of SDNs are that they can work without human intervention, can be programmed to make decisions, can allocate network resources, and can route traffic to the right destination within the SDN. The SDN controller is responsible for the security of the SDN. Unfortunately, the SDN controller itself is vulnerable to DDoS attacks.

The centralized architecture of the SDN exposes it to DDoS attacks and is regarded as a single source of failure. An SDN itself is unable to determine whether the network traffic is normal or anomalous, which makes it difficult for the SDN to detect and prevent DDoS attacks. This vulnerability has led to the introduction of Deep Learning for the detection of DDoS attacks. Deep learning intelligently learns data flow features in the network traffic and classifies them as either normal or anomalous.

Below we discuss published papers on deep learning approaches used to detect DDoS attacks in SDN. The section is divided into Discriminative, Generative, and Hybrid Learning approaches. Table 2 is a summary of published papers on approaches to Detect DDoS Attacks in SDNs. The tables is divided into approaches, year the papers were published, the techniques used and the datasets used.

6.1 Discriminative Learning Techniques

Lee *et al.* [48] proposed DL Intrusion Detection and Prevention System (DL-IDPS) to detect and prevent DDoS and brute force attacks in SDN. They evaluated the performance of the proposed system with MLP, CNN, and LSTM models. Results showed that the system produced the best performance with an accuracy of 99% detecting brute force attacks and 100% accuracy detecting DDoS attacks.

Wang *et al.* [88] proposed an SDN-Home Gateway (HGW) framework that improves the SDN controller management of smart devices connected to the network. The SDN-HGW can control end-to-end network management. But SDN-HGW cannot carry out real-time encrypted packet inspection, which puts the network at risk of DDoS attacks. To overcome this risk, the authors proposed a classifier called DataNet, developed based on MLP and CNN models. DataNet can detect and classify encrypted network packets in real-time. Results from an evaluation showed that DataNet had a detection and classification accuracy of 98%.

Narayanadoss *et al.* [59] proposed a DL model that relies on SDN traffic to get information about the flow size and timestamp measurements. They compared 3 techniques, MLP, CNN, and LSTM, to determine how many correlations are present in the traffic flow from compromised nodes. Results showed that all the models achieved above 80% detection rate of compromised nodes and LSTM had the best detection rate of 87%.

Janabi *et al.* [39] proposed a DL Early Warning Proactive System (DL-EWPS) predict network attacks in SDNs

Table 2: Summary of approaches to detect DDoS attacks in SDNs

Approach	Publication	Year	Technique	Dataset Used
Discriminative	Lee *et al.* [48]	2020	MPL,CNN,LSTM	-
	Wang *et al.* [88]	2018	CNN	ISCX
	Narayanadoss *et al.* [59]	2019	CNN,RNN,LSTM	Mininet-WiFi
	Janabi *et al.* [39]	2022	CNN	InSDN
	Haider *et al.* [33]	2020	CNN	CICIDS2017
	Polat *et al.* [66]	2022	RNN	-
Generative	AlEroud *et al.* [6]	2019	GAN	DARPA
	Novaes *et al.* [62]	2021	GAN	CICDDoS 2019
	Ujjan *et al.* [85]	2020	SAE	SM1, SM2
	Meng *et al.* [56]	2020	SOM	-
Hybrid	Khan *et al.* [45]	2021	CNN-LSTM	IOT-23
	Nugraha *et al.* [63]	2020	CNN-LSTM	-
	Ding *et al.* [21]	2020	Hybrid CNN	UNSW-NB15, KDDCup 99
	Qin *et al.* [69]	2019	CNN+RNN	SIM-DATA, CTU-13
	Gadze *et al.* [29]	2021	RNN-LSTM	-
	Elsayed *et al.* [27]	2020	RNN	CICDDoS2019
	Deepa *et al.* [19]	2019	SVM-SOM	CAIDA 2016
	Nam *et al.* [58]	2018	SOM	DDoS Attack 2007
	Novaes *et al.* [61]	2020	LSTM-Fuzzy	CICDDoS 2019

using CNN for classification. The system converted numerical data to RGB images to improve CNN classification and added extra features from flow tables statistics. The system achieved 100% DDoS attack classification accuracy.

Haider *et al.* [33] proposed an ensemble CNN framework to detect DDoS attacks in SDN. The authors compared the ensemble CNN with ensemble RNN, ensemble LSTM, and hybrid reinforcement learning. They used the CICIDS2017 dataset which was fully labeled with 80 features of network traffic with benign and attack traffic. They used random forest regression for the feature selection of the 80 features of the network traffic. They evaluated the proposed model with other models. The proposed ensemble CNN framework achieved the best accuracy in detecting DDoS attacks with 99.45%.

Polat *et al.* [66] proposed an Recurrent Neural Network (RNN) classifier to detect DDoS attacks in SDN-based Supervising Control and Data Acquisition (SCADA) system. The RNN classifier was evaluated in with Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) models. Results showed the proposed RNN classifier had the best accuracy of 96.67% detecting attacks in SDN-based SCADA system.

6.2 Generative Learning Techniques

AlEroud *et al.* [6] proposed an approach that generates attacks on SDN to train the SDN to learn to detect attacks. Generative Adversarial Networks were used for adversarial training. They evaluated the approach using two scenarios; in the first scenario, GAN was not used and in the second scenario, GAN was used. Results showed the second scenario had the best performance by accurately identifying attacks in SDN when GAN was used, and the first scenario had low attack detection accuracy in identifying attacks in SDN when GAN was not used.

Novaes *et al.* [62] proposed an adversarial training system that uses Generative Adversarial Network (GAN) to detect and defend the SDN against DDoS attacks. The system was evaluated with other methods, CNN, LSTM, and MLP. The methods were tested in two separate scenarios. In the first scenario, the GAN had the best accuracy of 99.78% detecting DDoS attacks in SDN. In the second scenario, the GAN had the best accuracy of 94.38% detecting DDoS attacks in SDNs. Thus, the GAN had the best performance in both scenarios.

Ujjan *et al.* [85] proposed a sflow and Adaptive Polling band sampling with Snort IDS and Stacked Autoencoders (SAE) for detecting DDoS attacks in IoT networks. The model uses snort IDS to identify network traffic and uses Stacked Autoencoder to classify traffic as either benign or DDoS attack traffic. Snort IDS and SAE were used in both sFlow and Adaptive polling sampling. Adaptive polling sampling is an algorithm is used to refine polling intervals based on the rate of change of network traffic flow. Results showed good sflow achieved 95% DDoS attack accuracy with less than 4% false positive rate and Adaptive polling-based sampling achieved 95% DDoS attack accuracy with less than 8% false positive rate.

Meng *et al.* [56] proposed a SOM-based DDoS attack defense mechanism to detect DDoS attacks on Internet of Things devices. The mechanism uses the connection between the SDN and the Internet of Things (IoT) devices to detect DDoS attacks. When a DDoS attack is detected, the proposed mechanism blocks traffic to and from the IoT devices. Results showed the proposed mechanism accurately detected DDoS attacks in IoT devices.

6.3 Hybrid Learning Techniques

Khan *et al.* [45] The proposed Hybrid Deep Learning architecture comprises CNN (Convolution Neural Network) and LSTM (Long Short Term Memory) models to detect sophisticated malware attacks in the Internet of Medical Things (IoMT). The model was evaluated with Convolution Neural Network (CNN) and Gated Recurrent Units (GRU) model, and Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) model. Results showed that the CNN-LSTM model outperformed CNN-GRU and GRU-LSTM models with an accuracy of 99.83% detecting sophisticated IoMT malware.

Nugraha *et al.* [63] proposed a Hybrid CNN-LSTM model used to detect Slow DDoS attacks in SDN-based networks. Slow DDoS attacks are a type of DDoS attack that aim to disrupt services provided by an application to a network server by sending small amounts of attack data with legitimate traffic over a long period. They used this model because of its high accuracy and recall when detecting different types of DDoS attacks. The model uses the output from the feature extractor to classify. Results showed the model achieved 99% accuracy in detecting slow DDoS attacks. The model was evaluated against MLP and 1-class SVM models, and it outperformed both methods.

Ding *et al.* [21] Proposed a Hybrid Convolutional Neural Network (HYBRID-CNN) to extract deep features from the smart grid network flow that traditional ML methods connect extract. The HYBRID-CNN is a method that utilizes CNNs to effectively memorize global features by one-dimensional (1D) data and to generalize local features by two-dimensional (2D) data. Two datasets were used for the evaluation. The method was evaluated and compared with other models, LSTM and CNN-LSTM. The Hybrid-CNN had the highest performance with an accuracy of 95.64% and had the highest detection rate of 98.56%.

Qin *et al.* [69] proposed a CNN-RNN model to detect and classify anomalies in network traffic. They compared the CNN-RNN model with a Tree-Shaped deep Neural Network (TSDNN) model using two datasets, CTU-13 [78] and a self-generated dataset Sim-data dataset. CNN-RNN model had the best accuracy of 99.8% compared to TSDNN with 99.7% accuracy in detecting network attacks.

Gadze *et al.* [29] proposed using CNN and RNN-LSTM to detect DDoS attacks such as TCP, UDP, and ICMP flood attacks on the SDN Controller. They also compared the performance of the DL models with traditional ML techniques. Performance was based on accuracy, recall, true-negative rate, and time taken in detecting and mitigating DDoS attacks in the SDN controller. RNN-LSTM had the best results overall in detecting DDoS attacks in the SDN controller.

Elsayed *et al.* [27] proposed DDoSNet, an intrusion detection method to detect DDoS attacks in SDN. This method uses an RNN-AE to detect and classify benign or DDoS attack traffic on input. The model is in two stages, the unsupervised pre-training stage and the time-tuning stage. The first stage extracts useful feature representation, and the second stage trains the last layer of the network using labeled samples. The proposed method had an accuracy of 99% correctly detecting and classifying DDoS in SDN.

Deepa *et al.* [19] proposed an ensemble technique using k-Nearest Neighbors (kNN), Naïve Bayes (NB), SVM, and SOM techniques to detect DDoS attacks in SDN controller. The hybrid SVM-SOM outperformed the other models with a detecting accuracy of 98%.

Nam *et al.* [58] proposed a DDoS attack detection algorithm that uses Self Organizing Map (SOM) with other techniques such as k-Nearest Neighbors (kNN), SOM-kNN, SOM distributed neurons, and SOM distributed center to classify network traffic as normal or anomalous. The techniques' performance was evaluated based on detection rate, false-positive rate, and processing time. kNN had the best detection rate and largest processing time. SOM-kNN had the second-best detection rate and the lowest false positive rate. SOM-kNN had the best performance because it had the best DDoS attack classification rate and the lowest false positive rate.

Novaes *et al.* [61] proposed an LSTM-FUZZY model that characterizes, detects, and mitigates DDoS and Portscan attacks in SDN environments. Two scenarios were used to evaluate the model. In the first scenario, the LSTM-FUZZY model was evaluated against k-Nearest Neighbor (KNN), LSTM-2, MLP, Particle Swarm Optimization Digital Signature (PSO-DS) [25], and SVM models to detect DDoS attacks in SDN environment by applying mitigation policies based on the attack type identified by the detection module. LSTM-FUZZY had the best performance with 96.22% DDoS attack accuracy. In the second scenario, the CICDDoS 2019 dataset was used. LSTM-FUZZY achieved 99.20% DDoS attack accuracy.

7 Research Issues and Challenges

Attackers have improved their DDoS attack approaches and techniques. These improvements have made DDoS attacks more sophisticated and hard for most defense systems to detect. The improvements have also led to an increase in the number of DDoS attack types. Each attack type has a different attack pattern, which makes it difficult to put effective defensive systems in place. Most of these attacks are zero-day attacks, which have no defense system developed. Defense approaches are usually developed only after an attack has already materialized. Deep learning techniques have shown to be effective in the accurate detection and classification of attacks, including making zero-day defense possible. Based on the published papers we have read, we have observed the following.

- Experiments conducted show that the source and destination IP addresses were not used when extracting features and classification of the network traf-

fic. But the source and destination IP addresses are very important for analyzing and classifying traffic as either normal or anomaly. After the features are extracted, they are used to classify and categorize as either normal or anomaly traffic. Removing the source and destination IP addresses will not classify the traffic accurately.

- DL techniques are used in both virtual and physical networks to detect DDoS attacks. An experiment was performed [29] to compare the performance between RNN-LSTM, SVM, and Naive Bayes to detect DDoS attacks such as TCP, UDP, and ICMP in the SDN controller. The experiment was conducted only on virtual networks and concluded that the defense system was effective in detecting and classifying DDoS attacks accurately. The problem with this approach is that the authors did not experiment with their approach on physical networks. The types of DDoS attacks on virtual and physical networks are different. For example, DDoS attacks like the Reflection attacks can only attack physical networks, but not virtual networks. The results may have been accurate on virtual networks, but physical networks were not tested.

- Many defense systems leverage DL techniques to improve the detection and classification accuracy of DDoS attacks in SDNs. Most of these defense systems focus only on attacks that are in the network traffic, from input to output. They do not focus on attacks that attack physical devices such as switches and routers. For example, attacks that are not detected on input might attack physical devices on the network that are responsible to store or route data packets. These DDoS attacks can overwhelm the system by rerouting traffic and sending fake requests to the SDN controller. This problem can be overcome using an Intrusion Prevention System (IPS). The IPS is a piece of network security software or hardware that continuously monitors the network for threats and can automatically apply countermeasures to stop the attack or attacks by dropping packets and blocking traffic to affected hardware.

- Published literature tells us how good the defense systems that use DL techniques are in detecting DDoS attacks in SDN. Results show the proposed systems have excellent accuracy in classifying traffic as normal or as DDoS. But most of these results do not tell us what the false positive and negative rates are. With an increase of new DDoS at- tacks, it is important to be able to know how accurately the defense systems can still detect and classify traffic correctly without having high false positives and negatives. It is important to develop defense systems that will have low or no false positive and negative rates on new attacks, as the attackers are becoming good at fooling the defense system in place.

- DL techniques use datasets that contain different historical DDoS attack patterns to help train the DL models so that they can accurately detect and classify DDoS attacks that match the patterns in the dataset. This has proved to be an effective approach for detecting known attacks. But this approach is not effective for detecting zero-day DDoS attacks, because the attack pattern(s) are unknown. The problem is that historical attack patterns cannot be used to predict or discover new attacks and generate new patterns. The attack has to materialize for the attack patterns to be added to a dataset. Datasets are effective in detecting and classifying known DDoS attacks. The DL technique's accuracy is dependent on how accurate the data is in the dataset.

- Even if there is a sophisticated machine learning approach, datasets used for training may not provide all the feature patterns needed, to carry out a comprehensive analysis of network traffic so that a system can identify DDoS attacks accurately. Datasets may have limitations. One of the most important limitations is the size of datasets; a smaller dataset may not store all the necessary features that come with an attack. Datasets should be large enough to be able to contain a large amount of DDoS attack patterns and features that can be used by DL techniques training models and help the models accurately detect and classify DDoS attacks.

8 Deep Learning Techniques Limitations

- DL techniques need large amounts of data to learn patterns in the data. With the rising number of new DDoS attacks, the model needs to be continuously updated with the new attack data. But the new attack data may not be available to train the model on the new DDoS attack patterns.

- Using a small amount of training data in DL models such as CNNs is likely to produce low accuracy and is unable to generalize well to unseen examples. The situation may be ameliorated with transfer learning [97] and/or multi-task learning [83] but this needs further investigation.

- Good hardware support is another DL limitation. DL techniques require a good amount of hardware support because of their high computational power, which is expensive to acquire and maintain. This problem can be addressed by using a Graphical Processing Unit (GPU). GPUs are needed for training, but trained models usually run fast.

- DL techniques are unable to reassign, re-categorize and relabel previously stored data without retraining the model.

- Using poor-quality of data with errors and noise for training will prevent the DL model from detecting required patterns and the model will not perform well.

9 Conclusion

In this paper, we examined different published papers that use DL techniques to detect DDoS attacks in SDN. We compared three categories of DL, discriminative, generative, and hybrid learning. Results show that DL techniques have good performance in accurately detecting and classifying DDoS attacks. With the increase in internet connection traffic through smart devices, IoT plays a major role in the increase of DDoS attacks. DDoS attacks will spread by being part of the internet or network traffic. DL techniques are the best methods to detect DDoS attacks because an increase in internet traffic (training data), will make the techniques learn more robust features and increase the classifier's performance. If performance falls, the number of hidden layers can be increased to improve performance and feature classification accuracy. But DL techniques still depend on datasets for training on known attack patterns, which are stored as historical data in the dataset. Combining different DL techniques has also shown improvement in DDoS attack detection accuracy on SDN networks.

Acknowledgments

All authors approved the version of the manuscript to be published.

References

[1] A. Abuarqoub, "A review of the control plane scalability approaches in software defined networking," *Future Internet*, vol. 12, no. 3, pp. 49, 2020.

[2] A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," *International Journal of Information Management Data Insights*, vol. 1, no. 1, pp. 100004, 2021.

[3] A. Ahmad, E. Harjula, M. Ylianttila, and I. Ahmad, "Evaluation of machine learning techniques for security in SDN." in *IEEE Globecom Workshops*, pp. 1–6, 2020.

[4] M. A. Aladaileh, M. Anbar, I. H. Hasbullah, Y. W. Chong, and Y. K. Sanjalawe, "Detection techniques of distributed denial of service attacks on software defined networking controller–a review," *IEEE Access*, vol. 8, pp. 143985–143995, 2020.

[5] H. A. Alamri and V. Thayananthan, "Analysis of machine learning for securing software-defined networking," *Procedia Computer Science*, vol. 194, pp. 229–236, 2021.

[6] A. AlEroud and G. Karabatis, "Sdn-gan: generative adversarial deep nns for synthesizing cyber attacks on software defined networks," in *OTM Confederated International Conferences*, pp. 211–220, 2020.

[7] J. Ali and B. Roh, "Quality of service improvement with optimal software-defined networking controller and control plane clustering," *Comput. Mater. Contin*, vol. 67, pp. 849–875, 2021.

[8] N. Anerousis, P. Chemouil, A. A. Lazar, N. Mihai, and S. B. Weinstein, "The origin and evolution of open programmable networks and SDN," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1956–1971, 2021.

[9] J. Arevalo Herrera and J. E. Camargo, "A survey on machine learning applications for software defined network security," in *International Conference on Applied Cryptography and Network Security*, pp. 70–93, 2019.

[10] M. Asad, M. Asim, T. Javed, M. O. Beg, H. Mujtaba, and S. Abbas, "Deepdetect: Detection of distributed denial of service attacks using deep learning," *The Computer Journal*, vol. 63, no. 7, pp. 983–994, 2020.

[11] M. R. Belgaum, S. Musa, M. M. Alam, and M. M. Suúd, "A systematic review of load balancing techniques in software-defined networking," *IEEE Access*, vol. 8, pp. 98612–98636, 2020.

[12] P. Beslin Pajila and E. Golden Julie, "Detection of DDoS attack using SDN in IoT: A survey," in *Intelligent Communication Technologies and Virtual Mobile Networks*, pp. 438–452, 2019.

[13] M. S. Bonfim, K. L. Dias, and S. F. Fernandes, "Integrated NFV/SDN architectures: A systematic literature review," *ACM Computing Surveys*, vol. 51, no. 6, pp. 1–39, 2019.

[14] P. Borylo, G. Davoli, M. Rzepka, A. Lason, and W. Cerroni, "Unified and standalone monitoring module for NFV/SDN infrastructures," *Journal of Network and Computer Applications*, vol. 175, pp. 102934, 2021.

[15] A. Cetinkaya, H. Ishii, and T. Hayakawa, "An overview on denial-of-service attacks in control systems: Attack models and security analyses," *Entropy*, vol. 21, no. 2, pp. 210, 2019.

[16] S. Chen, G. I. Webb, L. Liu, and X. Ma, "A novel selective näive bayes algorithm," *Knowledge-Based Systems*, vol. 192, pp. 105361, 2020.

[17] K. A. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, 2019.

[18] F. S. Dantas Silva, E. Silva, E. P. Neto, M. Lemos, A. J. Venancio Neto, and F. Esposito, "A taxonomy of DDoS attack mitigation approaches featured by SDN technologies in IoT scenarios," *Sensors*, vol. 20, no. 11, pp. 3078, 2020.

[19] V. Deepa, K. M. Sudar, and P. Deepalakshmi, "Design of ensemble learning methods for DDoS detec-

tion in SDN environment," in *International Conference on Vision Towards Emerging Trends in Communication and Networking*, pp. 1–6, 2019.

[20] C. Dharmadhikari, S. Kulkarni, S. Temkar, S. Bendale, and B. Student, "A study of DDoS attacks in software defined networks," *IRJET*, vol. 6, no. 12), 2019.

[21] P. Ding, J. Li, L. Wang, M. Wen, and Y. Guan, "Hybrid-cnn: An efficient scheme for abnormal flow detection in the SDN-based smart grid," *Security and Communication Networks*, vol. 2020, 2020.

[22] Z. Ding, Y. Huang, H. Yuan, and H. Dong, "Introduction to reinforcement learning," in *Deep Reinforcement Learning*, pp. 47–123, 2020.

[23] S. Dong, K. Abbas, and R. Jain, "A survey on distributed denial of service , no. ddos) attacks in SDN and cloud computing environments," *IEEE Access*, vol. 7, pp. 808130-828, 2019.

[24] Z. Eghbali and M. Z. Lighvan, "A hierarchical approach for accelerating IoT data management process based on SDN principles," *Journal of Network and Computer Applications*, vol. 181, pp. 103027, 2021.

[25] M. Elbes, S. Alzubi, T. Kanan, A. Al-Fuqaha, and B. Hawashin, "A survey on particle swarm optimization with emphasis on engineering and network applications," *Evolutionary Intelligence*, vol. 12, no. 2, pp. 113–129, 2019.

[26] L. F. Eliyan and R. Di Pietro, "Dos and DDoS attacks in software defined networks: A survey of existing solutions and research challenges," *Future Generation Computer Systems*, vol. 122, pp. 149–171, 2021.

[27] M. S. Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, "Ddosnet: A deep-learning model for detecting network attacks," in *IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks*, pp. 391–396, 2020.

[28] A. P. Fajar and T. W. Purboyo, "A survey paper of distributed denial-of-service attack in software defined networking," *International Journal of Applied Engineering Research*, vol. 13, no. 1, pp. 476–482, 2018.

[29] J. D. Gadze, A. A. Bamfo-Asante, J. O. Agyemang, H. Nunoo-Mensah, and K. A. B. Opare, "An investigation into the application of deep learning in the detection and mitigation of DDoS attack on SDN controllers," *Technologies*, vol. 9, no. 1, pp. 14, 2021.

[30] T. T. Gao, H. Li, and S. L. Yin, "Adaptive convolutional neural network-based information fusion for facial expression recognition," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 17–23, 2021.

[31] A. A. Gebremariam, M. Usman, and M. Qaraqe, "Applications of artificial intelligence and machine learning in the area of SDN and NFV: A survey," in *16th International Multi-Conference on Systems, Signals & Devices*, pp. 545–549, 2019.

[32] B. B. Gupta and A. Dahiya, "Distributed Denial of Service , no. DDoS) Attacks: Classification, Attacks, Challenges and Countermeasures," *CRC press*, 2021.

[33] S. Haider, A. Akhunzada, I. Mustafa, T. B. Patel, A. Fernandez, K. K. R. Choo, and J. Iqbal, "A deep cnn ensemble framework for efficient DDoS attack detection in software defined networks," *Ieee Access*, vol. 8, pp. 53972–53983, 2020.

[34] S. H. Haji, S. Zeebaree, R. H. Saeed, S. Y. Ameen, H. M. Shukur, N. Omar, M. A. Sadeeq, Z. S. Ageed, I. M. Ibrahim, and H. M. Yasin, "Comparison of software defined networking with traditional networking," *Asian Journal of Research in Computer Science*, vol. 9, no. 2, pp. 1–18, 2021.

[35] A. A. Hameed, B. Karlik, M. S. Salman, and G. Eleyan, "Robust adaptive learning approach to self-organizing maps," *Knowledge-Based Systems*, vol. 171, pp. 25–36, 2019.

[36] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, "Recurrent neural network wave functions," *Physical Review Research*, vol. 2, no. 2, pp. 023358, 2020.

[37] M. Hussain, J. J. Bird, and D. R. Faria, "A study on cnn transfer learning for image classification," in *UK Workshop on computational Intelligence*, pp. 191–202, 2018.

[38] G. A. Jaafar, S. M. Abdullah, and S. Ismail, "Review of recent detection methods for http DDoS attack," *Journal of Computer Networks and Communications*, vol. 2019, 2019.

[39] A. H. Janabi, T. Kanakis, and M. Johnson, "Convolutional neural network-based algorithm for early warning proactive system security in software defined networks," *IEEE Access*, vol. 10, pp. 14301–14310, 2022.

[40] D. Jiang, H. Li, and S. Yin, "Speech emotion recognition method based on improved long short-term memory networks," *International Journal of Electronics and Information Engineering*, vol. 12, no. 4, pp. 147–154, 2020.

[41] B. Karan, D. Narayan, and P. Hiremath, "Detection of DDoS attacks in software defined networks," in *3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions*, pp. 265–270, 2018.

[42] S. Karapoola, P. K. Vairam, S. Raman, and V. Kamakoti, "Net-police: A network patrolling service for effective mitigation of volumetric DDoS attacks," *Computer Communications*, vol. 150, pp. 438–454, 2020.

[43] R. Karthika and M. Maheswari, "Detection analysis of malicious cyber attacks using machine learning algorithms," *Materials Today: Proceedings*, vol. 68, pp. 26-34, 2022.

[44] S. Kaur, K. Kumar, N. Aggarwal, and G. Singh, "A comprehensive survey of DDoS defense solutions in SDN: Taxonomy, research challenges, and future directions," *Computers & Security*, vol. 110, pp. 102423, 2021.

[45] S. Khan and A. Akhunzada, "A hybrid dl-driven intelligent SDN-enabled malware detection framework for internet of medical things," *Computer Communications*, vol. 170, pp. 209–216, 2021.

[46] T. Y. Kim and S. B. Cho, "Predicting residential energy consumption using cnn-lstm neural networks," *Energy*, vol. 182, pp. 72–81, 2019.

[47] Z. Latif, K. Sharif, F. Li, M. M. Karim, S. Biswas, and Y. Wang, "A comprehensive survey of interface protocols for software defined networks," *Journal of Network and Computer Applications*, vol. 156, pp. 102563, 2020.

[48] T. H. Lee, L. H. Chang, and C. W. Syu, "Deep learning enabled intrusion detection and prevention system over SDN networks," in *IEEE International Conference on Communications Workshops*, pp. 1–6, 2020.

[49] J. Li, D. Li, Y. Yu, Y. Huang, J. Zhu, J. Geng, "Towards full virtualization of SDN infrastructure," *Computer Networks*, vol. 143, pp. 1–14, 2018.

[50] P. Li, M. Abdel-Aty, and J. Yuan, "Real-time crash risk prediction on arterials based on LSTM-CNN," *Accident Analysis & Prevention*, vol. 135, pp. 105371, 2020.

[51] H. Lin, S. Cao, J. Wu, Z. Cao, and F. Wang, "Identifying application-layer DDoS attacks based on request rhythm matrices," *IEEE Access*, vol. 7, pp. 164480–164491, 2019.

[52] X. Liu and C. J. Hsieh, "Rob-gan: Generator, discriminator, and adversarial attacker," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11234–11243, 2019.

[53] A. D. Lopez, A. P. Mohan, and S. Nair, "Network traffic behavioral analytics for detection of DDoS attacks," *SMU data science review*, vol. 2, no. 1, pp. 14, 2019.

[54] T. Luo and S. G. Nagarajan, "Distributed anomaly detection using autoencoder neural networks in wsn for IoT," in *IEEE International Conference on Communications*, pp. 1–6, 2018.

[55] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research*, vol. 9, pp. 381–386, 2020.

[56] Y. Meng, Z. Huang, S. Wang, G. Shen, and C. Ke, "Som-based DDoS defense mechanism using SDN for the internet of things," *arXiv preprint arXiv*, 2003.06834, 2020.

[57] M. Morchid, "Parsimonious memory unit for recurrent neural networks with application to natural language processing," *Neurocomputing*, vol. 314, pp. 48–64, 2018.

[58] T. M. Nam, P. H. Phong, T. D. Khoa, T. T. Huong, P. N. Nam, N. H. Thanh, L. X. Thang, P. A. Tuan, V. D. Loi, *et al.*, "Self-organizing map-based approaches in DDoS flooding detection using SDN," in *International Conference on Information Networking*, pp. 249–254, 2018.

[59] A. R. Narayanadoss, T. Truong-Huu, P. M. Mohan, and M. Gurusamy, "Crossfire attack detection using deep learning in software defined its networks," in *IEEE 89th Vehicular Technology Conference*, pp. 1–6, 2019.

[60] T. N. Nguyen, "The challenges in SDN/ml based network security: A survey," *arXiv preprint arXiv*, 1804.03539, 2018.

[61] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proenca, "Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment," *IEEE Access*, vol. 8, pp. 83765–83781, 2020.

[62] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proenca Jr, "Adversarial deep learning approach detection and defense against DDoS attacks in SDN environments," *Future Generation Computer Systems*, vol. 125, pp. 156–167, 2021.

[63] B. Nugraha and R. N. Murthy, "Deep learning-based slow DDoS attack detection in SDN-based networks," in *IEEE Conference on Network Function Virtualization and Software Defined Networks*, pp. 51–56, 2020.

[64] H. H. Patel and P. Prajapati, "Study and analysis of decision tree-based classification algorithms," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 10, pp. 74–78, 2018.

[65] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning*, pp. 101–121, 2020.

[66] H. Polat, M. Türkoğlu, O. Polat, and A. Sengür, "A novel approach for accurate detection of the DDoS attacks in SDN-based scada systems based on deep recurrent neural networks," *Expert Systems with Applications*, vol. 197, pp. 116748, 2022.

[67] A. Pradhan and R. Mathew, "Solutions to vulnerabilities and threats in software defined networking , no. sdn)," *Procedia Computer Science*, vol. 171, pp. 2581–2589, 2020.

[68] M. Priyadarsini and P. Bera, "Software defined networking architecture, traffic management, security, and placement: A survey," *Computer Networks*, vol. 192, pp. 108047, 2021.

[69] Y. Qin, J. Wei, and W. Yang, "Deep learning based anomaly detection scheme in software-defined networking," in *20th Asia-Pacific Network Operations and Management Symposium*, pp. 1–4, 2019.

[70] D. S. Rana, S. A. Dhondiyal, and S. K. Chamoli, "Software defined networking (SDN) challenges, issues and solution," *International Journal of Computer Science and Engineering*, vol. 7, no. 1, pp. 884–889, 2019.

[71] R. Sahay, W. Meng, and C. D. Jensen, "The application of software defined networking on securing computer networks: A survey," *Journal of Network and Computer Applications*, vol. 131, pp. 89–108, 2019.

[72] K. S. Sahoo, A. Iqbal, P. Maiti, and B. Sahoo, "A machine learning approach for predicting DDoS traffic in software defined networks," in *International*

Conference on Information Technology, pp. 199–203, 2018.

[73] 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, no. 7, pp. 5320–5363, 2020.

[74] A. Sangodoyin, B. Modu, I. Awan, and J. P. Disso, "An approach to detecting distributed denial of service attacks in software defined networks," in *IEEE 6th International Conference on Future Internet of Things and Cloud*, pp. 436–443, 2018.

[75] I. H. Sarker, "Machine learning: Algorithms, real world applications and research directions," *SN Computer Science*, vol. 2, no. 3, pp. 1–21, 2021.

[76] A. Shirmarz and A. Ghaffari, "Performance issues and solutions in SDN-based data center: A survey," *The Journal of Supercomputing*, vol. 76, no. 10, pp. 7545–7593, 2020.

[77] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions," *Computer Science Review*, vol. 37, pp. 100279, 2020.

[78] K. Sinha, A. Viswanathan, and J. Bunn, "Tracking temporal evolution of network activity for botnet detection," *arXiv preprint arXiv*, 1908.03443, 2019.

[79] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, "A comparison of random forest variable selection methods for classification prediction modeling," *Expert Systems with Applications*, vol. 134, pp. 93–101, 2019.

[80] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, 2019.

[81] J. Sun, Y. Zhang, F. Liu, H. Wang, X. Xu, Y. Li, "A survey on the placement of virtual network functions," *Journal of Network and Computer Applications*, vol. page 103361, 2022.

[82] H. Taud and J. Mas, "Multilayer perceptron , no. mlp)," in *Geomatic Approaches for Modeling Land Change Scenarios*, pp. 451–455, 2018.

[83] K. H. Thung and C. Y. Wee, "A brief review on multitask learning," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29705–29725, 2018.

[84] T. Ubale and A. K. Jain, "Survey on DDoS attack techniques and solutions in software-defined network," in *Handbook of Computer Networks and Cyber Security*, pp. 389–419, 2020.

[85] R. M. A. Ujjan, Z. Pervez, K. Dahal, A. K. Bashir, R. Mumtaz, and J. González, "Towards sflow and adaptive polling sampling for deep learning based DDoS detection in SDN," *Future Generation Computer Systems*, vol. 111, pp. 763–779, 2020.

[86] C. Wang, B. Wang, H. Liu, and H. Qu, "Anomaly detection for industrial control system based on autoencoder neural network," *Wireless Communications and Mobile Computing*, vol. 2020, 2020.

[87] H. Wang, H. Xu, C. Qian, J. Ge, J. Liu, and H. Huang, "Prepass: Load balancing with data plane resource constraints using commodity SDN switches," *Computer Networks*, vol. 178, pp. 107339, 2020.

[88] P. Wang, F. Ye, X. Chen, and Y. Qian, "Datanet: Deep learning based encrypted network traffic classification in SDN home gateway," *IEEE Access*, vol. 6, pp. 55380–55391, 2018.

[89] Z. Wang and T. Hong, "Reinforcement learning for building controls: The opportunities and challenges," *Applied Energy*, vol. 269, pp. 115036, 2020.

[90] C. S. Wickramasinghe, K. Amarasinghe, and M. Manic, "Deep self-organizing maps for unsupervised image classification," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 11, pp. 5837–5845, 2019.

[91] R. Xia, H. Dai, J. Zheng, H. Xu, M. Li, and G. Chen, "Packet-in request redirection: A load balancing mechanism for minimizing control plane response time in SDNs," *Journal of Systems Architecture*, vol. 129, p. 102590, 2022.

[92] J. Yan and J. Yuan, "A survey of traffic classification in software defined networks," in *1st IEEE International Conference on Hot Information-Centric Networking*, pp. 200–206, 2018.

[93] L. Yang, B. Ng, W. K. Seah, L. Groves, and D. Singh, "A survey on network forwarding in software-defined networking," *Journal of Network and Computer Applications*, vol. 176, pp. 102947, 2021.

[94] L. Yao, J. Liu, D. Wang, J. Li, and B. Meng, "Formal analysis of SDN authentication protocol with mechanized protocol verifier in the symbolic model," *International Journal of Network Security*, vol. 20, no. 6, pp. 1125–1136, 2018.

[95] A. Yazdinejadna, R. M. Parizi, A. Dehghantanha, and M. S. Khan, "A kangaroo-based intrusion detection system on software-defined networks," *Computer Networks*, vol. 184, pp. 107688, 2021.

[96] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A survey of networking applications applying the software defined networking concept based on machine learning," *IEEE Access*, vol. 7, pp. 95397–95417, 2019.

[97] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[98] X. Zou, Y. Hu, Z. Tian, and K. Shen, "Logistic regression model optimization and case analysis," in *IEEE 7th International Conference on Computer Science and Network Technology*, pp. 135–139, 2019.

Biography

Ntumpha Patrick Mwanza received his Bsc. in Computer Science degree from Cavendish University, Zambia, and M.S in Cybersecurity and Leadership from the University of Washington Tacoma, USA. Currently, he is a

Ph.D. student at the University of Colorado, Colorado Springs. His research interests include Distributed Denial of Service (DDoS) attack detection, Malware Detection and Classification using deep learning.

Jugal Kalita received his Ph.D. from the University of Pennsylvania, Philadelphia. He is a professor and department chair of computer science at the University of Colorado, Colorado Springs. His research interests are in machine learning and natural language processing. He has published over 250 papers in international journals and referred conference proceedings and has written four books.

Research on Privacy and Security of Federated Learning in Intelligent Plant Factory Systems

Wen-Pin Hu[1], Chin-Bin Lin[1], Jing-Ting Wu[2], Cheng-Ying Yang[3], and Min-Shiang Hwang[4]
(Corresponding author: Cheng-Ying Yang)

Department of Bioinformatics and Medical Engineering, Asia University, Taiwan[1]
Department of Management Information Systems, National Chung Hsing University, Taiwan[2]
Department of Computer Science, University of Taipei, Taiwan[3]
Email: cyang@utaipei.edu.tw
Department of Computer Science & Information Engineering, Asia University, Taiwan[4]
*(Received Aug. 30, 2022; Revised and Accepted Feb. 23, 2023; First Online Feb. 28, 2023)**

Abstract

Intelligent plant factories must continuously collect sensing data from other factories to update parameters in real time and obtain the best plant growth environment. However, if the collected sensing data is forged or collected, it will affect the performance of machine learning. In this paper, after using the federated learning technology to train the local data in each region, upload the gradient loss and generate blocks in the blockchain. Finally, verify and aggregate it into a complete model trained with all the data instead of directly uploading the local data to ensure privacy and data security.
keywords: Deep Learning; Federated Learning; Intelligent Plant Factory Systems; Machine Learning

1 Introduction

Plant factories could be the environment that could be controlled and opened according to the plan and could be the open sites for plant production throughout the year [27,36]. The successful conditions of the plant's products include controlling the appropriate growth environment and reducing plant growth stress. For example, in the growth environment control, if it needs shadow effects, the light from the artificial device should be turned down lightly. Similarly, it does the same processes for temperature and humidity control [24, 29].

In 2018, we proposed an intelligent indoor plant factory system framework [9]. The system uses a wireless sensor network [2, 3, 7, 8, 10, 15, 16, 28]. In the system, we will plant with natural agricultural cultivation. There are no fertilizers and pesticides in the system. There are four modules in the proposed system. They are the plant factory system setup model, the collecting data model, the transmitting data model, and the analyzing data module. These models describe the best plant growth environment with big data and artificial intelligence technologies.

In this paper, after using the federated learning technology to train the local data in each region, upload the gradient loss and generate blocks in the blockchain. Finally, verify and aggregate it into a complete model trained with all the data instead of directly uploading the local data to ensure privacy and data security.

The rest of this paper is organized as follows. First, in Section 2, we will review our framework of the intelligent plant factory system. Then, in Section 3, we introduce the federated learning. Next, in Section 4, we propose the framework of privacy and security of the federated learning for intelligent plant factory systems. Finally, Section 5 gives the conclusion.

2 The Framework of the Intelligent Plant Factory System

In 2018, we designed an intelligent plant factory system for the indoor environment. The proposed system architecture of the indoor plant factory system is given in Figure 1 [9]. There are four modules in the proposed system: Setup plant factory system, collecting data, transmitting data, and analysis of data modules.

Module 1 is the construction of the plant factory environment. In the plant factory system, we were planted by natural agricultural cultivation. There are no fertilizers and pesticides in the system. This module will include an automatic water-delivered system. The air conditioning system controls temperature. Also it includes the humidity control system, the carbon dioxide supply system, the LED light control system, the photography monitoring system, and the automatic control system. These automatic control systems are according to the collected data from the other plant factors.

*This article is an extended version of an ICICT'18 paper [9].

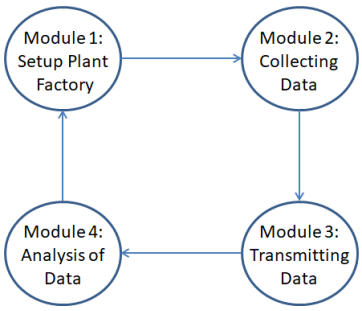


Figure 1: Hu *et al.*'s indoor plant factory system architecture [9]

1) The automatic liquid water delivery system includes the containers, the receiving pumps, piping, the sprinkler, and the recovery system. The container is used to store water. Water should be prepared enough for use first. The pumps and the pipes are mainly provided in a fixed mode. Water from the container was consistently sprayed onto each plant. The humidity control system controls the sprinkler system. When it is too dry, the sprinkler system begins to work. If it is too wet, the recovery system begins to work. The recovery system has a storage tank to recover the irrigated water. The recovery system could refollow water to the container by the heavy-water pump from the pipeline or canal after the container is drained.

2) The temperature in the plants must be kept at a range of a specific degree for a high-quality metabolism. Unstable temperatures give a disadvantage to the growth of plants. Therefore, the factory needs to make a temperature control to give a better cultivation effect. The plant factory achieves temperature control with temperature sensors and the air conditioning system. Besides, the air conditioning system also provides the necessary breeze for the plant growth environment. With the breeze, the surface of leaves will make the stomatal conductance and enhance the speed of growth.

3) The plant factory uses a water spray system for environmental humidity control. For the small houseplant factory, a spray system is suitable. For large plants, the water system could be employed. The equipment of the spray system is relatively simple. Usually, it includes the water supply pipe and the spray head. For the nozzle choice, it chooses the specific atomization spray head generally.

4) In contrast to oxygen for human beings, carbon dioxide is food for plants. It is the most critical component of leaf photosynthesis. Plant factories have the potential to provide carbon dioxide for large production. However, since carbon dioxide can be consumed quickly in the limited cultivated space, the plants grow poorly without enough carbon dioxide. Designing a carbon dioxide supply system requires a ventilation duct from the top of the plant to increase the rate of carbon dioxide. It will solve the difficulty and make uniformly distributed carbon dioxide within the apace.

5) A light control system within the plant is most important. It constitutes the energy needed in the plant. Without lights, leaf photosynthesis cannot occur, all metabolism activity cannot happen, and the average growth cannot be followed. The light resources of the plant factories could be classified as artificial light and sunlight. Artificial light is used in the plant system with the sensors and the controllers. Sunburst plants usually need the sunlight coming through the transparent material, e.g., the glass or the plastic cover, that functions as the exterior of the building. In addition, the plant also selects the coating material with heat-insulating and reflected light-absorbing properties. The technologies used in the sunlight plants include heat insulation material, automation equipment, and air conditioning. According to the different light sources, the plants could be classified into three types. They are the sun-use plant factory, a mix of sunlight and artificial light, and artificial light.

   Artificial light sources, such as T5 lamps and LED lamps, come with high electricity efficiencies. In this study, LED is used as the light source. LED has the advantages of easy installation, energy efficiency, no heat generation, and high photosynthetic efficiency. Because chlorophyll prefers light sources from a particular band, choosing different color light sources and the usage frequency of light will affect plant growth.

To monitor the plant factories remotely, cameras are usually installed by the plant factories for managers. The camera could rotate 360 degrees to let managers observe the plant's growth at any time. In addition, to approach the goal of the intelligent plant factory, this monitoring system will record all plant growth data. Big data technology could be applied to advanced processing for the collected data.

The automatic control system is at the kernel of the plant. It deals with all the monitored data and responses to the measurements. For example, when the temperature sensor detects over high, the system will send a cooling process command and turn on the air conditioning. When the temperature decreases to the setting, it turns off the air conditioning. When the temperature drops below the limitation of the

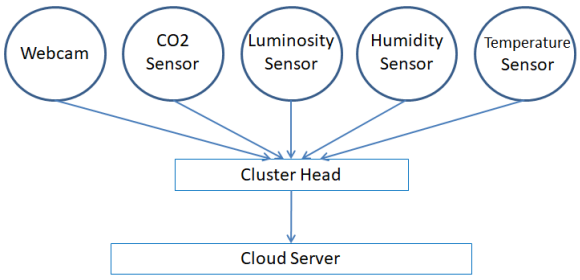


Figure 2: Hu *et al.*'s wireless sensor network of indoor plant factory system architecture [9]

setting, the system will send a heating process command and turn on the heater. Similarly, the processing in the light control system, the humidity control system, and the water control system is controlled by the automatic control system to approach the optimal growth environment for the plants. Automation has become an indispensable technology in plant factories to meet the control requirement for environmental monitoring.

Module 2 is used for the data collection. In this module, we will install various wireless sensors and a webcam to collect data from the plant factory. The elements include temperature sensors, humidity sensors, luminosity sensors, and CO2 sensors. The webcam could be used to observe plant growth at any time from different perspectives. These sensors and webcam technologies are integrated with the sensors and the automatic control devices. The front sensor detects the temperature, humidity, light, water, oxygen, and another plant environment at any time. When it changes to set the environment parameters, it immediately starts the automatic control system to adjust. If the environment has not changed significantly, the sensing data is regularly transmitted to the cloud server and accumulated as a vast database [19, 26, 38, 40]. It could be used for cultivation reference. Also, it could be used for production resumes. The proposed system architecture of the wireless sensor network is shown in Figure 2.

Module 3 is used for information transmission. In this module, we construct a wireless sensor network to transmit the sensor's data. We also have a cloud system to collect all sensor data.

Module 4 is used for the data analysis. In this module, we analyze big data technology and the artificial intelligence process. When the cloud server has collected the various sensing data from sensors, the cloud server will analyze these data and obtain the optimization for the plant growth. Finally, these parameters will be returned to the automatic control system in Module 1 to provide the best growth environment.

3 The Federated Learning

Federated learning is a developing technology, so there are many attacks against it. Common ones include privacy attacks and computer virus attacks. To increase the security of federated learning and encourage data owners to participate in the training of federated learning, federated learning is often combined with blockchain technologies. The security of federated learning can be enhanced through the anti-tampering and decentralization of the blockchain and privacy. However, a large amount of data transmission in the blockchain will cause delays, which may cause data fork problems.

With the rise of privacy awareness, federated learning has gradually emerged [6]. The passage of GDPR, the strictest privacy law in history, is undoubtedly a big blow to companies like Google that use data collection. Therefore, Google proposed the concept of federated learning and the algorithm of federated training (Federated Averaging) in 2016 [31]. First of all, in each round of training, the central server sends a determined global model to the participating users in this round. After the participating users get the model, they conduct model training based on their local data (for example, the typing records of the users of each device using the input method, etc.). Next, calculate the gradient through SGD and update the model parameters. After the model converges, it is sent back to the central server. After the central server receives the models updated by participating users, these parameters are averaged to generate a new model. Finally, the central server returns the updated model to each user, and each user updates the model. Yang *et al.* proposed a federated learning method to improve Google's input method [33].

Chen *et al.* studied privacy issues from the perspective of models [4]. Mothukuri *et al.* explained in detail the concept of the operation mode of federated learning and the classification of federated learning (vertical, horizontal, and migration federated learning) [17]. It also mentioned security issues, privacy issues, and various attack methods of federated learning. It explains in detail the threats to various attack methods and how to defend against them. And put forward various privacy issues that federated learning will face in the future and the results of comparison with decentralized machine learning. Finally, Li *et al.* proposed the prospect of federated learning and the possible crisis and introduced the mechanism and application of federated learning [13].

Regarding the research on the privacy of federated learning, the standard privacy attack is an inference attack. Shokri *et al.* studied the privacy leakage of members in inference attacks of federated learning [22]. Salem *et al.* proposed the first effective defense mechanism against this broader class of membership inference attacks [20]. This membership attack preserves the usefulness of machine learning models. Nasr *et al.* proposed a white-box reasoning attack for deep learning, which designed a white-box reasoning attack to conduct a comprehensive privacy analysis of the deep learning model [18]. Privacy

leakage is measured by the parameters of the fully trained model and by the parameter updates of the model during training. Aiming at passive and active inference attackers, a federated learning inference algorithm is designed, and different prior knowledge of opponents is assumed. And choose membership inference attack as the basis of deep learning model privacy analysis. This method allows us to analyze how to solve the privacy attack problem more efficiently. Since federated learning requires each client to perform calculations and then upload, it is prone to client-side attacks and challenging to identify. Therefore Song *et al.* proposed a framework combining GAN with a multi-task discriminator called multi-task GAN-assisted identification (mGAN-AI) [23]. It can simultaneously distinguish the input sample category, real-world situations, and customer identities. A novel distinction of client identity enables generators to recover user-specified private data to identify attackers. Zhao proposes a novel poisoning defense generative adversarial network (PDGAN) to defend against persistent attacks [39]. PDGAN can reconstruct training data from model updates and audit each participant's model's accuracy using the generated data. Participants whose accuracy is below a predefined threshold are identified as attackers, and the attacker's model parameters are removed from the training process in this iteration.

Chen *et al.* proposed a new user-level reasoning attack mechanism in federated learning [4]. Using the generated adversary network inclusion method can increase the richness of the final member inference model training set, which can be used in single-label and multi-label cases. Guowen Xu *et al.* designed a novel solution to reduce the negative impact of irregular users on training accuracy, thereby ensuring that the training results mainly come from the contribution of high-quality data, thereby improving the system's robustness [30].

Regarding the security research of federated learning, the common one is poisoning attacks. Chen *et al.* studied data poisoning attacks in decentralized machine learning [5]. Biggio *et al.* first proposed the concept of data poisoning attacks against ML algorithms in 2012 [1]. Attackers target vulnerabilities in support vector machine algorithms and try incorporating malicious data points during training to maximize classification errors. Since then, various approaches have been proposed to mitigate data poisoning attacks of ML algorithms in different environments.

The FL environment enables clients to participate in training data and send model parameters to the server. But it allows malicious clients to corrupt the global model by manipulating the training process. Data poisoning in FL is defined as generating dirty samples to train a global model in the hope of generating falsified model parameters, sending them to the server, and designing a secure and resilient Distributed Support Vector Machines (DSVM) algorithm in adversarial environments where attackers can manipulate training data to achieve their goals [37]. They developed a game-theoretic approach to capture the conflict between an opponent and a set of distributed data-processing units. Nash equilibrium in this game method can predict the outcome of learning algorithms in adversarial environments and enhance the resilience of machine learning through dynamic distributed learning algorithms. Zhang *et al.* studied poisoning attacks in generative adversarial networks. By using generative adversarial networks to perform poisoning attacks on alliance learning models, many experiments were done to prove the feasibility of their attacks [35].

This research will design federated learning based on a blockchain-based intelligent plant factory system to solve the problems above of federated learning security and privacy attacks. Because of the decentralized nature of the blockchain [32], Kim *et al.* proposed a blockchain federated learning (BlockFL) architecture by utilizing the blockchain [11]. In this architecture, locally learned model updates can be exchanged and validated. On-device machine learning can be achieved without any centralized training material or coordination by leveraging the consensus mechanism in the blockchain. However, the blockchain has a large data transmission, so it is prone to delay problems. Seike analyzes the end-to-end latency model of blockchain federated learning and describes the optimal block generation rate by considering communication, computation, and consensus latency [21]. Kim *et al.* enhanced the performance of federated learning blockchains through lightweight encryption methods [12]. Martinez *et al.* and Yu *et al.* used the blockchain reward mechanism to promote data owners to participate in federated learning training and increase the accuracy of the global model [34]. Wang *et al.* propose a framework with two perturbation methods for differentially private collaborative filtering to prevent the threat of inference attacks against users [25].

4 The Proposed Privacy and Security of Federated Learning for Intelligent Plant Factory Systems

Machine learning can predict results or obtain important data features through training data. However, the concept of privacy has gradually increased, and information has become unable to be freely shared and utilized. Therefore, Google proposed federated learning. This allows multiple users to jointly train a model without knowing each other's data, which can break the problem of data islands. However, since federated learning is still under development, many loopholes exist. Many malicious actors launch many attacks out of curiosity or malicious intent. Among them, two common attacks are privacy and security. Privacy attacks are mainly to steal data content, while security attacks are mainly to destroy the correctness and accuracy of the trained model. This paper proposes federated learning for intelligent plant fac-

tory systems to address privacy and security attacks.

Figure 3 is the proposed framework of the indoor plant factory system architecture. Intelligent plant factories must continuously collect sensing data from other factories to update parameters in real time and obtain the best plant growth environment. The cluster head will first collect all sensing data (temperature, humidity, luminosity, CO2, the plant growth) of each intelligent plant factory system. Then all cluster heads send their sensing data to a cloud server for training and classification by deep learning. However, if the collected sensing data is forged or collected, it will affect the performance of machine learning. In this paper, after using the federated learning technology to train the local data in each region, upload the gradient loss and generate blocks in the blockchain. Finally, verify and aggregate it into a complete model trained with all the data instead of directly uploading the local data to ensure privacy and data security.

4.1 The Proposed Privacy of Federated Learning for Intelligent Plant Factory Systems

Nasr *et al.* proposed white-box inference attacks against deep learning [18]. White-box inference attacks measure privacy leakage through the parameters of a fully trained model and the parameter updates of the model during training. This research is based on the white-box inference attack as the defense goal [18], and the white-box inference attack uses gradient correction as the primary reasoning tool. Therefore, the method of this study addresses this problem of gradient leakage.

To solve the problem that federated learning is vulnerable to reasoning attacks by attackers, this study designs a method to hide part of the gradient correction value to reduce the attack information obtained by attackers. In addition, this study uses differential privacy to add noise [18]. That is to add randomized noise to the uploaded gradient correction amount to cover up the actual data and increase the difficulty for attackers in inference attacks [25].

1) Conceal part of the gradient correction amount:
This study hides some gradients to enhance the Privacy of federated learning. However, the accuracy rate will decrease while hiding a part of the gradient. Therefore, it is necessary to balance the hidden gradient amount and accuracy and design experiments to test its accuracy. Finally, strike the best balance between accuracy and Privacy to find the best-hidden ratio.

2) Add noise using differential privacy
This project will use Laplace noise to add noise. Since the mathematical properties of the Laplace distribution coincide with the definition of differential privacy, this noise is used in many researches and applications. For a randomization algorithm S, the two output distributions obtained by acting on two adjacent data sets (D & D') are indistinguishable. The formal definition of differential privacy is:

$$PrS(D) \in R \leq e^{\epsilon} \cdot PrS(D') \in R$$

The probability of obtaining a particular output R should be about the same if the algorithm is applied to any adjacent data set. Then we say that this algorithm can achieve the effect of differential privacy. That is to say; it is difficult for observers to detect a slight change in the data set by observing the output results to protect privacy.

4.2 The Proposed Security of Federated Learning for Intelligent Plant Factory Systems

This research addresses the security attacks of federated learning, such as poisoning attacks, backdoor attacks, and so on [14]. The attack most likely to occur in FL is called a poisoning attack. Because every client in FL can access the training data. Therefore the possibility of adding tampered material weights to the global ML model is very high. Poisoning can occur during the training phase and affect the performance and accuracy of the training dataset or the local model, indirectly tampering with the global ML model. A large number of clients perform model updates in FL. That is, the likelihood of a poisoning attack from one or more customers' training materials is high, and the severity of the threat is high. Poisoning attacks are divided into (1) data poisoning, (2) data tampering, and (3) model poisoning.

This study utilizes the defensive approach mentioned in Biggio *et al.* method [1] for federated learning in intelligent plant factory systems. And based on these defense methods, design a new defense method combined with blockchain. The federated learning of the intelligent plant factory system combined with the blockchain can effectively prevent data from being tampered with or injected through the blockchain's immutable and effective verification mechanism.

Federated learning combined with blockchain can effectively solve the low participation rate of data owners and security attacks in federated learning through the immutability of blockchain, an effective verification mechanism, and a quantified reward mechanism (virtual currency). However, because of the distributed node structure of federated learning, there will be a lot of transfer work, and these transfer jobs will generate a lot of delay. Furthermore, it includes upload and download, model update, and cross-validation, so it is prone to blockchain forks. An end-to-end latency model for consortium blockchains is formulated to address this issue by considering communication, computation, and proof-of-work (PoW) latency [21]. This study designs lightweight encryption for transmission, and by adjusting the block generation rate (i.e., PoW difficulty), finds the minimum delay, and then conducts experiments with various

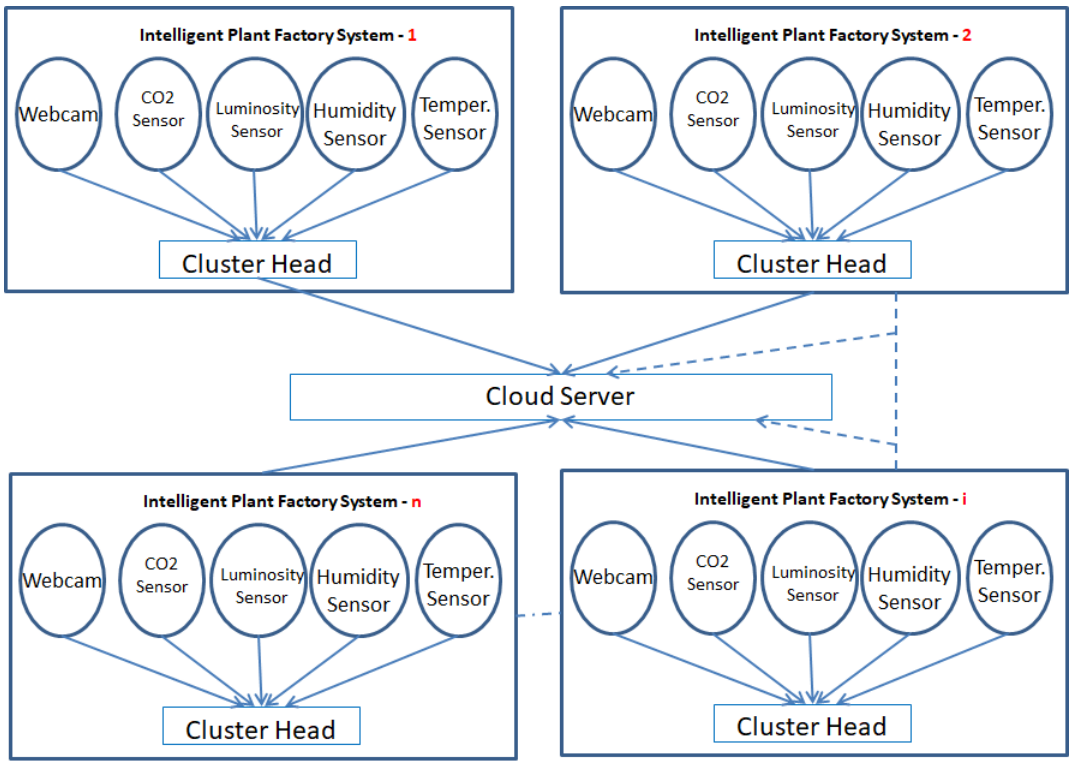


Figure 3: The proposed framework of the indoor plant factory system architecture

lightweight encryption transmission methods and compares them to prove that it can reduce delay in transmission. The research method is as follows:

1) Use Hyperledger to build a virtual environment for federated learning.

2) Design a lightweight encryption method and adjust the block generation rate in the blockchain to find the optimal delay rate.

3) Experiment and compare various lightweight encryption systems, and select the lightweight encryption system with the best delay rate.

5 Conclusion

This paper proposes the privacy and security of federated learning for intelligent plant factory systems. There are three main contributions as follows:

1) This research develops a method to hide some gradient corrections so that node users can reduce the risk of data being derived from inference attacks when uploading gradient corrections. In addition, differential privacy can increase the difficulty for attackers in reasoning attacks. This research can improve the data security and privacy of data owners and can increase the willingness of data owners to participate.

2) This research develops a blockchain-based defense method to defend against poisoning attacks and can effectively strengthen the security of models and data to prevent tampering attacks. In addition, it has dramatically improved the reliability of the model.

3) This study applies blockchain technology to design a federated learning blockchain system that can enhance privacy and security and apply it to the intelligent plant factory system to ensure the correctness of data. The traditional central server is not entirely trustworthy and safe, and using blockchain can build models more safely and improve the security of data and models.

Acknowledgments

The Ministry of Science and Technology partially supported this research, Taiwan (ROC), under contract no.: MOST 109-2221-E-468-011-MY3, MOST 108-2410-H-468-023, and MOST 111-2622-8-468-001 -TM1.

References

[1] B. Biggio, B. Nelson, P. Laskov, "Poisoning attacks against support vector machines," *arXiv: 1206.6389v3*, 2012.

[2] E. F. Cahyadi, C. Y. Yang, N. I Wu, and M. S. Hwang, "The study on the key management and billing for wireless sensor networks," *International Journal of Network Security*, vol. 23, no. 6, pp. 937-951, 2021.

[3] L. Cao, Y. Zhang, M. Liang, and S. Cao, "An improved user identity authentication protocol for multi-gateway wireless sensor networks," *International Journal of Network Security*, vol. 24, no. 4, pp. 713-726, 2022.

[4] J. Chen, J. Zhang, Y. Zhao, H. Han, K. Zhu, B. Chen, "Beyond model-level membership privacy leakage: An adversarial approach in federated learning," in *29th International Conference on Computer Communications and Networks (ICCCN'20)*, 2020.

[5] Y. Chen, Y. Mao, H. Liang, S. Yu, Y. Wei, S. Leng, "Data poison detection schemes for distributed machine learning," *IEEE Access*, vol. 8, pp. 7442–7454, 2020.

[6] Council of the European Union, *General Data Protection Regulation*, Feb. 28, 2023. (`https://gdpr-info.eu/`)

[7] R. H. Dong, B. B. Ren, Q. Y. Zhang, and 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.

[8] R. H. Dong, H. H. Yan, and Q. Y. Zhang, "An intrusion detection model for wireless sensor network based on information gain ratio and bagging algorithm," *International Journal of Network Security*, vol. 22, no. 2, pp. 218-230, 2020.

[9] W. P. Hu, C. B. Lin, C. Y. Yang, M. S. Hwang, "A framework of the intelligent plant factory system," in *Procedia Computer Science: The 8th International Congress of Information and Communication Technology (ICICT'18)*, vol. 131, pp. 579-584, 2018.

[10] M. Hussain, J. Ren, and A. Akram, "Classification of DoS attacks in wireless sensor network with artificial neural network," *International Journal of Network Security*, vol. 22, no. 3, pp. 542-549, 2020.

[11] H. Kim, J. Park, M. Bennis, S. L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279-1283, 2020.

[12] Y. J. Kim, C. S. Hong, "Blockchain-based node-aware dynamic weighting methods for improving federated learning performance," in *20th Asia-Pacific Network Operations and Management Symposium*, 2019.

[13] T. Li, A.K. Sahu, A. Talwalkar, V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process*, vol. 37, no. 3, pp. 50–60, 2020.

[14] J. Lin, M. Du, J. Liu, "Free-riders in federated learning: Attacks and defenses," *arXiv: 1911.12560*, 2019.

[15] Y. C. Lu and M. S. Hwang, "A cryptographic key generation scheme without a trusted third party for access control in multilevel wireless sensor networks," *International Journal of Network Security*, vol. 24, no. 5, pp. 959-964, 2022.

[16] T. Maitra, R. Amin, D. Giri, P. D. Srivastava, "An efficient and robust user authentication scheme for hierarchical wireless sensor networks without tamper-proof smart card," *International Journal of Network Security*, vol. 18, no. 3, pp. 553-564, 2016.

[17] V. Mothukuri, R. M. Parizi, S. Pouriyeh, *et al.*, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619-640, 2021.

[18] M. Nasr, R. Shokri, A. Houmansadr, "Comprehensive privacy analysis of deep learning, passive and active white-box inference attacks against centralized and federated learning," in *IEEE Symposium on Security and Privacy (SP'19)*, pp. 739-753, 2019.

[19] R. Qiu, Y. Fu, J. Le, F. Zheng, G. Qi, C. Peng, and Y. Li, "A software-defined security framework for power IoT cloud-edge environment," *International Journal of Network Security*, vol. 24, no. 6, pp. 1031-1041, 2022.

[20] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, M. Backes, "ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv: 1806.01246*, 2018.

[21] H. Seike, Y. Aoki, N. Koshizuka, "Fork rate-based analysis of the longest chain growth time interval of a PoW blockchain," in *IEEE International Conference on Blockchain*, pp. 253-260, 2019.

[22] R. Shokri, M. Stronati, C. Song, V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE Symposium on Security and Privacy (SP'17)*, pp. 3–18, 2017.

[23] M. Song, Z. Wang, Z. Zhang, *et al.*, "Analyzing user-level privacy attack against federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2430–2444, 2020.

[24] Z. Tian, W. Ma, Q. Yang, F. Duan, "Application status and challenges of machine vision in plant factory — A review," *Information Processing in Agriculture*, vol. 9, no. 2, pp. 195-211, 2022.

[25] J. Wang, A. Wang, "An improved collaborative filtering recommendation algorithm based on differential privacy," in *IEEE 11th International Conference on Software Engineering and Service Science (ICSESS'20)*, pp. 310-315, 2020.

[26] K. Wang, J. Guo, "Secure search over encrypted enterprise data in the cloud," *International Journal of Network Security*, vol. 25, no. 1, pp. 103-112, 2023.

[27] X. J. Wang, M. Z. Kang, U. Lewlomphaisarl, J. Hua, H. Y. Wang, "Optimal control of plant growth in a plant factory using a plant model," in *Australian and New Zealand Control Conference*, pp. 166-170, 2022.

[28] B. Wu, "Stable transmission algorithm for 5G wireless sensor networks based on energy equalization-delay reduction mechanism," *International Journal of Network Security*, vol. 24, no. 3, pp. 428-435, 2022.

[29] H. Xie, Y. Yan, T. Zeng, "Simulations of fuzzy PID temperature control system for plant factory," *Lec-*

ture Notes in Electrical Engineering, vol. 942, pp. 1089-1099, 2022.
[30] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, R. Deng, "Privacy-preserving federated deep learning with irregular users," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1364-1381, 2022.
[31] Q. Yang, Y. Liu, T. Chen, Y. Tong, "Federated machine learning: Concept and applications," *arXiv: 1902.04885*, 2019.
[32] S. Yang, B. Ren, X. Zhou, L. Liu, "Parallel distributed logistic regression forvertical federated learning without third-party coordinator," *arXivpreprint arXiv: 1911.09824*, 2019.
[33] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, F. Beaufays, "Applied federated learning: improving google keyboard query suggestions," *arXiv: 1812.02903*, 2019.
[34] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, Q. Yang, "A sustainable incentive scheme for federated learning," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 58-69, 2020.
[35] J. Zhang, J. Chen, D. Wu, B. Chen, S. Yu, "Poisoning attack in federated learning using generative adversarial nets," in *18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications & 13th IEEE International Conference on Big Data Science and Engineering (TrustCom'19/BigDataSE'19)*, pp. 374–380, 2019.
[36] P. Zhang, D. Li, "EPSA-YOLO-V5s: A novel method for detecting the survival rate of rapeseed in a plant factory based on multiple guarantee mechanisms," *Computers and Electronics in Agriculture*, vol. 193, 2022.
[37] R. Zhang, Q. Zhu, "A game-theoretic approach to design secure andresilient distributed support vector machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5512-5527, 2018.
[38] Y. Zhang, "Research on network security intrusion identification and defense against malicious network damage in a cloud environment," *International Journal of Network Security*, vol. 24, no. 3, pp. 551-556, 2022.
[39] Y. Zhao, J. Chen, J. Zhang, D. Wu, J. Teng, S. Yu, "PDGAN: A novel poisoning defense method in federated learning using generative adversarial network," in *International Conference on Algorithms and Architectures for Parallel Processing*, pp. 595-609, 2019.
[40] J. Zhu, "Research on secure storage of network data based on cloud computing technology," *International Journal of Network Security*, vol. 24, no. 1, pp. 68-74, 2022.

Biography

Wen-Pin Hu received his B.S. degree in Mechanical Engineering from National Sun Yat-sen University in Kaohsiung City, Taiwan. He completed his M.S. and Ph.D. degrees in the Institute of Biomedical Engineering at National Cheng Kung University in Tainan City, Taiwan. He was a visiting scholar at the University of Washington in Seattle, USA, from August 2005 to March 2006. His research interests include biosensors, biomolecular simulations, aptasensors, biomechanics, the internet of things and more. Recently, his research has focused on the refinement of aptamers using computational approaches, the application of aptamers in sensors, and pressure sensing insoles for gait analysis.

Chin-Bin Lin received his Bsc. in Department of Agricultural Management from National Chiayi Institute of Agriculture, Taiwan, in 1992, and M.S. degree in Horticulture from National Chiayi University, Taiwan, in 2008. He is currently pursuing a Ph.D. degree in the Department of Bioinformatics and Medical Engineering, Asia University, Taiwan. His research interests include Intelligent Plant Factory System, Horticulture, and Bioinformatics.

Jing-Ting Wu received her Bsc. in Department of Business Administration, National Changhua University of Education, Taiwan, in 2020, and and M.S. degree in Department of Management Information Systems, NCHU, in 2022. Her research interests include Federated Learning, Deep Learning, and Information Security.

Cheng-Ying Yang received the M.S. degree in Electronic Engineering from Monmouth University, New Jersey, in 1991, and Ph.D. degree from the University of Toledo, Ohio, in 1999. He is a member of IEEE Satellite & Space Communication Society. Currently, he is employed as a Professor at Department of Computer Science, University of Taipei, Taiwan. His research interests are performance analysis of digital communication systems, error control coding, signal processing and computer security.

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 distinguished professor and Chairman of the Department of Management Information Systems, National Chung Hsing University, during 2003-2011. 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 (AU), Taichung, Taiwan. He is currently a chair professor with the Department of Computer Science and Information Engineering, AU. 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.

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.