Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Construction of Emergency Response Mechanism and Data Protection System in Network Security

Xibao Wang[1], Hengming Yuan[1], Yueyue Bu[2], Wenhui Chu[1], Chengcheng Gao[1], Lei Wang[1,✉]

[1] *Industrial Internet Business Division, Yunding Technology Co., Ltd., Jinan, Shandong, 250000, China*
[2] *Integrated Machine Parts Department, Yankuang Energy Group Co., Ltd., Material Supply Center, Jining, Shandong, 250000, China*

ABSTRACT
In order to improve the accuracy of automatic detection of malicious code, this paper focuses on the "texture" features of malicious code and the characteristics of different types of malicious code, which are also different, and uses them for the automatic detection of unknown malicious code by using the four machine learning algorithms of KNN, RF, NB and SVM to perform single-feature detection and multi-feature (GLCM, LBP and ngram feature merging) detection respectively. Four machine learning algorithms, namely KNN, RF, NB, and SVM, are used to perform single-feature detection and multi-feature (GLCM, LBP, and n-gram feature merging) detection respectively, and analyze the accuracy of the spatial relationship feature-oriented malicious code detection scheme. A multi-version oriented data protection model is proposed for the data storage space, data version, quantity management and recovery requirements involved in service emergency response. The relative performance errors between its data protection scheme and the plaintext scheme and the simple add noise scheme are analyzed. In all four machine learning algorithms, the detection rate of fused features is higher than that of single features, and the maximum difference can reach more than 60%. When ε takes the value of 9 or 3, the data privacy protection algorithm, the plaintext algorithm, and the noise-only addition algorithm in this paper have similar accuracy rates. With proper noise selection, this paper's scheme has good performance in real simulation.

Keywords: multi-feature, malicious code, automatic detection, data protection model, emergency response

1. Introduction

Cybersecurity is one of the most important issues that cannot be ignored in today's society. As the popularity and dependence on the Internet increase, cybersecurity incidents have become

✉ Corresponding author.
E-mail address: 15254106762@163.com (L. Wang).
Received 10 January 2024; Revised 20 February 2024; Accepted 10 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-446

increasingly frequent and complex. In order to cope with these incidents, organizations and institutions have not only established emergency response mechanisms but also constructed data protection systems to cope with cybersecurity threats and protect sensitive data and information security [1-4].

Emergency response mechanism means that when a cybersecurity incident occurs, all relevant parties react quickly and take appropriate measures to prevent, mitigate, or restore the damage caused by the cybersecurity incident in accordance with the pre-established plans and procedures [5-7]. Its purpose is real-time monitoring, early warning, rapid disposal and late rectification to maximize the protection of network and system security. And the construction of network security system should start from the measures to prevent network attacks, including network security infrastructure, network security institutional mechanism, network security technology means [8-11]. First of all, in terms of network security infrastructure, a strong information security management system should be established and a perfect security management system should be constructed. Through the implementation of strict security review and technical application control, real-time monitoring of information security is achieved to prevent possible network attacks [12-15]. Secondly, in terms of network security mechanism, it should establish a full-time network security institutions, and clarify their respective responsibilities and tasks. Especially for large-scale new organizations, enterprises and government departments, they should set up full-time network security protection departments and establish professional network security protection organizations [16-18].

This paper organizes the characteristics and types of cybersecurity incidents, points out the four aspects of remediation of cybersecurity and the four major countermeasures. It breaks down the emergency response linkage process of cybersecurity incidents, and analyzes the main operational steps and positioning of the six stages of preparation, detection, suppression, eradication, recovery and tracking. Aiming at the unique "texture" characteristics of malicious code samples in terms of spatial relationship, the multi-feature-based malicious code detection technology is proposed for its special "texture" nature. Using KNN, RF, NB, SVM machine learning algorithms for malicious code feature detection, we verify the feasibility of the holistic inspection scheme based on the texture characteristics of malicious code proposed in this paper. Design multi-data version of the protection model for performance analysis.

2. Emergency response mechanisms and data protection models

2.1. Cybersecurity emergency management mechanism

Network security means that the network system, the hardware and software of the terminal and the data in its system are protected from being changed or leaked due to malicious damage. Continuous, reliable and normal operation of the system and uninterrupted network services are, by their very nature, information security on the network [19-20].

Emergency response usually refers to the measures taken by an organization after the occurrence of various unexpected and unforeseen events with the aim of reducing hazards and losses, as well as recovering from them.

2.1.1. Classification and characterization. Network security incidents occur in a variety of ways, according to the threat caused by the incident can be broadly categorized into five categories: physical threats, malicious code threats, illegal invasion threats, denial-of-service attacks, and subjective and unintentional errors.

Network security incidents are based on the technical development of the Internet, so a prominent feature of network security incidents is "strong technology". In addition, because the network is virtual, whether it is office, trading or games, chatting, are not face-to-face, so its other two distinctive features are "covert" and "high degree of virtual". Therefore, security incidents that occur on the network are

characterized by “high technology”, “virtualization”, “covert”. In contrast, most of the general security events that occur in the physical world can be observed directly, and do not have the characteristics of virtualization.

2.1.2. Development trends. The core element of cybersecurity emergency management is cybersecurity incident emergency response, but that is not all. Doing management must consider all aspects of cybersecurity emergency response, such as cybersecurity culture construction, cybersecurity incident prevention, security quality management, cybersecurity risk assessment, cybersecurity design and implementation.

To comprehensively manage the Internet network security from four aspects: legal norms, administrative supervision, industry self-regulation and technical security.

1) Strengthen the construction of network security culture.

2) Scientific planning, effective integration, streamlining and integration of government portal resources.

3) Adhere to active defense, and build a globalized and front-loaded active emergency protection system.

4) Actively explore and conduct in-depth research to maintain the sustainable development of informatization.

2.1.3. Emergency response mechanism countermeasures:

1) Raising information risk awareness and cultivating an information security culture. Risk awareness is the degree of individual knowledge of risk, and learning scientific risk culture and improving the level of social awareness of risk are important cornerstones for maintaining healthy and sustainable development of society.

At the ideological level, it is necessary to use scientific knowledge to understand risk. Recognize the universality of the risk phenomenon, and be calm and organized in front of network security events.

2) Improve the organizational mechanism and establish a one-master, multi-dimensional governance model. The development of network technology has made the side effects of institutional “fragmentation” more and more obvious. A series of problems such as institutional decentralization, difficulties in departmental coordination, and low efficiency of public services require the government to move from part to whole and from fragmentation to integration. Therefore, a sound organizational mechanism is the organizational basis for achieving network information security governance.

3) Improve relevant regulations to form a standardized governance environment. It is embodied in the following three steps to promote the implementation of information security regulations. Make up for the regulatory shortcomings of network service organizations. Strengthen the self-regulatory norms of the information security industry.

4) Optimize the operation mechanism and establish a composite risk prevention operation mode

The social risk of the risk society is characterized by composite and suddenness, and in the information age of rapid technological development, there is a possibility of infinite amplification of risk. Through the platform of rapid dissemination, it radiates to all areas of social life, forming a comprehensive impact effect of risk. The composite and sudden nature of non-traditional threats is characterized by the requirement of network information security governance to break through a single governance model, the establishment of a composite risk prevention operating model. Network information operation mechanism is a mechanism to deal with network information security issues on a regular basis. Only by establishing a perfect risk prevention operation mechanism can network information security events be handled efficiently.

2.2. Emergency Response Linkage for Cyber Security Incidents

2.2.1. Preparation phase. Stage 1 is the preparation stage, i.e., preparing in advance for all possible scenarios. In the whole incident response methodology, this phase has the most important significance for maintaining network security.

The specific work to be done in the preparation phase includes at least five aspects:

1) Conducting risk assessment
2) Developing security policies and establishing security defenses/controls
3) Establishing an emergency response plan
4) Prepare emergency response personnel and resources
5) Establishing a platform to support emergency response

2.2.2. Detection phase. Phase 2 is the detection phase. The first thing that needs to be made clear is that detection is not the same as intrusion detection. Detection has a much larger scope than IDS. Simply put, the task of this phase is to collect information. Not only does it collect information to determine whether an event has occurred and what the event is related to, but it also collects important information about the victim and the victimized system or network. From an operational point of view, all the operations of the incident response process rely on detection, which actually triggers the emergency response.

The specific tasks in the detection phase include the following:

1) Detection software
2) Initial response
3) Estimation of incident development
4) Reporting

2.2.3. Suppression phase. Stage 3 of the response-suppression aims to limit the scope of the impact of the incident and the tendency for it to worsen. It builds on the detection phase. The importance of containment is to prevent the security incident from getting out of control quickly. It is usually accomplished by the victim's network administrators. This is because it is prudent for unsure victims to refrain from immediate, self-perceived eradication until a professional response team arrives, as this can often lead to more serious consequences if it is botched.

2.2.4. Eradication phase. The fourth stage is the eradication stage, i.e. finding the root cause of the incident and removing it completely.

Usually, this stage requires the use of software, and we have prepared various security tools in the preparation stage, which should be used to thoroughly clean the system at this point. It is better to format the files infected by the malicious program and replace them with clean backups.

It should be noted that if the victim is a specialized system, then the eradication operation must follow special rules of operation. For confidential environments, low-level formatting is a bit more thorough.

The eradication phase varies from system to system and network to network, and both the victim and the responders should be prepared to follow the established process of eradication in the normal course of events. And it is always important to keep the appropriate people and organizations informed of any progress.

2.2.5. Recovery phase. The fifth phase is recovery, the goal of which is to restore all systems and network environments affected by the incident to a normal operating state. Backup mechanisms play an important role here.

The recovery process varies for different systems and network environments and for different organizations, but there are credibility considerations, and it is important to use full or incremental

backups that are confidently clean. The process should be based on recovery guidelines developed during the preparation phase or for specialized systems.

2.2.6. Follow-up phase. The final phase is tracking, the goal of which is to review and consolidate information about what happened. This phase is critical to the successful completion of the response mission, the provision of information useful for legal action, and the refinement of the responders' knowledge, body of experience, and response capabilities.

The most important aspect of the tracking phase is to dedicate sufficient effort and time to conduct an after-action review of the incident to analyze the technical, process, and other aspects of the incident and the response.

2.3. Automatic Detection of Unknown Malicious Code with Multiple Features

The malicious code detection method proposed in this paper is not bound to the code details of the malicious code samples themselves, and does not require disassembling the malicious code, as well as detecting it in the executed state. The spatial relation characteristics of the malicious code samples reflect the connection between its internal functional blocks [21]. In this paper, from the spatial relationship characteristics of malicious code, we consider the unique "texture" characteristics of malicious code as a whole, and use it in the automatic detection of unknown malicious code, which achieves good detection results.

2.3.1. Design and Algorithms. The method of extracting the spatial relation features of malicious code in this paper is described as follows: firstly, the malicious code sample is automatically chunked, the spatial relation regions contained in the sample are delineated, and the features of the malicious code sample are extracted and indexed for each region. The mathematical description of the chunking of malicious code samples is as follows.

With the help of the concept of set, the chunking of malicious code samples is defined as follows.

Definition 1: Let set R represent the entire sample of malicious code, and the chunking of R is also the division of R into a number of non-empty subsets (subregions) $R_1, R_2, \ldots\ldots, R_N$. The subregions need to satisfy the following conditions:

1) $\bigcup_{i=1}^{N} R_i = R$.

2) All i and j, $i \neq j, R_i \cap R_j = \Phi$.

3) For $i = 1, 2, \ldots\ldots, N$, there is $P(R_i) = TRUE$.

4) For $i \neq j$, there is $P(R_i \cup R_j) = FALSE$.

5) For $i = 1, 2, \ldots\ldots, N$, R_i is a connected region.

where the logical predicates of the elements in set R_i are denoted by $P(R_i)$ and the empty set is denoted by Φ.

Condition (1) states: chunking should divide all chunked characters in the malicious code sample into some sub-region.

Condition (2) states that there are no intersections between subregions, i.e., no common elements between subregions.

Condition (3) states that characters with some of the same characteristics should belong to the same region.

Condition (4) states that characters with different characteristics should belong to different subregions.

Condition (5) requires that the region resulting from chunking must be a continuous block.

This algorithm is based on the nature of clustering formed in the sample space of regions with similar code characteristics in the malicious code to intelligently segment the malicious code samples.

2.3.2. Chunked Feature Extraction. In this section, the malicious code samples that have been successfully chunked are extracted with different features such as sequence character moments, information entropy and relevance, and the different feature vectors of the malicious code samples are normalized so as to improve the detection accuracy of the malicious code.

Definition 2: Character moments is a feature extraction method that does not require vectorization of features, and its principle is that the distribution of malicious code samples can all be represented by its moments. The basic form of n st order moments is:

$$n_i = \{[\sum_{i=1}^{N_i}(p_i - \mu_i)^n] / N_i\}^{1/n} \tag{1}$$

In this paper, using the first order moment $\mu_i(Mean)$, second order moment $\sigma_i(Variance)$ and third order moment $s_i(Skewness)$ of the malicious code samples, the distribution of the malicious code samples can be effectively expressed and the sample features of the malicious code can be extracted. Eq:

$$\mu_i = \sum_{i=1}^{N_i} p_i / N_i \tag{2}$$

$$\sigma_i = \{[\sum_{i=1}^{N_i}(p_i - \mu_i)^2] / N_i\}^{1/2} \tag{3}$$

$$s_i = \{[\sum_{i=1}^{N_i}(p_i - \mu_i)^3] / N_i\}^{1/3} \tag{4}$$

where p_i is the character component in the ind sample chunk in the sample. N_i is the number of characters in the sample chunk.

The information entropy measures the degree of dispersion in the distribution of the character sequence of the malicious code sample. The ENT value is larger when the values in the character sequence do not differ much and are more dispersed. Eq:

$$ENT_i = -\sum_{i=1}^{N_i} p_i lb(p_i) \tag{5}$$

The correlation coefficient measures the degree of correlation of the character sequences of a malicious code sample. From there, the degree of correlation of a malicious code sample chunk can be determined. The formula is:

$$COR_i = \frac{\sum_{j=1}^{N_i}(p_j - \mu_i)^*(p_{j+1} - \mu_i)}{\{(\sum_{j=1}^{N_i}(p_j - \mu_i)^2)*(\sum_{j=1}^{N_i}(p_{j+1} - \mu_i)^2)\}^{1/2}} \tag{6}$$

Equation (1) to Equation (6), are the important basis for feature vector extraction in this paper.

2.3.3. Integrated multi-feature weighted similarity matching. Since the unknown malicious code cannot precisely determine the chunk to which the sample feature vector belongs in advance, multi-feature matching is used, followed by confirmation using weighted comprehensive judgment. As a result, it can be determined whether a certain unknown malicious code is a malicious code or not. At the same time, it can also determine the specific type of malicious code it belongs to, thus improving the accuracy of retrieval. The specific calculation process is as follows.

1) All the feature vectors are normalized so that the processing feature vectors take values between [-1, 1] to ensure that all the component values have the same status before weighting.

2) To calculate the distance between feature vectors, the Euclidean distance formula is used:

$$D_Q = \sqrt{\sum_{i=1}^{K} (f_i - \overline{f}_K)^2} \tag{7}$$

3) The calculated result is noted as $D_{1Q}, D_{2Q}, \ldots D_{NQ}$ to which a Gaussian normalization is applied. so that the distance between the samples falls on the interval [-1, 1]. Eq:

$$D_{iQ}^{(N)} = \frac{D_{iQ} - \mu_Q}{3\sigma_Q} \tag{8}$$

4) The distance D is made to finally fall on [0, 1] by linear translation:

$$D_{iQ}^{(N)'} = \frac{D_{iQ}^{(N)} + 1}{2} \tag{9}$$

Distance $D_Q^{(N)'}$ is able to represent the similarity of different features equally. The smaller the distance, the more similar the feature is.

5) The minimum of N $D_Q^{(N)}$ is taken as the final similarity distance between the malicious code and the sample related feature vector. Eq:

$$D = M_{i=1}^{N} in^{\{D_Q^N\}} \tag{10}$$

6) Repeatedly use the above steps to synthesize the similarity distances of multiple feature vectors to determine whether a particular suspicious code is a malicious code, and also to determine the specific type of malicious code to which it belongs.

7) Weighted matching. The distance vectors of multiple feature vectors are univariate using the weighted average method, in this paper, m represents the dimension of the distance vector, e_i is the corresponding component, and k is mainly used to regulate the influence produced by the larger e_i. The formula is:

$$E = \frac{\sum_{i=1}^{m} ie_i^k}{\sum_{i=1}^{m} e_i^k} \tag{11}$$

2.4. Modeling Multi-Version Data Protection for Service Emergency Response

2.4.1. Model preparation. The target object protected in a multiversion data protection system is a data storage space, which corresponds to actual objects such as disks and file systems. The storage space consists of a finite number of storage units, each of which can correspond to an actual system object such as a bit, byte, sector, data block, file, etc. Without loss of generality, it can be assumed that each storage unit in a storage space has a unique logical address and the entire storage space has a corresponding logical address space.

Storage units and storage space are not static; the contents of the same logical address may change at different moments due to write operations. To describe this dynamic change in the storage system, the time dimension is further introduced.

Definition 3 (Time Dimension) Remembering that the initial moment of multiversion data protection is t_0 and the current moment is t_{cur}, the time dimension is defined as the time span from moment t_0 to moment t_{cur}, denoted as $Time = [t_0, t_{cur}]$. For any moment t in multiversion data protection, there is $t \in [t_0, t_{cur}]$.

Definition 4 (Data version) A data version is the complete data image available in a storage system, denoted as $State$. The data version of a moment t is defined as the state of the storage space at that moment, denoted as S_t, which is expressed as $S_t = \{< addr, val_t > | addr \in Addr \wedge val_t \in Value\}$ using the set approach.

Moment t can also be used as a version number to distinguish the data version at different moments. For example, the initial data version is denoted as S_{t_0} and the current data version is denoted as $S_{t_{cur}}$.

2.4.2. Data versioning model. In a multi-version data protection system, this time-ordered series of versions can be described by the sequence S_1, denoted $S_1 = (S_{t_0}, ..., S_{t_i}, ..., S_{t_{cur}})$.

Definition 5 (Amount of Data Updates) The amount of data updates is the primary measure of the magnitude of the difference information between two data versions in a storage system, denoted as ΔD. The amount of data update ΔD between two data versions $S_{t_i} S_{t_j} (1 \le i, j \le cnt)$ is defined as the symmetric difference between data versions S_{t_i} and S_{t_j}, denoted as $\Delta D(S_{t_i}, S_{t_j}) = S_{t_i} \oplus S_{t_j}$. Combined with the storage unit definition, the data update is expressed using a set approach as:

$$\Delta D(S_{t_i}, S_{t_j}) = \{unit \mid (unit \in S_{t_i} \wedge unit \notin S_{t_j}) \vee (unit \in S_{t_j} \wedge unit \notin S_{t_i})\} \tag{12}$$

The data update quantity ΔD describes the difference between two data versions, so that any two elements of the triad $(S_{t_i}, S_{t_j}, \Delta D(S_{t_i}, S_{t_j}))$ are known to compute the third element.

The data version generation model, i.e., the data version sequence S generation process model.

Definition 6 (Data Version Generation Model) The data version generation model refers to the process of generating the corresponding version sequence S on time series $T = (t_1, t_2, ..., t_n)$, denoted as $generateS$. The data version information to be generated is denoted as S_{t_i}, and the specific process of generating sequence S is as follows:

1) If i equals 1, generate data version S_{t_i} and then add S_{t_i} to sequence S.

2) If $i > 1$, if only the amount of data update needs to be saved, calculate S_{t_i} according to the definition of the amount of data update. The amount of data update $\Delta D(S_{t_{i-1}}, S_{t_i})$ based on $S_{t_{i-1}}$ or $\Delta D(S_{t_{i-2}}, S_{t_{i-1}})$, and then add to the sequence S. If the amount of data version needs to be saved, generate S_{t_i} and add to the sequence S.

2.4.3. Data versioning model. Data versioning in a multi-version data protection system focuses on creating a version organization structure, retrieving specified versions, and deleting outdated versions.

Definition 7 (Data Versioning Structure) Data versioning structure refers to the method of organizing a sequence S of data versions in a multi-version data protection system, denoted $structS$. Each node $node$ in $structS$ maps one-to-one to each data version or data update quantity in S. Notation $node =< key, ver, pointer >$, where key is the keyword used to query for the specified version, ver is a pointer to the data version S_t or data update quantity ΔD, and pointer is used to organize $structS$ the structure pointer.

Definition 8 (Data Versioning Model) A data versioning model is a method of data versioning defined over a version structure $structS$ and a version operation $Operate(structS, x, op)$, denoted $manageS$, where op denotes the type of operation. It is further defined as a binary group $manage\mathrm{S} = (struct\mathrm{S}, Operate(struct\mathrm{S}, x, op))$ of version structures and version operations.

2.4.4. Data versioning recovery model. In multi-version data protection, for data version sequence S, there are two methods of forward recovery and backward recovery to the corresponding data

version S_{t_i} of $\Delta D(S_{t_{i-1}}, S_{t_i})$. In multiversion data protection, the recovery of a specified version S_{t_i} of a data version sequence S can be defined as a data version recovery model.

Definition 9 (Data Version Recovery Model) The data version recovery model refers to the process of recovering a specified version S_{t_i} in a sequence S, denoted as $recoverS$. Data version S_{t_i} is the recovery target, $1 \le i \le cnt$. The recovery process is to recover directly if S_{t_i} corresponds to one of the data versions in S. Otherwise, data version S_{t_i} is recovered using forward version recovery or backward version recovery.

Combining the above definitions, the final multi-version data protection model can be defined as follows.

Definition 10 (Multi-version data protection model) A multi-version data protection model is defined over a data version generation model $generateS$, a data version management model $generateS$, and a data version recovery model $recoverS$, denoted UMV. It is further defined using a ternary as $UMV = (generate\mathrm{S}, manage\mathrm{S}, re\,\mathrm{cov}\,er\mathrm{S})$.

3. Cybersecurity incident-oriented emergency response processing

3.1. Performance Analysis of Malicious Code Detection Algorithms

3.1.1. Data sets. The dataset selected for this paper is provided by Microsoft's Malware Classification Challenge, which has a total of 10054 samples of labeled malicious code. There are two types of files included in this dataset, one is byte file which is the binary executable file of malicious code without PE header. The other category is .asm files, which are files obtained by disassembling the binary executable of malicious code. There are nine types of malicious code in this dataset and they are Ramnit, Lollipop, Kelihos_ver3, Vundo, Simda, Tracur, Kelihos_ver1, Obfuscator.ACY, Gatak.

3.1.2. Feature fusion. This is because the format of each malicious code file after extracting GLCM, LBP and n-gram features is a Dataframe of python data structures. Moreover, the first column of each Dataframe is the file name of the malicious code, and the extracted GLCM, LBP and n-gram features are merged based on the same file name to realize the feature fusion of the malicious code.

3.1.3. Experimental results. Classification comparison is done using k nearest neighbor, random forest, plain bayes and support vector machine after fusion of three single features namely LBP features, GLCM features and n-gram features and above three single features. The accuracy comparison is shown in Fig. 1.

Comparison of accuracy in terms of single features shows that LBP features have the highest accuracy in KNN and Random Forest algorithms, but LBP features have the lowest accuracy in Park Bayes and Support Vector Machine algorithms. Moreover, the dimensions of the three features, LBP features, GLCM features and n-gram features, are 32, 145 and 9800, respectively, and it can be concluded that it is not the longer the dimensions of the features, the higher the accuracy of the classification, and the classification algorithm chosen is also important. It can be seen that no matter which algorithm is used, the fused features are more accurate than the single features. And the fused features are 68.25% higher than the LBP features in the plain Bayesian classification algorithm.

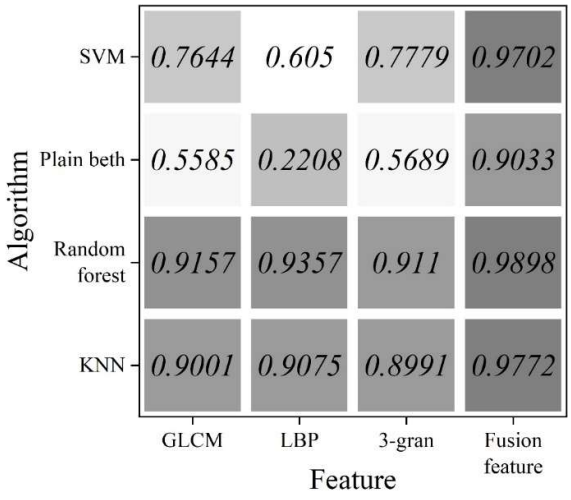


Fig. 1. Accuracy comparison

The most widely used to evaluate the predictive ability of a model is the 2D confusion matrix, which is shown in Table 1.

Table 1. Two dimensional confusion matrix

Real category	Predictive result	
	Class 1 (positive example)	Category 2 (counter example)
Class 1 (positive example)	True Positive/TP	False Negatibe/FN
Category 2 (counter example)	False Positive/FP	True Negatibe/TN

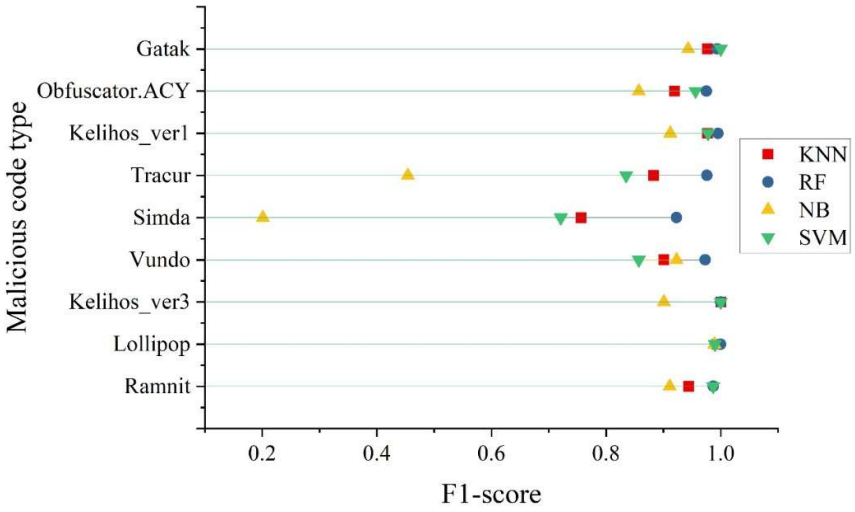
The precision and recall values of the four machine learning classifications are shown in Table 2. It can be seen that the nine malicious code types in the dataset have high recall, precision and f1-score, except for some of them which have less data and insufficient features extracted during training, resulting in low recall, precision and f1-score.

Table 2. The accuracy and recall value of four kinds of machine learning classification

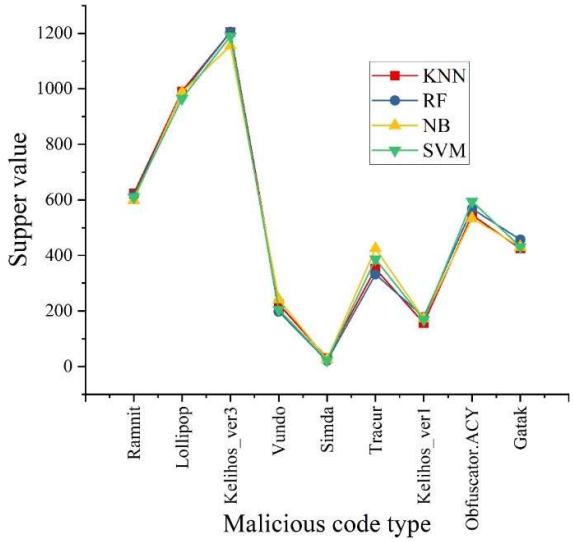
Malicious code type	Precision				Recall			
	KNN	RF	NB	SVM	KNN	RF	NB	SVM
Ramnit	0.936	0.967	0.835	0.945	0.935	0.991	0.942	0.983
Lollipop	1.000	1.000	1.000	1.000	0.991	0.989	0.993	0.987
Kelihos_ver3	0.981	1.000	0.812	1.000	1.000	1.000	0.994	1.000
Vundo	0.897	0.956	1.000	1.000	0.925	0.936	0.838	0.775
Simda	0.723	1.000	1.000	1.000	0.865	0.824	0.153	0.568
Tracur	0.822	0.969	1.000	0.717	0.973	0.972	0.239	0.993
Kelihos_ver1	0.951	0.991	1.000	0.929	0.983	0.993	0.865	0.942
Obfuscator.ACY	0.953	0.987	0.855	0.933	0.868	0.964	0.891	0.925
Gatak	1.000	1.000	0.993	1.000	0.934	0.993	0.876	0.993

The F1 value and supper of the four machine learning classifications are shown in Fig. 2. In Fig. (a), it can be seen that the F1 value of malicious code detection algorithms for the detection of nine kinds of malicious code is distributed in the interval of [0.8,1.0], and only two kinds of codes, Tracur and Simda, have a low F1 value. Figure (b) four machine learning classification algorithms on the malicious

code detection of the SUPPER value curve trend is the same, so this detection method can be very good detection of existing malicious code and have a certain degree of anti-confusion ability.



(a) F1 value



(b) Supper value

Fig. 2. Four kinds of machine learning classification of F1 and supper

3.2. Emergency response performance testing

3.2.1. Test environment. In terms of experimental equipment, the boundary routing switch used in the system is Extream X670, and the OpenFlow switch is H3C S6300. The response task management server is running on an ordinary PC, while the response task execution server researched and designed in this paper is running on a high-performance blade server DELLR 720 with 15 physical cores and 30 logical cores. The response task execution server configuration is shown in Table 3.

Table 3. Respond to the task execution server configuration

Configuring	Specification
Operating system	CentOS release 9.7 (Final) Linux monster 3.7.45-612.8.1.el7.x86-98
CPU	Two processors, Each processor contains eight physical cores, Each physical nucleus has two logical nuclei,Intel(R)Xeon(R)CPU E9-2650 0 @ 5.00GHz
Memory	DDR3 64GB
Card type	Intel Corporation Ethernei 10G 5P X520 Adapter (rev 01)×2
Network driver	Ixgbe 9.1.2-DNA
Disk	SCSI 320GB

3.2.2. Performance testing. The test of system performance on the one hand should test the packet loss rate of forensic collection, and on the other hand, it is necessary to test the utilization rate of resources (CPU, memory and hard disk) in the process of system operation.

The packet loss rate of forensic capture is calculated by comparing the statistics of the number of messages forwarded by the OpenFlow switch with the number of offline messages stored by the forensic capture, and the packet loss rate is calculated by the formula:

$$Acquisition\ rate = 100\% * \left(Forwarding\ number - Collected\ number \right) / Forwarding\ number \tag{13}$$

CPU utilization requires reading fields from /proc/stat, and seven fields of data need to be extracted: user mode (user), low priority user mode (nice), kernel mode (system), idle processor time (idle), disk IO wait time (iowait), hard interrupt time (irq), and soft interrupt time (softirq). The times indicated in the fields are statistics from server startup to the current read file time.

Read the values of the fields in the /proc/stat file between two points in time to calculate the average CPU utilization between the two points in time, which is the quotient of CPU occupancy time (excluding system state occupancy time and idle time) and overall CPU time (excluding system state occupancy time). The calculation of memory utilization requires two data from the /proc/meminfo file, the current empty limit of memory, memFree, and the total memory, memTotal. Memory utilization is the quotient of the size of memory used to the total memory available.

Test 1: A stress test of the response process is performed, in which the CHAIRS system continuously sends out the response tasks for the test, and tests the changes in the system CPU, memory, and collection packet loss rate.

The summary of security incidents in CERNET Education Network for August 2018-2023 is shown in Table 4. The average case detection rate in 2022 and 2023 is 850 security events per hour. Since it takes multiple security events to aggregate into a single case, the stress test will be conducted with 1,200 cases (response tasks) per hour as the peak, i.e., 30 response tasks per minute. For the specific test, an active Class C address segment is used as the monitoring object, and each response task lasts 10 minutes. Each time the forensic message collection (message collection number threshold is set to 20,000 messages) and IDS-related detection operations are performed, and the system's CPU and memory utilization as well as packet loss rate are collected once at every minute interval, lasting for a total of 20 minutes.

Table 4. Summary of security events

	2019	2020	2021	2022	2023	Total
DDOS	256311	213565	60142	1758466	2365414	4653898
Botnet	553905	145965	25246	98693	1896	825705
Backdoor	6457	5536	15	957	8254	21219
DGA Domain name	0	0	0	0	23	23
Non-authorized traffic	18366445	74156093	53166429	5784665	0	151483632
Total	19183118	74521159	53251832	7642781	2375587	156974477

The variation of CPU and memory utilization under stress test is shown in Fig. 3, which gives the variation of CPU, memory utilization of the system at peak speed (20 responses per minute). The first 3 minutes CPU, memory utilization grows because the number of concurrent tasks of the system is not yet occupied, so it will keep increasing. The memory utilization does not exceed 20% for the first 3 minutes. Then when the number of concurrent tasks reaches the threshold, the CPU utilization reaches stability.

In particular, there is a large mutation in CPU and memory utilization at every 5 minutes because of the start of the IDS detection program, which is observed for the 5min, 10min and 15min time points, there is a 300% mutation in CPU and about 20% mutation in memory. In particular, the memory utilization does not drop because of Linux's memory buffering mechanism, where the Linux system caches as much memory as possible to improve read and write performance.

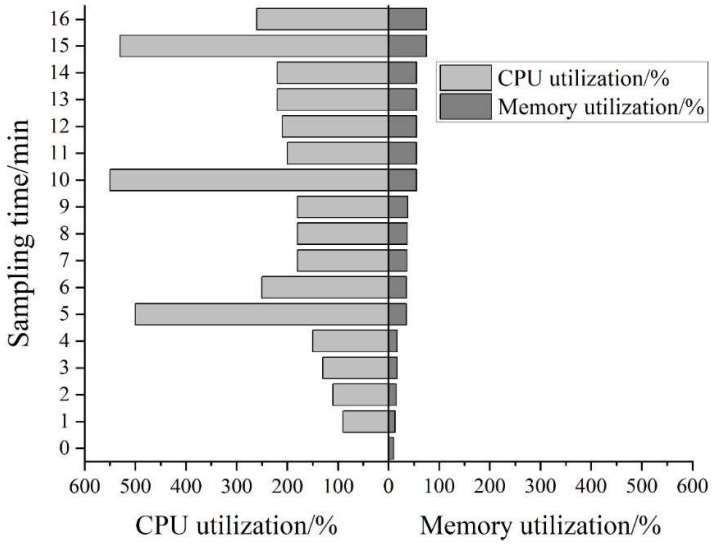


Fig. 3. The CPU and memory utilization changes under the pressure test

The packet loss rate for forensic capture under stress test is taken from 30 response tasks out of 300 response tasks executed and tested for packet loss rate. From the graph, we can see that the packet loss rate is always zero under the stress test.

Test 2: For the 35 cases in which CHAIRS actually completed the response statistics on the average time of execution of a single response task, compared with the preset 308s. Test the overall performance of automated intrusion tracking during actual operation.

The average response time of the actual cases is shown in Figure 4, demonstrating the average response inches of the sampled 35 cases, with the response time fluctuating from 314s to 318s. Due to the different detection objects, the message size of forensic collection will be different. The time spent on simultaneous IDS offline detection and subsequent protocol behavior analysis is also different. As a whole, the response time basically fluctuates around 316s, except for the 308s necessary

for packet acquisition, the rest of the automated response steps are completed in around 8s toward the completion, which meets the timeliness requirements of the case response.

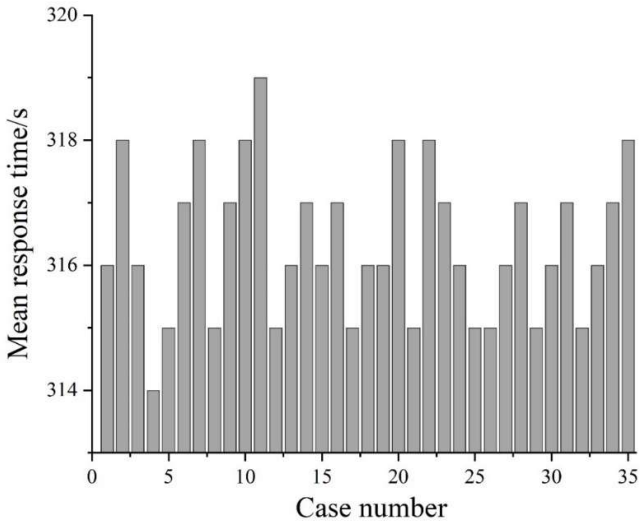


Fig. 4. The average response time of the actual case

4. Joint privacy-preserving model training for multiple data sources

4.1. Experimental conditions and optimization

This simulation scheme uses the MNIST handwritten dataset as the training dataset, where the training data size is about 50 Mb containing about 60,000 samples and the test data size is about 9.5 Mb containing about 10,000 samples.

The dataset data are all 28×28 pixel grayscale images, and the dataset is first divided into two parts with approximate numbers using random sampling. The training set and test set are divided according to the ratio of 6:1 (both parts are divided according to this ratio), and the neural network is set with 1000 hidden neurons. The ReLU function is used as the activation function, and the loss function adopts the cross-entropy function, and the softmax hierarchical determination of ten types of numbers is added.

In the training process, in order to prevent the parameter range is too large, the parameter needs to be regularized. At the same time, batch-Normlization is added at each level to limit the parameter range. In this experiment, the parameter value is restricted to be no larger than 5, and the learning rate is set to be 0.06 initially and linearly decreases to 0.063 in 20 epochs, and remains unchanged in the subsequent epochs. After saving the parameters at the end of the training phase, the parameters were scaled up uniformly.

Firstly, any PC device performs the model training task of the two participants, i.e., it owns the two parts of data mentioned above, trains them locally using the TensorFlow platform separately, obtains the processed parameter files for encryption, and stores the corresponding parts locally or sends them to the other PC device. Next, each of the two PC devices acts as two servers for privacy-preserving parameter processing, and the processing result settings are sent to the PC device performing the local training task, and the above steps are repeated until the end of the iteration rounds.

4.2. Experimental objectives

Since the simulation is intended to compare the feasibility and accuracy of the privacy-preserving scheme in this paper, the optimization of the model training itself has not been adjusted much. Under the same conditions, we mainly compare the relative errors of the performance of the plaintext scheme, the purely added noise scheme and the privacy-preserving scheme in this paper. Therefore, for the

evaluation of this scheme, it is mainly carried out in the following aspects.

1) Verify the performance results and differences between this algorithm and the compared schemes under the setting of ε =9.

2) Verify the performance results and differences between the present algorithm and the comparison scheme under the setting of ε =3.

where the value of ε represents two scales of differential privacy noise, respectively, which are used in this paper to verify the feasibility and accuracy of the scheme under different noise conditions.

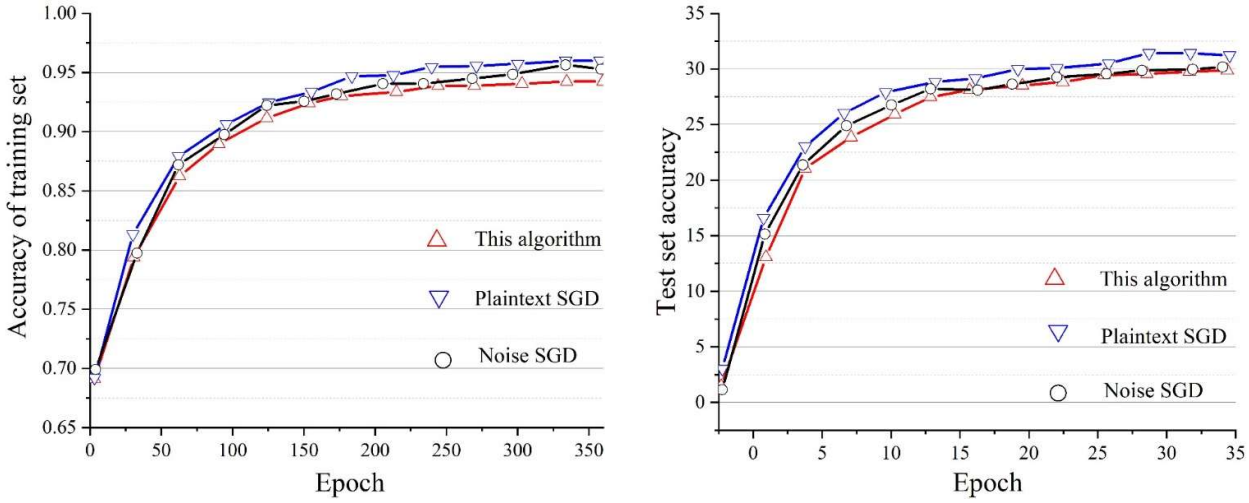
The main purpose of the above two types of experimental objectives is to test whether the present privacy-preserving scheme is feasible and how the algorithm changes in the presence of changes in differential privacy cost, and to analyze the reasons that may affect the results.

4.3. Experimental results and analysis

In this paper, simulation verification is carried out for the above results that need to be evaluated and the results and analysis are as follows:

1) Performance of plaintext SGD algorithm, SGD algorithm after simple noise addition and privacy preserving algorithm of this paper on training set and test set for ε =9.

The performance of the three algorithms on the test set and dataset is shown in Fig. 5. The figure illustrates that both the algorithm with noise added alone and the privacy preserving algorithm of this paper have similar accuracy to the plaintext algorithm at ε =9. That is, the privacy preserving algorithm is closer to the accuracy of the plaintext algorithm when the size of the added noise is small.



(a) Training data set accuracy

(b) Test data set accuracy

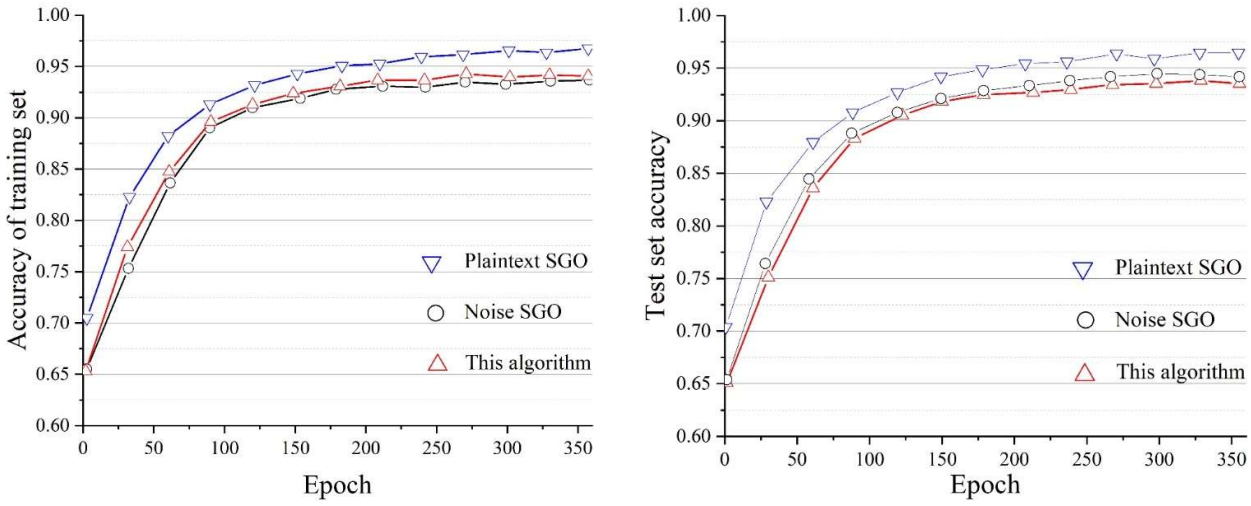
Fig. 5. Three algorithms are shown in the test set and the data set

2) The performance of plaintext SGD algorithm, SGD algorithm after simply adding noise and privacy preserving algorithm of this paper on the training set and test set at ε =3. The test rates of the three schemes are shown in Fig. 6. At ε =3, due to the increase in the size of the added noise, it can be seen that comparing the plaintext SGD algorithm, the SGD algorithm with added noise and the scheme of this paper that adds the same noise have a significant performance degradation. Therefore, for the algorithm of adding differential privacy, both the selection and design of noise are very important, which directly affects the final algorithm effect.

Also comparing the privacy preserving scheme of this paper with the plaintext SGD algorithm, it can be seen that the scheme of this paper is able to provide sufficient accuracy with proper noise selection. Some of the minor losses are also due to the server in the calculation of the average value of the parameters will produce decimal results, resulting in a certain loss of accuracy, as well as the same can

be increased to represent the decimal part of the number of bits to minimize the computational losses.

In summary, the privacy-preserving data cleaning and model training algorithms proposed in this paper have a more satisfactory performance in the actual simulation, proving the feasibility and effectiveness of this paper's scheme.



(a) Training data set accuracy

(b) Test data set accuracy

Fig. 6. The accuracy of the three schemes

5. Conclusion

This paper establishes the automatic detection technology of unknown malicious code for the spatial relationship characteristics of malicious code, which will consider the unique “texture” characteristics of the malicious code as a whole, and carry out the comprehensive multi-feature weighted similarity calculation. Aiming at the service-oriented emergency response data protection technology design needs to comprehensively consider the data recovery point index, data recovery time index and backup data storage overhead, a service-oriented emergency response multi-version data protection model is proposed.

Nine kinds of malicious codes are classically merged with GLCM, LBP and n-gram features to realize the feature fusion of malicious codes. Malicious code detection accuracy calculations are performed for four types of machine learning, KNN, RF, NB and SVM, respectively. The fused features in all four algorithms have higher accuracy than single features. And the fused features are 68.25% higher than LBP features in the plain Bayesian classification algorithm. The automated malicious code detection technology with feature combination designed in this paper in CHAIRS system, the automated response step can complete the processing role in about 8s, which meets the timely demand of case response.

Analyze the relative error of the performance of this paper's privacy-preserving scheme versus the plaintext scheme and the simply-added-noise scheme for different ε values. Combining the accuracy direction of the three schemes in the figure, it can be seen that the scheme in this paper can provide sufficient accuracy with proper noise selection.

References

[1] Wolf, W., White, G. B., Fisch, E. A., Crago, S. P., Pooch, U. W., McMahon, J. O., ... & Lebak, J. M. (2017). Computer system and network security. CRC press.

[2] Fakiha, B. S. (2020). Effectiveness of Security Incident Event Management (SIEM) System for Cyber Security Situation Awareness. Indian Journal of Forensic Medicine & Toxicology, 14(4).

[3] Oriola, O., Adeyemo, A. B., Papadaki, M., & Kotzé, E. (2021). A collaborative approach for national cybersecurity incident management. Information & Computer Security, 29(3), 457-484.

[4] Husák, M., Laštovička, M., & Tovarňák, D. (2021, August). System for continuous collection of contextual information for network security management and incident handling. In Proceedings of the 16th International Conference on Availability, Reliability and Security (pp. 1-8).

[5] Mosenia, A., Sur-Kolay, S., Raghunathan, A., & Jha, N. K. (2017). DISASTER: dedicated intelligent security attacks on sensor-triggered emergency responses. IEEE Transactions on Multi-Scale Computing Systems, 3(4), 255-268.

[6] Mistry, H. K., Mavani, C., Goswami, A., & Patel, R. (2024). A Survey Visualization Systems For Network Security. Educational Administration: Theory and Practice, 30(7), 805-812.

[7] Elahraf, A., Afzal, A., Akhtar, A., Shafiq, B., Vaidya, J., Shamail, S., & Adam, N. R. (2020). A framework for dynamic composition and management of emergency response processes. IEEE transactions on services computing, 15(4), 2018-2031.

[8] Liu, Y., Yuan, Y., Shen, J., & Gao, W. (2021). Emergency response facility location in transportation networks: A literature review. Journal of traffic and transportation engineering (English edition), 8(2), 153-169.

[9] Death, D. (2017). Information security handbook: develop a threat model and incident response strategy to build a strong information security framework. Packt Publishing Ltd.

[10] Azam, M., Khan, M. S. A., Yang, S., Jan, S. U., Senapati, T., Moslem, S., & Mashwani, W. K. (2023). Novel dual partitioned Maclaurin symmetric mean operators for the selection of computer network security system with complex intuitionistic fuzzy setting. IEEE Access, 11, 85050-85066.

[11] Clements, J., Yang, Y., Sharma, A. A., Hu, H., & Lao, Y. (2021, December). Rallying adversarial techniques against deep learning for network security. In 2021 IEEE symposium series on computational intelligence (SSCI) (pp. 01-08). IEEE.

[12] Dolan, E., & Widayanti, R. (2022). Implementation of authentication systems on hotspot network users to improve computer network security. International Journal of Cyber and IT Service Management, 2(1), 88-94.

[13] Ghafir, I., Prenosil, V., Svoboda, J., & Hammoudeh, M. (2016, August). A survey on network security monitoring systems. In 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW) (pp. 77-82). IEEE.

[14] Mughal, A. A. (2022). Well-architected wireless network security. Journal of Humanities and Applied Science Research, 5(1), 32-42.

[15] Neupane, K., Haddad, R., & Chen, L. (2018, April). Next generation firewall for network security: a survey. In SoutheastCon 2018 (pp. 1-6). IEEE.

[16] Dastres, R., & Soori, M. (2021). A review in recent development of network threats and security measures. International Journal of Information Sciences and Computer Engineering.

[17] Sanders, C., & Smith, J. (2013). Applied network security monitoring: collection, detection, and analysis. Elsevier.

[18] Nazir, R., Laghari, A. A., Kumar, K., David, S., & Ali, M. (2021). Survey on wireless network security. Archives of Computational Methods in Engineering, 1-20.

[19] Zhiwei Liu,Xiaoyu Li & Dejun Mu. (2024). Intelligent Analysis and Prediction of Computer Network Security Logs Based on Deep Learning. Electronics(22),4556-4556.

[20] yaping Zhang. (2024). E-commerce Network Security Monitoring Based on HTM Algorithm and Random Forest. Computer Informatization and Mechanical System(6),37-39.

[21] Li Fengjiao & Ren Jianguo. (2024). Suppression of Malicious Code Propagation in Software-Defined Networking. Wireless Personal Communications(1),493-516.