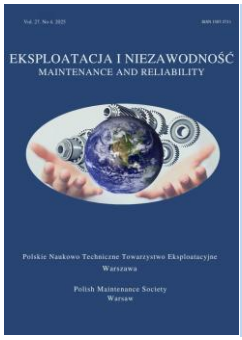

Eksploatacja i Niezawodnosc – Maintenance and Reliability
Volume 27 (2025), Issue 4
journal homepage: http://www.ein.org.pl

Article citation info:
Li B, Zhang J, Network security situation assessment method based on ACDAE-ResBiGRU, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2025: 27(4) http://doi.org/10.17531/ein/205696

Network security situation assessment method based on ACDAE-ResBiGRU

Indexed by:
Web of Science Group

Binyong Li[a,b,c], Jie Zhang[a,*]

[a] School of Cybersecurity (Xin Gu Industrial College), Chengdu University of Information Technology, China
[b] Advanced Cryptography and System Security Key Laboratory of Sichuan Province, China
[c] SUGON Industrial Control and Security Center, China

Highlights

- Uses convolutional denoising autoencoder to enhance spatial representation of attack data.
- Channel attention reduces key feature loss in reconstruction improving assessment accuracy.
- Integrates residual BiGRU to mitigate information loss, boosting classification accuracy.

Abstract

This paper tackles the limitations of traditional network security assessment methods, which suffer from weak feature representation and low classification accuracy. The proposed approach uses a convolutional denoising autoencoder (CDAE) to enhance feature extraction from attack data, with a channel attention mechanism added in the decoder to retain critical spatial information. Additionally, a BiGRU with residual connections is utilized to better extract and preserve contextual information. The network security situation is assessed by calculating a value based on attack severity and impact. Experimental results show that this method significantly outperforms existing models in accuracy, precision, recall, F1-score, and mean square error, proving its effectiveness for large-scale, high-dimensional data. This study is the first to combine CDAE, channel attention, and residual BiGRU, providing new insights into feature extraction and classification for network security. Future work may evaluate its robustness on varied datasets.

Keywords
situational assessment, convolutional denoising autoencoder, attention mechanism, bidirectional gated recurrent unit, residual connections,deep learning

1. Introduction

Network security situation assessment (NSSA) provides innovative perspectives on network behavior by constructing appropriate models through relevant security events, which helps in macroscopic understanding and intent recognition, and provides powerful support for network security decision-making. With the development of artificial intelligence, the application of machine learning and deep learning to network security situation assessment has become a key research direction for experts and scholars.

FAN [1] et al. combined cognitive map with hierarchical analysis based on fuzzy theory to assess and quantify the cybersecurity posture level, and determined the posture status based on the posture value.Doynikova [2] et al. utilized the CVSS scoring system to comprehensively evaluate the indicators such as the network characteristics, attack characteristics, and countermeasures, and analyzed the cybersecurity information status, and the accuracy and efficiency were certain improvement.Lin [3] et al. proposed a cybersecurity posture assessment method based on the combination of Bayesian and attack graphs, and utilized

(*) Corresponding author.
E-mail addresses: B. Li, (ORCID: 0000-0003-3615-1129) lby@cuit.edu,cn, J. Zhang (ORCID: 0009-0008-6077-9382) 1264009657@qq.com,

advanced big data technology to fuse the cybersecurity posture factors and infer the subsequent attack behaviors of the attackers.Li [4] et al. utilized D-S Evidence Theory to fuse the probability values obtained from the data processed by numerous models, so as to analyze the cybersecurity.ALALI [5] et al. applied the idea of fuzzy reasoning in network security posture assessment to assess the security posture of the network from four perspectives: vulnerability, threat, likelihood, and impact level.

However, the network security posture assessment methods based on mathematical statistics and knowledge reasoning rely too much on a priori knowledge or expert experience, with

a large influence of human factors, and although the assessment accuracy is higher, such methods require certain a priori knowledge and there are a large number of mathematical calculations and logical reasoning, which is less efficient and lacks reasonable quantitative standards.

Different from traditional situation assessment methods, deep learning-based models and methods provide new ideas for the research of network security. Lin[6] et al. based on gate recurrent unit (GRU), bi-directional gate recurrent unit (Bi-directional Gate Recurrent Unit), Several neural network models such as BiGRU have been used to detect UNSW-NB15 data set, and the results show that BiGRU has the highest accuracy compared with other models. Deng et al. [7] proposed a feature extraction and fault classification method based on sparse stacked autoencoder network in the environment of industrial complex systems. Compared with other traditional methods, the deep structure of the autoencoder network can learn and fit nonlinear relations in the process well, effectively extract features, and improve classification accuracy. Yang [8] et al. used a parallel feature extraction network composed of encoders for information fusion, then used a bidirectional gated cycle unit for attack category detection, and used the attention mechanism for optimization and improvement, and finally achieved a good network security situation assessment effect. Hu[9] et al. aimed at the problem of increasing training time cost caused by massive network security data. A support vector machine (SVM) network security situation prediction model optimized by MapReduce method is proposed, which effectively improves the prediction efficiency of SVM, but its evaluation index is relatively simple, and it cannot accurately evaluate the global network situation. Li Wangfa [10] et al. studied the feature fusion and recognition method based on multi-core learning, and used the combination of kernel matrix for multi-core learning to effectively improve the recognition accuracy. Chakravarthi et al. [11] proposed an intrusion detection method based on auto-encoder (AE) to extract features, and obtained features with stronger characterization ability. However, gradient disappearance is a problem when using this method to train network models.

In recent years, some researchers have tried to improve deep learning networks with attention mechanisms and residual structures to improve the performance of security detection. The attack detection model designed by Liu et al. [12] combined with the idea of attention mechanism pays more attention to harmful attacks, and the experiment proves that this method is superior to traditional methods. Peng Xingwei et al. [13] integrated the prediction model of ADE and ABiGRU. The model captures the key features of time series data by multi-head attention mechanism, and uses residual structure to reduce the gradient disappearance problem and improve the stability of model training. Qin[14] et al. used the unsupervised learning features of deep CDAE and healthy samples as training sets to solve the problem of insufficient training samples in the initial stage of equipment debugging. The original signal is denoised by the convolution check with good filtering characteristics, which enhances the robustness of the model. Convolutional Denoising Autoencoder algorithm was used for feature extraction and feature dimension reduction. Select the best recruitment and feature criteria. Zhang et al. [15] proposed

a hybrid model that takes advantage of residual network (ResNet) and gated cycle unit (GRU). The results show that the accuracy is 0.81, which is significantly better than traditional machine learning models such as KNN and SVM, and reaches 70% of the human classification benchmark. Chorney et al. [16] proposed a new autoencoder architecture to denoize ECG by utilizing channel and spatial attention and skipping connections, and tested various denoising models. Experiments showed that the convolutional denoising autoencoder with CBAM attention mechanism had the best effect.

Aiming at the shortcomings of the current cyber security posture assessment methods in terms of poor representativeness of the selected features in the dataset, extraction of features,

construction of models, etc., in order to effectively and comprehensively assess the cyber security posture, this paper proposes a cyber security posture assessment method based on the noise-reducing auto-encoder model and bi-directionally gated cyclic unit. The spatial features of different attack types are efficiently and accurately extracted by incorporating the Efficient Channel Attention (ECA) mechanism [17] improved Convolutional Noise Reducing Auto Encoder , and the problem of loss of key features in reconstructing the data is effectively reduced, and then the residual structure is used to improve the BiGRU network , and then use the improved network model for cybersecurity posture element extraction, which effectively alleviates the problem of losing important information of posture data and improves the classification accuracy so as to enhance the precision of posture assessment, and finally calculate the quantized value of cybersecurity posture according to the results of cybersecurity posture assessment.

2. Cybersecurity posture assessment program

In order to solve the problems of poor representativeness of the selected features in the dataset and low accuracy of cyber-attack classification, this paper proposes an evaluation scheme based on an improved convolutional noise reduction autoencoder (ACDAE) with attention mechanism and an improved bidirectional gated recurrent unit (ResBiGRU) utilizing residual structure, i.e., ACDAE-ResBiGRU, which is shown in Fig. 1.

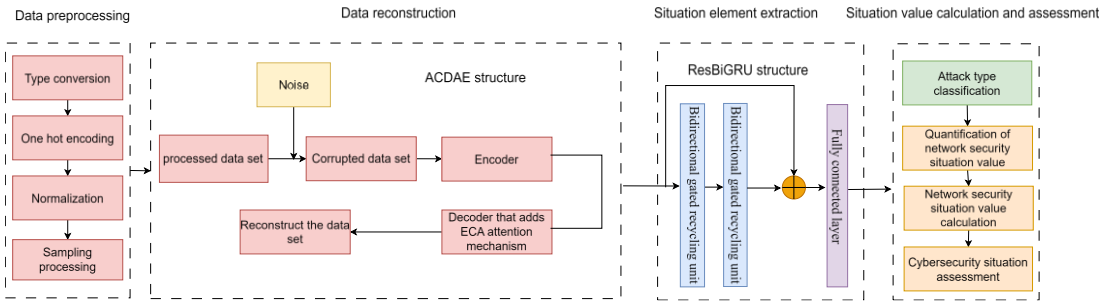

Fig 1. Cybersecurity posture assessment program.

The program consists of the following four steps: data preprocessing, data reconstruction, situational element extraction, and situational value calculation and evaluation.

Step 1 Data preprocessing: non-numeric columns in the data are type converted for solo thermal coding, followed by normalization, which can help with sparse matrices in classification and reduce the inconsistency of the impact of different features on the model, and finally for a small number of samples over-sampling is carried out to balance the distribution of the categories, thus improving the performance of the model.Bert pre-training is not used in this paper,while BERT and similar pre-trained models excel in capturing semantic relationships in textual data, our dataset primarily consists of structured, non-textual features (e.g., network traffic attributes such as IP addresses, ports, and protocol types). These features are more effectively processed using numerical encoding and normalization techniques, as they represent spatial and temporal patterns rather than linguistic semantics. Additionally, BERT is computationally intensive and requires significant resources for fine-tuning, which may not be justified given the nature of our data and the already high performance achieved by our proposed method (as demonstrated by the experimental results). Therefore, we have opted for a more tailored approach that leverages convolutional denoising autoencoders (CDAE) and bidirectional gated recurrent units (BiGRU) to extract spatial and contextual features, which are better suited to the characteristics of network security data.

Step 2 Data reconstruction: the preprocessed dataset is added with noise, encoded by an encoder (mainly composed of convolutional layer, maximum pooling layer, and activation layer), and then decoded by a decoder that incorporates the ECA attention mechanism (mainly composed of convolutional layer, up-sampling, activation layer, and ECA layer), to obtain reconstructed data.

Step 3 Posture element extraction: the reconstructed dataset is input into the posture element extraction module, on the one hand, BiGRU can be utilized to extract temporal features, and since the encoder contains multiple one-dimensional convolutional modules, it can only extract spatial features of the data, so as to realize the extraction of spatial and temporal

dimensions of the features. On the other hand, adding the residual structure can better preserve the posture information and alleviate the problem of posture information loss in the two-layer BiGRU, and finally output the classification results through the fully connected layer.

Step 4 Posture value calculation and assessment: the model is used to identify and classify attacks and quantify attack severity, attack impact valorization, and cybersecurity posture, calculate the cybersecurity posture value, compare it with the real cybersecurity posture value, and finally perform cybersecurity posture assessment.

3. Data reconstruction

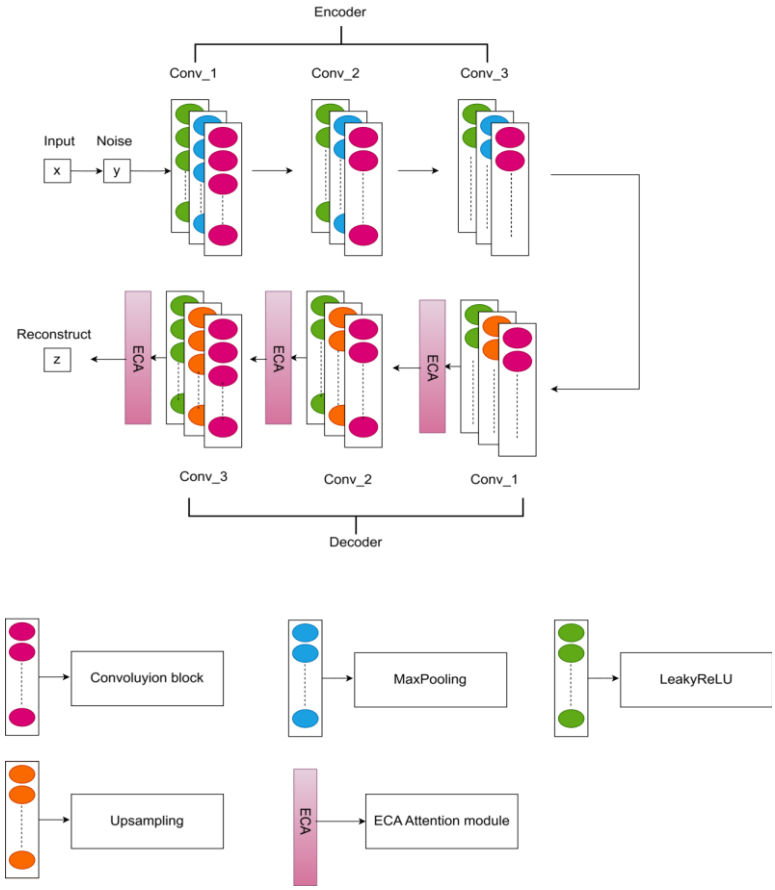

Fig 2. ACDAE module structure.

In order to extract features more efficiently, we propose an improved model based on the traditional noise reduction autoencoder, namely, the convolutional noise reduction autoencoder (ACDAE) that introduces the ECA channel attention mechanism. With this improvement, the model is able to better capture key features, improve the effect of noise reduction and coding, and perform better feature extraction. It consists of an encoder and a decoder:The encoder module and decoder module have three 1-dimensional convolutional layers. The specific module structure is shown in Figure 2

Firstly input the preprocessed dataset after adding noise to generate the post-corruption dataset y for input into the ACDAE model. Where y is generated as shown in equation (1).

$$y = x + \mathcal{N}(0, \sigma^2) \quad (1)$$

where x is the original input data, y is the data after adding Gaussian noise, $\mathcal{N}(0, \sigma^2)$ is Gaussian noise with mean 0 and variance σ^2

Noise processing is then performed to output the denoised traffic attack. Features are extracted using a convolutional layer and downsampled using a maximum pooling layer. In this coding task, noise is suppressed while preserving the underlying structure. The computational formula is shown below:

$$\text{EncoderBlock}(y) = \text{MaxPooling1D}(\text{step})\big(\text{ReLU}(\text{Conv1 D}(y, W_{conv}, b_{conv}))\big) \quad (2)$$

$$F(y) = (\text{EncoderBlock}(y))^3 \quad (3)$$

Where ReLU is the activation function, representing a one-dimensional convolution operation, where y is the input, W is

the convolution weight, and b is the bias.MaxPooling1D(step) is a pooling operation. Step indicates the step length. $(\text{EncoderBlock}(y))^3$ indicates the internal representation of the coding module function, and the corner symbol 3 indicates that three such function operations are performed. $F(y)$ represents the output of the function at the last layer of the coding section.

Next, the decoder module performs convolution and upsampling to decode the compressed traffic attack. As shown in the red box in the figure, the ECA module can effectively update the retrieved features through cross-channel interaction, making the network pay more attention to the relevant information between the feature channels. The output z is then reconstructed by 1D convolution and ECA modules. The calculation formula is as follows:

$$\text{DecoderBlock}(y) = \text{ECA}(\text{Upsample}(\text{size})(\text{ReLU}(\text{Conv1D}\,(F(y), W_{conv}, b_{conv})))) \quad (4)$$

$$G(y) = (\text{DecoderBlock}\,(y))^3 \quad (5)$$

$$z = \sigma(\text{Conv}\ 1D(G(y), W_{conv}, b_{conv})) \quad (6)$$

Upsample(size) indicates the up-sampling operation. size indicates the up-sampling size. ECA(y) indicates Efficient Channel Attention mechanism. $\sigma(x)$ indicates the activation function. $(\text{DecoderBlock}\,(y))^3$ is the internal representation of the decoding module function, and the Angle symbol 3 indicates that three such function operations are performed. $G(y)$ represents the function output of the last layer of the decoded section. z represents the reconstructed output after decoding.

The goal of training is to minimize the mean square error between the output z and the input x. The smaller the value of the loss function $\mathcal{L}(x, z)$, the more likely the output z is to reconstruct the input x, and the better the effect. The calculation formula is as follows:

$$\mathcal{L}(x, z) = \frac{1}{n}\sum_{i=1}^{n}(x_i - z_i)^2 \quad (7)$$

Select the encoder with the smallest reconstruction error as the current attack for feature extraction, and input it into the ResBiGRU module described in Section 4 for situation element extraction.

4. Situation element extraction

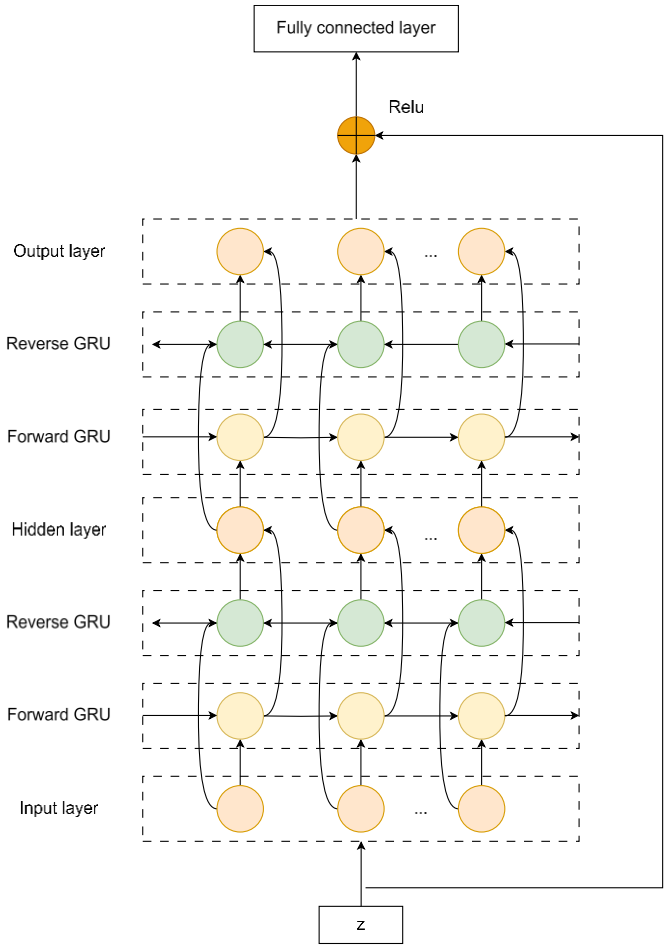

Fig 3. ResBiGRU module structure.

In order to effectively alleviate the problem of important information loss caused by BiGRU in the process of situation information transmission, this paper uses ResNet's residual connection idea for reference, bypasses BiGRU layer and directly transfers part of the input situation data to the output layer of BiGRU layer, and realizes the residual structure. In this way, the constructed ResBiGRU module can learn residual information during information transfer, thereby effectively reducing the loss of critical situation data and improving the accuracy of situation assessment.

Figure 3 shows the structure of the ResBiGRU module, where input data is passed between two-layer BiGRU, which facilitates adequate extraction of time features. The output of the ResBiGRU module is linearly transformed through a fully connected layer. In the BiGRU layer, due to its gating mechanism, the loss of important information becomes more prominent as the number of layers increases. To solve this problem, a residual connection is added between the ResBiGRU module and the fully connected layer in order to preserve and pass the original situation information. Specifically, the model assigns the input data to a variable and, after adding it to the ResBiGRU output, makes a nonlinear transformation using the ReLU activation function. The specific calculation process is as follows:

Input data z_t, first enter the first layer BiGRU, generate the middle hidden state $h_t^{(1)}$, the formula is:

$$h_t^{(1)} = \mathrm{BiGRU}_1(z_t) \tag{8}$$

Then the output $h_t^{(1)}$ of the first layer BiGRU is passed to the second layer BiGRU, and the features of the time series are further extracted. The formula is as follows:

$$h_t^{(2)} = \mathrm{BiGRU}_2\left(h_t^{(1)}\right) \tag{9}$$

Then add the original input data z_t directly to the output $h_t^{(2)}$ of the second layer BiGRU to form a residual connection, the formula is as follows:

$$r_t = z_t + h_t^{(2)} \tag{10}$$

Then, the result of residual connection r_t is nonlinear transformed by ReLU activation function, and the formula is as follows:

$$o_t = \mathrm{ReLU}(r_t) \tag{11}$$

Finally, o_t as the input of the fully connected layer, it is used for the linear transformation of the fully connected layer and the classification result is obtained.

In this method, the original input information is used as the supplement of BiGRU output, and the two are combined to generate the final output of the module, thus effectively alleviating the problem of important information loss in the situation information transmission of BiGRU layer.

5. Calculation and evaluation of network security situation value

In order to fully grasp the network security situation and respond to potential threats in a timely manner, the situation assessment process mainly includes four steps: attack severity quantification, attack impact quantification, network security situation calculation, and network security situation assessment, as shown in Figure 4.

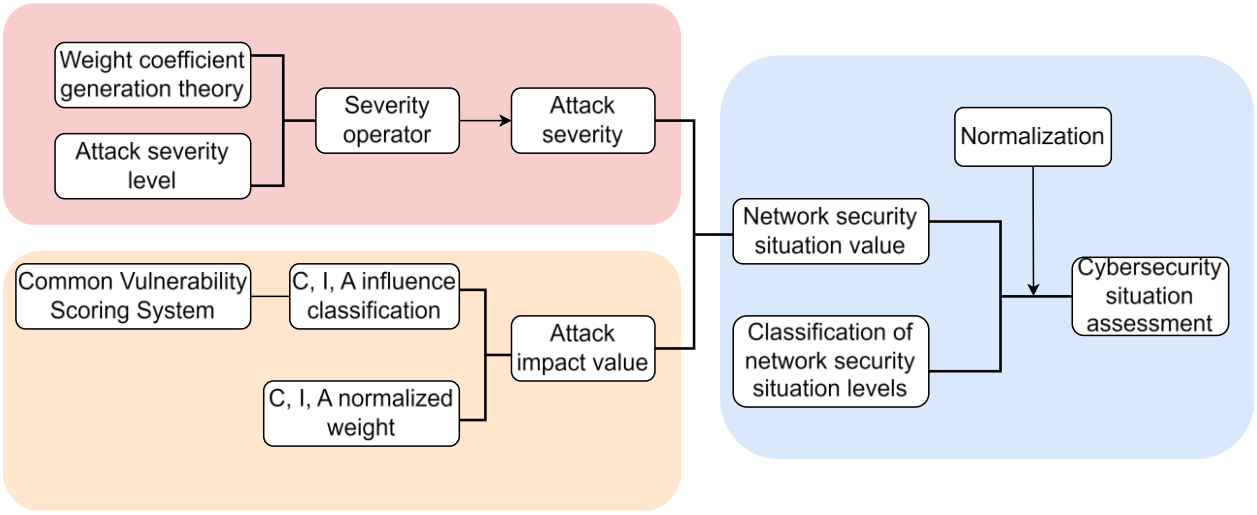

Fig 4. Network security situation value calculation and evaluation diagram.

5.1. Attack Severity Quantification

Combined with the weight coefficient generation theory [18] and the attack severity level, the attack severity operators of various attack types are calculated. The 9 types of attacks are divided into 6 attack severity levels from low to high. The attack severity operator of type i is calculated by equation (12), where k_i represents the attack severity level of type i.

$$T_i = \begin{cases} \frac{1}{2} + \frac{\sqrt{-2\ln\frac{2K_i}{n}}}{6}, & 1 \leqslant K_i < \frac{n}{2}; \\ \frac{1}{2}, & K_i = \frac{n}{2}; \\ \frac{1}{2} - \frac{\sqrt{-2\ln\ (2-\frac{2K_i}{n})}}{6}, & \frac{n}{2} < K_i \leqslant n. \end{cases} \quad (12)$$

The extent and prevalence of each attack type is considered, as well as their impact on system and network security. The nine attack types are divided into six levels, including highest level attack, high level attack, medium level attack, low level attack, low level attack, and lowest level attack.

Highest level attacks: Exploits and Worms have a very high degree of harm, can directly lead to the system being compromised or controlled, and can cause serious damage and data breaches to the entire network.

High level attacks: Backdoors are commonly used to install backdoors into a system that allow an attacker to remotely access a system or application without authorization, making it a high level of harm

Medium-level attacks: Although Dos attacks do not directly cause system intrusion, they can interrupt services and seriously affect services and users. Shellcode attacks generally mean that systems have been compromised or attacked, so they are somewhat harmful.

A Reconnaissance attack is used to obtain information about the target system or network to facilitate subsequent attacks. While they do not directly cause damage on their own, they provide an opportunity for attackers to gain insight into their targets

Low-level attacks: Fuzzers are used to find vulnerabilities that exist in an application or system, but do not attack them directly. Analysis is used to analyze malware samples or attacks to understand the attacker's behavior and purpose. They are less harmful, but can still help strengthen defensive measures.

Lowest level attacks: Generic attacks are general threats or attacks that may involve unspecified attack methods or targets. Generic attacks are general threats to network security. It is the least harmful, but still requires vigilance.

According to the attack severity levels of various attack types on the network and equation (11), the attack severity levels and attack severity operators of 9 attack types are obtained, as shown in Table 1. The higher the attack severity level, the larger the attack severity operator, and the more serious the threat caused by the attack.

Table 1. Attack severity levels and attack severity operators of 9 attack types.

Attack type	Attack severity level	Attack severity counter
Generic	6	0.350
Reconnaissance	4	0.581
Exploits	1	0.789
Fuzzers	5	0.419
Dos	3	0.650
Analysis	5	0.419
Worms	1	0.789
Backdoors	2	0.712
Shellcode	3	0.650

According to the severity operator, the attack severity AL_i can be obtained. The calculation method is as follows

$$\mathrm{AL}_i = M_i \times 10^{T_i} \quad (13)$$

T_i indicates the attack severity operator, and M_i indicates the occurrence times of various attack types.

5.2. Attack Impact Quantification

According to the latest common vulnerability scoring system (CVSS) [19], the impact of these three main aspects on network traffic data is evaluated, including confidentiality, confidentiality, confidentiality, and confidentiality. C), integrity (I), and availability (A) to further classify the attack impact of C, I, and A.

Table 2. Influence values of C, I and A.

index	degree of influence	impact value
Confidentiality	None/Low/High	0/0.2/0.6
integrity	None/Low/High	0/0.2/0.6
Availability	None/Low/High	0/0.2/0.6

Then, combined with Table 2, the logarithmic function quantization method [20] is used to calculate the attack impact values of various attack types, and the calculation formula is as follows.

$$AI_i = \log_2\left(\frac{w_1 2^{C_i} + w_2 2^{I_i} + w_3 2^{A_i}}{3}\right) \quad (14)$$

w_1, w_2, and w_3 are the normalized weights of C, I, and A, respectively. C_i, I_i, and A_i are the values of the C, I, and A impact of each attack type.

5.3. Calculation of Network security situation Value

The network security situation value in this paper is calculated by the following formula considering the severity and impact of attacks on the network:

$$SV = f(AL_i, AI_i) = \sum_{i=1}^{n} AL_i \cdot AI_i \quad (15)$$

AL_i indicates the attack severity, and AI_i indicates the attack impact.

The calculated network security situation value is mapped to the range of 0 to 1 to facilitate the classification of network security situation levels:

$$SV = y_{min} + \frac{y_{max} - y_{min}}{x_{max} - x_{min}}(x_i - x_{min}) \quad (16)$$

SV represents the normalized network security situation value, y_{max} and y_{min} represent the maximum and minimum values of the mapping interval, and x_{max} and x_{min} represent the maximum and minimum values of the situation value, respectively, and represent the current situation value.

5.4. Classification of network security situation

The selection of situation index is very important for the classification of situation level. Only scientific and reasonable selection can reflect the real situation of network. With reference to Chen Lisha et al. [21], four first-level indicators of disaster tolerance, vulnerability, stability and threat were constructed, and each first-level indicator was further refined into four second-level indicators, so as to extract elements of network security situation, as shown in Figure 5

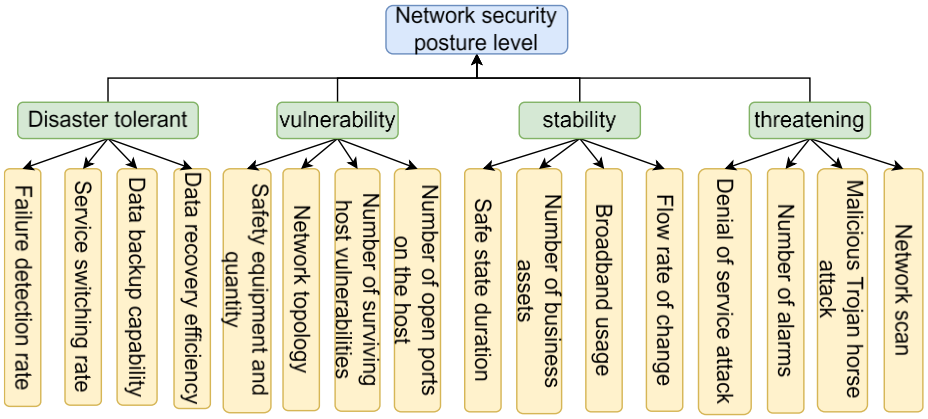

Fig 5. Classification of network security situation.

With reference to the National General Emergency Response Plan for Public Emergencies [22], according to the latest requirements issued by it, and combined with the characteristics of various network threats and attacks, this paper maps the security situation assessment level to the range 0 to 1, and divides it into five situation level ranges: ultra-high risk, high risk, medium risk, low risk and security, as shown in Table 3.

Table 3. Network security situation assessment level.

Situation level	Situation value range
Safety	[0.00，0.20]
low risk	[0.21，0.40]
medium risk	[0.41，0.70]
high risk	[0.71，0.85]
Very high risk	[0.86，1.00]

6. Experiment and analysis

6.1. Experimental Environment

Table 4. Experimental environment.

parameter	Model/version
operating system	Ubuntu 22.04
Memory	16 GB
processor	Intel Core i7-10700
graphics card	NVIDIA RTX 3090 24 GB

6.2. Introduction to Data Sets

UNSW-NB15 dataset: This dataset consists of degrees obtained from the University of New South Wales in 2015. Since its inception, the UNSWNB15 dataset has included a broader

family of attacks, the number of features extracted, and the number of different IP addresses used to simulate and collect data. The dataset is a mixture of real modern normal network traffic and a composite attack activity of contemporary network traffic. Table 5 shows the corresponding attack categories and numbers.

Table 5. UNSW-NB15 data set category and number.

category	Quantity/item
Analysis	2677
Backdoor	2329
DoS	16353
Exploit	44524
Fuzzers	24246
Generic	58871
Normal	93000
Reconaissance	13987
Shellcode	1511
Worms	174

6.3. Data Preprocessing

Unique thermal coding: The UNSW-NB15 dataset has classification features, so it is necessary to convert the classification features into numerical values to give good prediction results. Therefore, in the preprocessing part, the pandas library get-dummies function in python converts these non-numeric columns to numeric values. Since label encoders produce multiple numbers in the same column, choose a single thermal code over a label encoder.

Normalization: Normalization transforms all features into the range [0,1], ensuring that large-scale features do not dominate the model training, thereby improving model stability and convergence speed.In this process, in order to eliminate the impact on the model caused by the excessive range difference between the maximum and minimum values of certain features, numerical normalization is adopted to make the features fall into the same interval, which can be expressed as:

$$x_{i,j}^{*} = \frac{x_{i,j} - \min(x_{i,j})}{\max(x_{i,j}) - \min(x_{i,j})} \tag{17}$$

Where $x_{i,j}$ represents the original value of the feature, $\min(x_{i,j})$ represents the minimum value of the feature, and $\max(x_{i,j})$ represents the maximum value of the feature.

Oversampling: Random oversampling alleviates the problem of data imbalance and improves the prediction accuracy of a few classes by replicating a few class data points. In the UNSW-NB15 dataset, there were few records of Worms, Fuzzers and other categories. In particular, there were only 173 samples of Worms, which was obviously unbalanced compared with the total number of 257673 samples, which affected the prediction effect of a few categories. Oversampling techniques are employed to balance the dataset, particularly for underrepresented classes like Worms and Shellcode. This prevents the model from being biased towards majority classes, improving overall classification accuracy and recall for minority classes.After using random oversampling technique, the accuracy and detection rate of a few classes have been significantly improved.

6.4. Evaluation Indicators

In order to accurately identify various types of attacks faced by the network and improve the Accuracy of network security situation assessment, this paper adopts accuracy, precision, recall and f-score as evaluation indicators. The specific calculation formula is as follows:

Accuracy refers to the ratio of the number of samples correctly classified by the classifier to the total number of samples. Accuracy can be used as an overall evaluation index, reflecting the classification performance of all categories. Can be expressed as

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{18}$$

Precision refers to the proportion of samples predicted by the classifier as positive examples that are actually positive examples. The accuracy rate can help evaluate false positives in the classifier's prediction results, that is, cases in which the classifier incorrectly predicts a positive example for a sample that is actually negative. Can be expressed as

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{19}$$

Recall refers to the proportion of samples that are truly positive examples that are correctly predicted by the classifier as positive examples. In industrial Internet data sets, the recall rate can help evaluate the classifier's ability to recognize positive examples, that is, whether the classifier can correctly predict all real positive examples. Can be expressed as

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{20}$$

The value of F (f-score) is the harmonic average of the accuracy rate and the recall rate, considering the accuracy rate

and the recall rate of the classifier. F-value can be used as a comprehensive performance evaluation index, balancing the accuracy and comprehensiveness of the classifier, and can be expressed as

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{21}$$

TP (True Positive) indicates the number of normal samples that are correctly identified, TN (True Negative) indicates the number of abnormal samples that are correctly identified, and FP (False Positive) indicates the number of normal samples that are incorrectly identified. FN (False Negative) indicates the number of abnormal samples that are incorrectly identified.

6.5. Experimental Analysis

6.5.1. Model training analysis

In order to effectively classify network traffic data attacks, this paper uses a noise reduction autoencoder with improved attention mechanism and a bidirectional gated loop unit model with improved residual network structure for training. The model training loss function is shown in Figure 6.

As shown in Figure 6, the training loss steadily decreases with the increase in training iterations, indicating a gradual reduction in the model's prediction error and an improvement in its data-fitting ability. Notably, around the 40th iteration, the loss drops to a low of approximately 0.1. More importantly, the loss remains stable without significant fluctuations in subsequent training, maintaining this low level. These trends confirm the model's stability and robustness, demonstrating that it has achieved a high degree of fitting optimization after multiple training cycles.

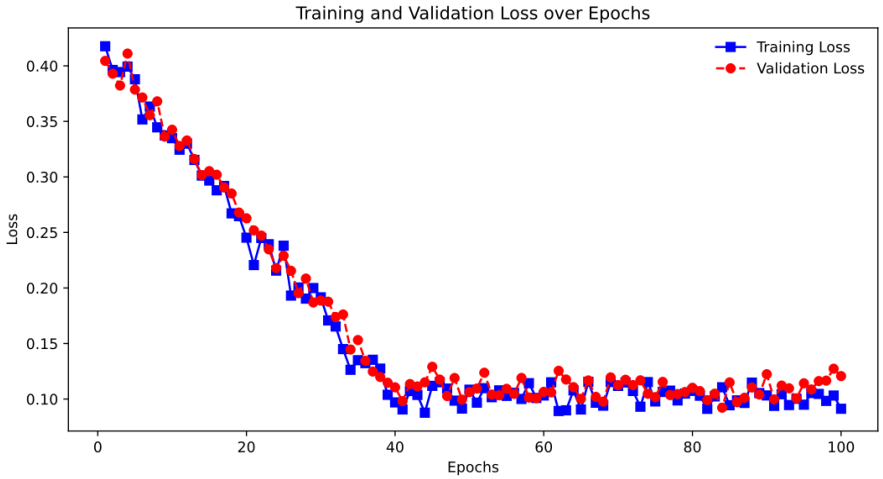

Fig 6. Training loss function diagram.

6.5.2. ROC-AUC curve analysis

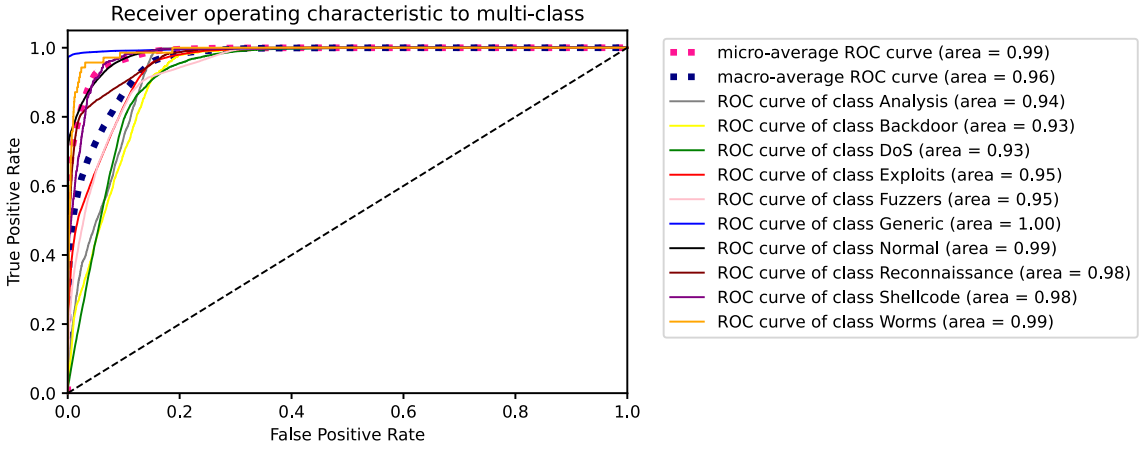

Fig 7. ROC-AUC curve.

Figure 7 of ROC-AUC on UNSW-NB15 shows that the Area Under Curve (AUC) for all categories ranges from 0.93 to 1.0, and the average AUC is 0.969. The model performs very well on multi-class classification tasks, and in particular on some classes (such as Generic, Normal, Shellcode, and Reconnaissance), the model achieves nearly perfect classification performance. However, for some categories (such as Analysis, Backdoor, and DoS), although performance is still good, there is still some room for improvement. Overall, the model is very efficient and accurate in distinguishing different categories of data sets.

6.5.3. Comparative experiment

Furthermore, the SVM model, the RandomForest model, the CNN model, the BiGRU model and the proposed model were compared with the model in this paper. The default network parameter settings of the five models are the same, and all of them use the training and test sets provided by the UNSW-NB15 dataset for training and testing. Figure 8 shows the accuracy, precision, recall, and f-score results of the five models.

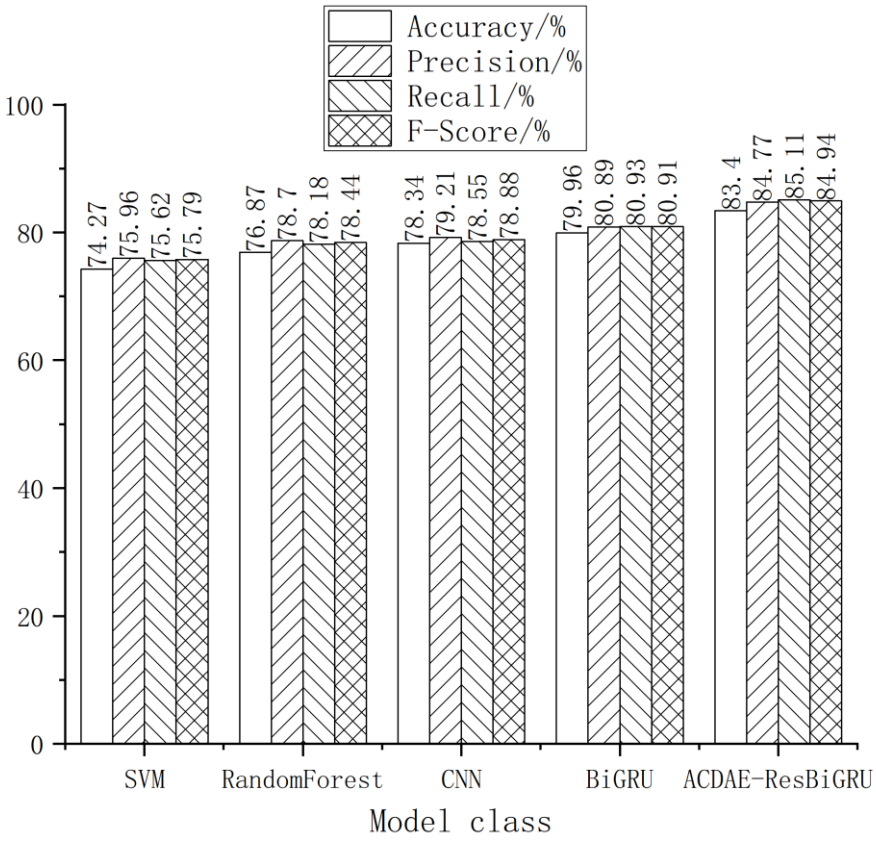

Fig 8. Comparison of detection results of the five models.

As shown in Figure 8, the proposed method demonstrates superior performance in all four evaluation criteria—accuracy, precision, recall, and F1 score—which are crucial for assessing network security situation values. The accuracy of the proposed method is 83.40%, which is 4.56% higher than BiGRU (79.96%), indicating that it is more effective at correctly distinguishing between threats and safe activities. SVM, with an accuracy of 74.27%, has the lowest accuracy, struggling to reliably classify network security situations. Random Forest improves on SVM with an accuracy of 76.87%, but still lags behind the proposed method by over 6.5%. CNN achieves an accuracy of 78.34%, performing better than SVM and Random Forest, yet still falls short compared to BiGRU and the proposed method. In terms of precision, the proposed method achieves 84.77%, 3.88% higher than BiGRU (80.89%), indicating that it is more accurate in identifying true threats while minimizing false positives. SVM (75.96%) and Random Forest (78.70%) perform relatively well but still misclassify a larger number of legitimate activities as threats compared to the proposed method. CNN (79.21%) improves on these models, but still lags behind BiGRU and the proposed method. Regarding recall, which reflects the model's ability to detect true threats, the proposed

method achieves 85.11%, surpassing BiGRU by 4.18%. This shows that the proposed method is particularly strong in detecting actual threats, while SVM (75.62%) and Random Forest (78.18%) miss a larger proportion of true threats. CNN (78.55%) performs better than SVM and Random Forest, but still doesn't match the proposed method's performance. For the F1 score, the proposed method achieves 84.94%, which is 4.03% higher than BiGRU (80.91%), demonstrating a better balance between precision and recall. SVM (75.79%) has the lowest F1 score, indicating an imbalanced performance with both lower precision and recall. Random Forest (78.44%) and CNN (78.88%) show improved F1 scores over SVM but still fall short of the proposed method. Overall, the proposed method outperforms all four models in all evaluation criteria, making it the most effective approach for network security situation assessment, offering improvements in accurately classifying threats, minimizing false positives, detecting actual threats, and maintaining a balanced precision-recall trade-off.

In order to further verify the effectiveness of the method presented in this paper, the confusion matrix of multi-classification detection is first shown, and it can be seen that the overall accuracy is relatively high, as shown in Figure 9.

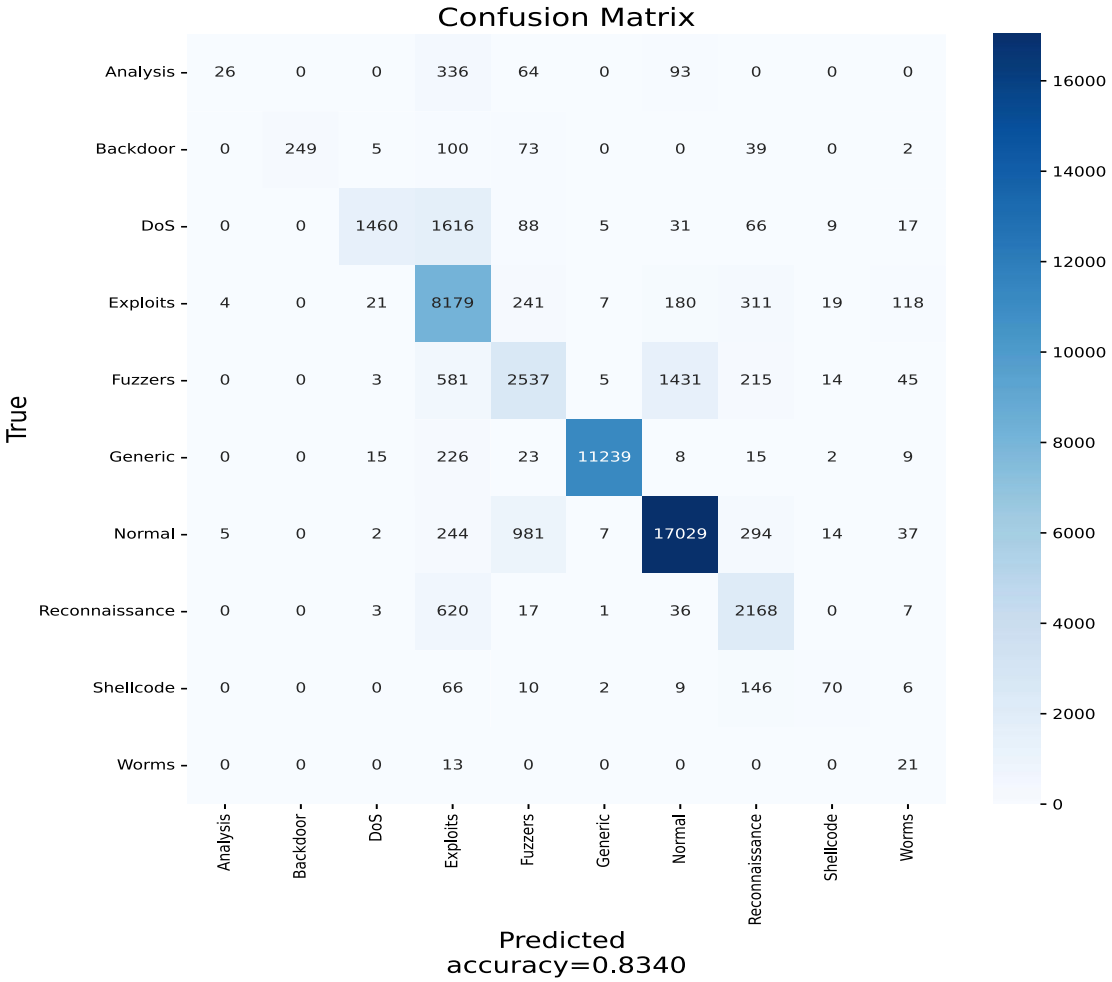

Fig 9. Confusion matrix of the model in this paper.

Then, a group of network traffic attack data is randomly selected, as shown in Table 6, showing seven attack categories and corresponding quantities. In addition, five groups of confusion matrices are used in this paper to visually display the predicted and actual values, as shown in Figure 10. It can be seen that the prediction of various attack types in this paper is basically consistent.

Table 6. Network traffic attack data.

Serial number	Category	Quantity/item
1	Backdoor	1
2	Exploit	6
3	Fuzzers	5
4	Generic	7
5	Normal	17
6	Reconaissance	3
7	Shellcode	1

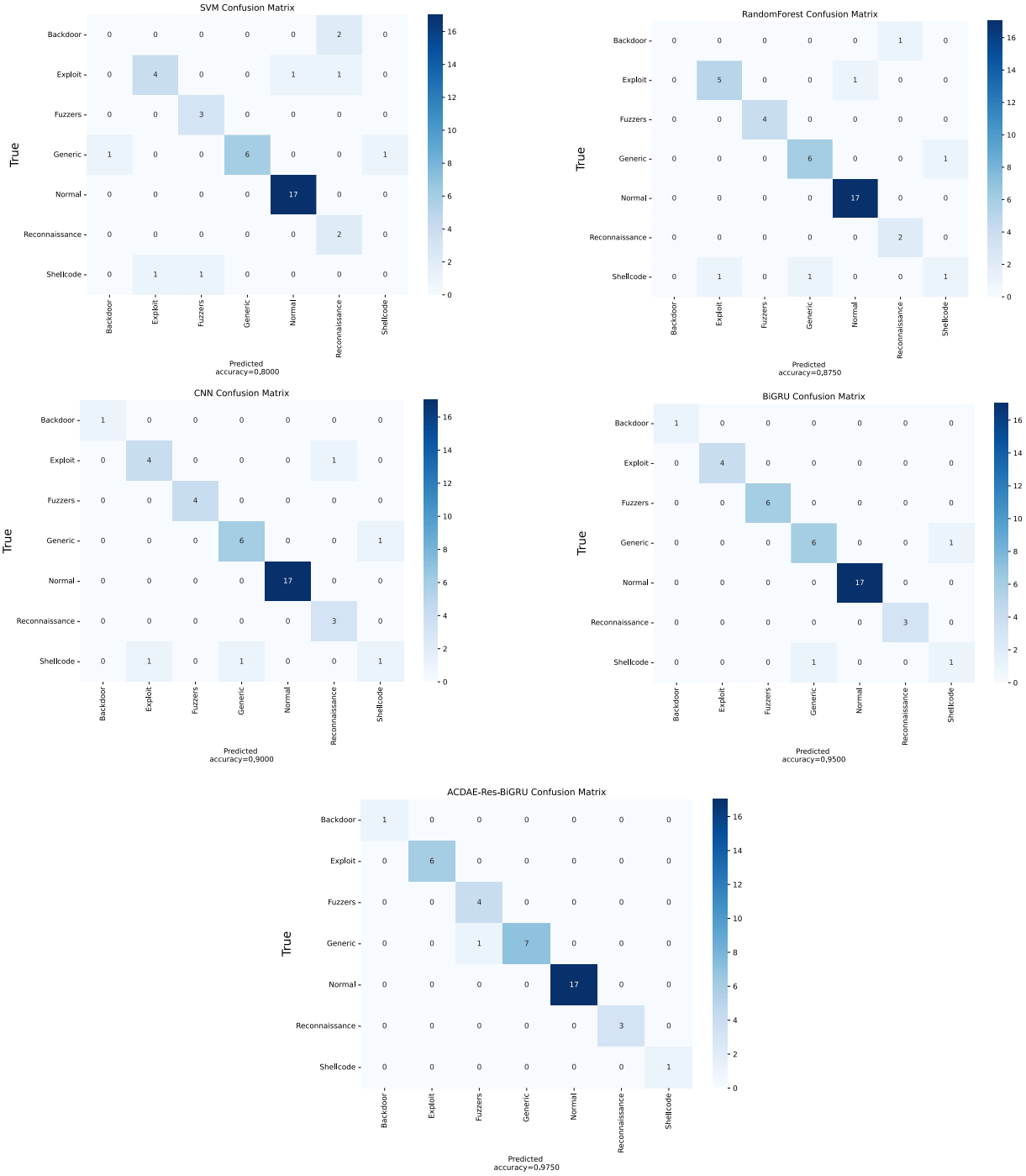

Fig 10. Multi-classification confusion matrix comparison.

From the comparison of the five groups of confusion matrices, it can be seen that both SVM and RandomForest predict Normal as Exploit. Normal is a normal type, and Exploit is the highest level attack type. However, there is a large gap between the two attack severity levels, which will lead to a large deviation in the situation value. Affect the effect of situation assessment. Although CNN and BiGRU avoid the above problems, the accuracy rate of Exploit is not high, and there are two or three types of attack prediction errors, which will also affect the effect of situation assessment to some extent. The method in this paper has only one prediction error, which is to predict Fuzzers as Generic. Since both attack severity levels are relatively low, the difference in attack impact factors is not large, so the final situation value will not be too big, and the network security situation can be reasonably assessed.

Next, 200 groups of the same number of test data were randomly selected from the test set, and SVM, RandomForest, CNN, BiGRU and the model in this paper were respectively

used for threat detection. The network security situation value was calculated and compared by using the attack severity operator and attack impact value. It can be found that the situation value calculated by the model in this paper is closest to the real situation value. Figure 11 shows the comparison of network situation values of 15 groups of experiments.

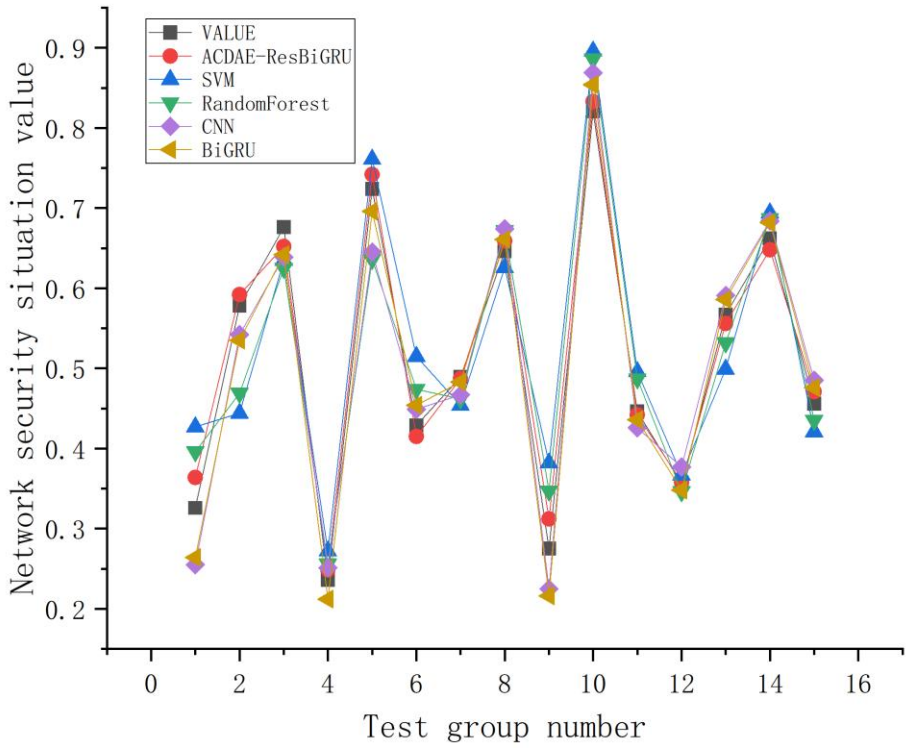

Fig 11. Comparison of security situation values of 15 experimental groups.

As can be seen from Figure 11, although the network security situation value calculated by each model is close to the real value, the situation evaluated by the method in this paper is more consistent with the real situation. A detailed analysis of Table 3 and Figure 11 shows that:

1) The network security situation value obtained by the model in this paper and the real situation value are always in the same situation assessment interval, and the error is minimal. In most groups, the situation assessment results of the five models are the same as the real situation assessment results, but the situation values of the model in this paper are closer to the real situation values.

2) The situation assessment results of four models based on SVM, RandomForest, CNN and BiGRU are different from the real situation results. For example, in group 1, the SVM model was assessed as moderate risk, while the present model and the real situation were both low-risk. In group 5, the assessment results of CNN and BiGRU models were medium risk, while both the model in this paper and the real situation were high risk. In group 10, SVM, RandomForest, CNN and BiGRU models all have high risk, while both the present model and the real situation are high risk.

In addition, this paper analyzes and compares the root-mean-square error calculation results of SVM, RandomForest, CNN and BiGRU models on the test set, and lists the MSE index of each model in detail, as shown in Table 7. Among them, the MSE of the proposed model is the lowest, reaching 0.111, which is significantly better than other models, especially compared with the traditional SVM and RandomForest models, the performance is significantly improved. There are also obvious advantages over CNN and BiGRU. This result shows the powerful capability of the proposed method in dealing with network security tasks, and shows its superior performance in the evaluation of network security situation value.

Table 7. Root mean square error calculation results.

Model	mean square error
SVM	0.963
RandomForest	0.742
CNN	0.565
BiGRU	0.384
ACDAE-ResBiGRU	0.111

According to the above analysis, it can be seen that the situation value obtained by the method in this paper can basically reflect the real situation value result, and it is more accurate and reliable than other methods, and has the lowest mean square error, which fully indicates that the model in this paper can effectively evaluate the network security situation.

6.5.4. Ablation experiment

In order to further evaluate the influence of Attention and ResNet components on the performance of the model, three groups of component ablation comparison experiments were used to verify the superiority and rationality of the model

For the Attention component, CDAE, CDAE-BiGRU and CDAE-ResBiGRU models were used respectively, and the effectiveness was verified by adding and removing Attention components in each model, as shown in Table 8.

Table 8. Influence of the Attention component on the model effect.

Model	Accuracy/%	Precision/%	Recall/%	F-Score/%
CDAE	79.65	79.28	79.58	79.43
ACDAE	80.32	80.74	80.38	80.56
CDAE-BiGRU	81.46	81.37	81.13	81.25
ACDAE-BiGRU	81.83	81.64	81.50	81.57
CDAE-ResBiGRU	82.12	82.56	82.08	82.32
ACDAE-ResBiGRU	83.40	84.77	85.11	84.94

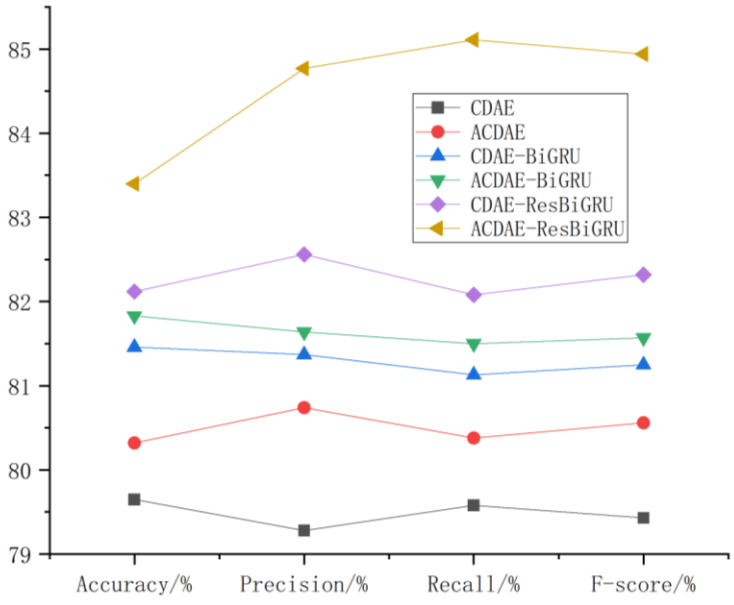

Fig 12. Influence of the Attention component on the three models.

Visualize the results of Table 8 as shown in Figure 12. We draw a curve with the four evaluation indicators and their values as horizontal and vertical coordinates. Through the analysis of the chart, it can be concluded that after adding the Attention module to the basic model, the evaluation performance has been significantly improved, and the increase of the four evaluation indicators is between 0.37% and 3.03%. The reasons for this are summarized as follows: The channel attention mechanism adaptively focuses on the most important features in the input data by assigning different weights to each channel. This weight allocation can not only help the model effectively select useful features in the process of noise reduction, ignore noise and

redundant information, so as to improve the accuracy of feature extraction, but also enhance the representation ability of features according to task requirements. Compared to traditional methods such as AE, DAE, and CDAE, these methods often assign the same importance to all input features, which may lead to a focus on irrelevant or noisy features, resulting in information loss. In addition, when they deal with high-dimensional data, they often lose key information and ignore the relationships between features. The channel attention mechanism can more accurately control the contribution degree of each feature channel, avoid the excessive attention of the model to irrelevant information, and thus improve the model evaluation performance.

For ResNet components, BiGRU, CDAE-BiGRU and ACDAE-BiGRU models were adopted respectively, and Attention components were added and removed in each model to verify the effectiveness, as shown in Table 9.

Table 9. Effects of ResNet components on model effects.

Model	Accuracy/%	Precision/%	Recall/%	F-Score/%
BiGRU	79.96	79.89	79.93	79.91
ResBiGRU	80.91	80.97	80.95	80.96
CDAE-BiGRU	81.46	81.37	81.13	81.25
CDAE-ResBiGRU	82.12	82.56	82.08	82.32
ACDAE-BiGRU	81.83	81.64	81.50	81.57
ACDAE-ResBiGRU	83.40	84.77	85.11	84.94

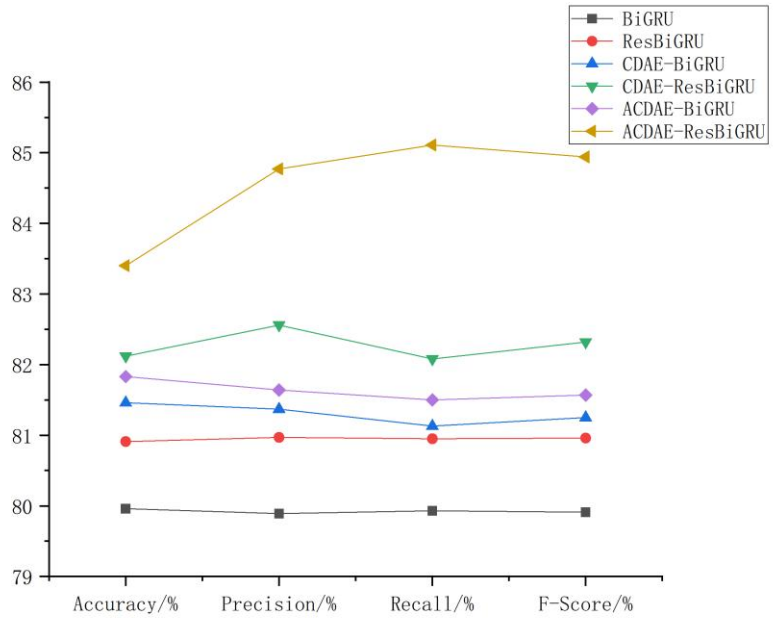

Fig 13. Influence of the ResNet component on the three models.

Table 9 results are shown in Figure 13. Four evaluation indicators show that the model performance is significantly improved after the addition of ResNet module: the accuracy rate increases by about 0.66%-1.57%, the precision rate increases by about 1.08%-3.13%, the recall rate increases by about 0.95%-3.61%, and the F-value increases by about 1.05%-3.47%. The reasons are summarized as follows: First, ResNet effectively solves the problem of gradient disappearance and explosion of deep networks through residual connections. The residual connection allows information to bypass the middle layer and pass directly to the deep layer, improving the

efficiency of feature transfer and gradient backpass,

reducing model degradation, and making it easier to train and capture complex features. Secondly, UNSW-NB15 data set is time series data. Residual connection enhances the ability of model to capture long-term dependence in recursive network, improves the limitation of traditional RNN, LSTM and GRU in gradient problem, and makes training more stable. Finally, the residual connection preserves the original information, avoids the error accumulation caused by information loss or ambiguity, improves the hierarchy of feature learning, and reduces the overfitting of the model to noise or redundant information.

In summary, ResNet's residual linkage mechanism can significantly improve model efficiency and stability when dealing with complex time series data, especially when dealing with long time dependencies and deep structures. Compared with the traditional AE, DAE and CDAE models, the addition of channel attention mechanism can bring significant improvements in feature selection, expression ability, noise reduction effect and robustness, especially when processing complex time series data. The combination of residual connection and channel attention mechanism can not only effectively alleviate the loss of important information, but also further improve the overall performance of the model.

To better highlight the performance differences between ACDAE, ResBiGRU, and ACDAE-ResBiGRU, the key evaluation metrics from the ablation study are summarized in the table 10:

Table 10. Significant improvements of ACDAE-ResBiGRU.

Model	Accuracy/%	Precision/%	Recall/%	F-Score/%
ACDAE	80.32	80.74	80.38	80.56
ResBiGRU	80.91	80.97	80.95	80.96
ACDAE-ResBiGRU	83.40	84.77	85.11	84.94

As shown in the table, the ACDAE-ResBiGRU model outperforms both ACDAE and ResBiGRU across all evaluation metrics, demonstrating significant improvements:

Compared to ACDAE: The accuracy improved by 3.08%, precision by 4.03%, recall by 4.73%, and F1-Score by 4.38%.These improvements can be attributed to the introduction of the residual connection, which enhances temporal feature extraction and preserves critical information, while the ACDAE module effectively captures spatial features.Compared to ResBiGRU: The accuracy improved by 2.49%, precision by 3.80%, recall by 4.16%, and F1-Score by 3.98%.This indicates the significant role of the ACDAE module in noise reduction and spatial feature enhancement, complementing ResBiGRU's temporal feature extraction capabilities.

Overall, the ACDAE-ResBiGRU model effectively combines the strengths of spatial and temporal feature extraction, bolstered by the attention mechanism and residual connections. This results in significantly improved accuracy and robustness for network security situation assessment.

7. Summarize

In this paper, an improved network security situation assessment method is proposed, combining a convolutional denoising autoencoder with an enhanced attention mechanism and a residual bidirectional gated recurrent unit (BiGRU) model to address the limitations of traditional approaches in feature representation and classification accuracy. The proposed method enhances feature extraction through a multi-layer convolutional layer in the denoising autoencoder, improving the accuracy of data reconstruction. To further optimize the process, a channel attention mechanism is incorporated into the decoder, ensuring that key features are preserved during reconstruction. Additionally, a residual connection is introduced to the traditional BiGRU, forming the residual BiGRU module, which effectively reduces the loss of critical situation information during transmission, improving classification accuracy and overall situation assessment performance.In order to fully grasp the network security situation and respond to potential threats in a timely manner, the situation assessment process consists of four essential steps: attack severity quantification, attack impact quantification, network security situation calculation, and network security situation assessment. Each of these steps plays a critical role in ensuring the accuracy and reliability of the overall assessment. Attack severity quantification involves determining the seriousness of an attack based on predefined criteria, while attack impact quantification evaluates the extent of the attack’s consequences on the network. Network security situation calculation then combines the results from these two steps to derive an overall situation value that reflects the current security state. Finally, network security situation assessment

applies this calculated value to assess the network's security posture, enabling timely decision-making and response.

The proposed method is particularly suitable for scenarios involving large-scale network traffic analysis, such as detecting distributed denial-of-service (DDoS) attacks . While the proposed method performs well on the UNSW-NB15 dataset, its adaptability to other datasets with different traffic patterns and attack types remains to be verified. Future work could explore domain adaptation techniques to improve generalizability. Additionally, the scalability and applicability of this method in diverse network environments and under different attack scenarios could be further investigated, potentially expanding its capability to address more complex and evolving network security threats.The proposed method not only provides an effective solution for network security situation assessment but also offers valuable insights for future research in this area.In the follow-up research, on the one hand, it is necessary to refine the quantitative evaluation indicators of network security situation and consider more factors affecting network security situation. On the other hand, it is necessary to improve the model and apply it to more kinds of network security detection tasks.

Funding

This research was funded by the Sichuan Science and Technology Program, Grant No.2024NSFSC0515 and No.2024ZHCG0182.

Reference

1. Fan Z, Tan C, Li X. A hierarchical method for assessing cyber security situation based on ontology and fuzzy cognitive maps[J]. International Journal of Information and Computer Security, 2021, 14(3-4): 242-262. https://doi.org/10.1504/IJICS.2021.114704
2. Doynikova E, Kotenko I. CVSS-based probabilistic risk assessment for cyber situational awareness and countermeasure selection[C]//2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP). IEEE, 2017: 346-353. https://doi.org/10.1109/PDP.2017.44
3. Lin P, Chen Y. Dynamic network security situation prediction based on bayesian attack graph and big data[C]//2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2018: 992-998. https://doi.org/10.1109/ITOEC.2018.8740765
4. Li Y, Yao S, Zhang R, et al. Analyzing host security using D-S evidence theory and multisource information fusion[J]. International Journal of Intelligent Systems, 2021, 36(2): 1053-1068. https://doi.org/10.1002/int.22330
5. Alali M, Almogren A, Hassan M M, et al. Improving risk assessment model of cyber security using fuzzy logic inference system[J]. Computers & Security, 2018, 74: 323-339. https://doi.org/10.1016/j.cose.2017.09.011
6. Lin Y, Wang J, Tu Y, et al. Time-related network intrusion detection model: a deep learning method[C]//2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019: 1-6. https://doi.org/10.1109/GLOBECOM38437.2019.9013302
7. Deng Z, Li Y, Zhu H, et al. Sparse stacked autoencoder network for complex system monitoring with industrial applications[J]. Chaos, Solitons & Fractals, 2020, 137: https://doi.org/10.1016/j.chaos.2020.109838
8. Yang H Y， Zhang Z X， Zhang L. Network security situation assessments with parallel feature extraction and an improved BiGRU[J]. Journal of Tsinghua University(Science and Technology)， 2022， 62(5): 842-848. (in Chi-nese)
9. Hujj，Mady，Liuc，etal. Network security situation prediction based on MR-SVM[J]. IEEE Access，2019，7：130937-130945. https://doi.org/10.1109/ACCESS.2019.2939490
10. Li Wangfa. Coal-rock image feature fusion and recognition method based on multi-core learning [J]. Electronic Technology and Software Engineering, 2019(07): 63-65. (in Chinese)
11. Chakravarthi S S, Kannan R J. Non-linear dimensionality reduction-based intrusion detection using deep autoencoder[J]. International Journal of Advanced Computer Science and Applications, 2019, 10(8). https://doi.org/10.14569/IJACSA.2019.0100822
12. Liu T, Qi Y, Shi L, et al. Locate-Then-Detect: Real-time Web Attack Detection via Attention-based Deep Neural Networks[C]//IJCAI. 2019: 4725-4731. https://doi.org/10.24963/ijcai.2019/656
13. Peng Xingwei, Yuan Lingyun. Internet of Things security situation prediction based on ADE-ABiGRU [J]. Network Security and Data Governance, 2023,42 (12): 48-53.
14. Qin Z, Chang Q, Li Q, et al. Research on Abnormal Feature Extraction and Early Fault Alarm Method of Rolling Bearings Based on CDAE and KLD[J]. International Journal of Acoustics & Vibration, 2023, 28(2). https://doi.org/10.20855/ijav.2023.28.21929

15. Zhang J. Music genre classification with ResNet and BiGRU using visual spectrograms[J]. arXiv preprint arXiv:2307.10773, 2023.
16. Chorney W, Wang H, He L, et al. Convolutional block attention autoencoder for denoising electrocardiograms[J]. Biomedical Signal Processing and Control, 2023, 86: https://doi.org/10.1016/j.bspc.2023.105242.
17. Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11534-11542. https://doi.org/10.1109/CVPR42600.2020.01155
18. Xiaowu Liu, Huiqiang Wang, Hong'wu Lv,et al.. Fusion-based cognitive awareness-control model for network security situation[J]. Journal of Software, 2016, 27(8): 2099-2114.
19. Deb R, Roy S. Common vulnerability scoring system for SDN environment[J]. Int. J. Eng. Adv. Technol., 2019, 8(6): 4022-4029. https://doi.org/10.35940/ijeat.F9302.088619
20. Xi Rongrong, Yun Xiaochun, ZHANG Yongzheng. Quantitative threat situational assessment based on contextual information[J]. Journal of Software, 2015, 26(7): 1638-1649.
21. Chen Lisha, ZHANG Fengli, WANG Juan. Construction of network security situation evaluation index system [J]. Journal of Chongqing University of Science and Technology: Natural Science Edition, 2008, 10(3): 135-137.
22. The State Department. National General Emergency Plan for Public Emergencies [M]. Beijing: China Legal Publishing House, 2006.