Deep Learning-Based Multidimensional Assessment for Network Security Situations

Yixian Liu, Zhiwei Yao*

Abstract—**Aiming at the problem of insufficient adaptability of traditional network security situation assessment methods in dynamic and complex environments, to improve the detection accuracy of complex attacks and the adaptability to complex environments, this paper proposes a multi-dimensional network security situation assessment method based on deep learning. Specifically, to improve the accuracy of attack detection, an intrusion detection model based on time convolution network (TCN), time pyramid attention mechanism (TPA) and gating mechanism is designed; to enhance the adaptability to the complex network environment, this paper combines the intrusion detection results with the network traffic analysis and generates a network security posture values. Experimental validation on the CICIDS2017 dataset shows that the model performs well in the assessment indicators such as accuracy and recall. At the same time, the method demonstrates strong adaptability and scalability and can effectively support situational awareness tasks in different network environments, providing a new solution for security protection in complex network environments.**

Index Terms—**Deep Learning, Network Security Situation Assessment, Temporal Convolutional Networks, Time Pyramid Attention, Gating Mechanism**

I. Introduction

The the field of modern cyber security, with the rapid development of information technology, the Internet has become a core component of global economic, social, and political activities. However, the wide application of Internet technology has also brought unprecedented network security challenges. As the means and strategies of cyber attacks become more and more complex, traditional security protection methods are no longer able to effectively cope with new threats [1]. Especially in the face of large-scale data traffic and complex network environments, how to effectively manage and deal with these security risks has become one of the core issues to be solved in the field of information security [2].

In response to these challenges, Network Security Situation Awareness (NSSA), as a key technical means, has received extensive attention in recent years [3]. Network security situation awareness includes situation identification, situation assessment, and situation prediction. These links can not only monitor and predict potential threats in the network environment in real-time but also provide information support for decision-makers to help them take effective protective measures in time, thus enhancing the accuracy and real-time performance of network protection [4].

In NSSA, network security situation assessment is its core component, which can comprehensively analyze various security factors in the network and provide comprehensive information about the current network security state. By identifying and responding to threats in a timely manner, situational assessments can guide defense decisions and ensure that cybersecurity is effectively protected. Therefore, improving the accuracy and efficiency of network security situation assessment is crucial to improving the capability of the entire network protection system.

Although the existing network security situation assessment methods have made some progress, due to the increasingly complex network environment and the continuous evolution of attack means, the traditional methods based on mathematical logic and knowledge reasoning often have great limitations when dealing with threats in large-scale and dynamic environments. These methods are usually unable to deal with the demands of massive network traffic and attacks. Therefore, it is difficult to evaluate the situation based on the real-time state of the network [5]. Although the current deep learning method can improve the accuracy, from the perspective of application, it cannot achieve targeted evaluation according to the actual situation.

To fill this research gap, this paper proposes a network security situation assessment method based on deep learning and multi-dimensional information. The method combines attack detection, traffic analysis, and environmental factors, and proposes a new intrusion detection model to monitor network traffic in real-time by introducing a sequential Convolutional network (TCN), Time Pyramid Attention (TPA), and Gating Mechanism. Different types of attacks are scored according to the environment requirements and attack characteristics. In addition, the method can further introduce more factors, such as user behavior analysis, according to actual needs to assess the network security posture more comprehensively. Finally, through the fusion of these multidimensional data, a comprehensive network security situation value is calculated to help decision-makers achieve more accurate protection strategies and improve network security protection capabilities.

The main contributions of this paper are as follows:

(1)An intrusion detection model is proposed: combining TCN, TPA, and gating mechanism, the timing relationship between data is fully extracted, and the attack accuracy is improved.

(2)Based on the intrusion detection model, a

Manuscript received March 4, 2025; revised August 5, 2025.

Yixian Liu is a senior engineer at the School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi 710000, China (e-mail: liu-yi-xian@xupt.edu.cn).

Zhiwei Yao is a master's student at the School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi 710000, China (corresponding author, e-mail: 2811978452@qq.com).

multidimensional situation assessment method is proposed: combining traffic distribution, attack impact, and environmental requirements, the assessment can be customized according to the environment.

(3)Comprehensive evaluation: CICIDS2017 was used to evaluate multiple indicators of the intrusion detection model, and the network security situation was visually evaluated in smart homes, smart cities, and smart agriculture.

II. Related Work

Network security situation assessment is an important means to ensure network security, which can provide security personnel with quantitative analysis and decision support for the global state of the network.

In the traditional method, some studies have adopted different techniques to conduct network security situation assessment. Wang et al. [6] proposed a network security situation assessment method based on an analytic hierarchy process (AHP), which can reflect the overall security situation of the network and provide support for high-level decision-making. Alali et al. [7] generated risk assessment results based on vulnerability, threat, possibility, and impact and improved the assessment model based on fuzzy logic reasoning, emphasizing the importance of the attack itself in the assessment. Li et al. [8] Based on the risk assessment method of the mrmrIg feature selection model and attack graph model, by combining the hidden Markov chain model, the real-time performance and accuracy of attack prediction are improved. Chen et al. [9] improved the capability of network security situation assessment through the data fusion model based on the RBF neural network and proposed a new idea of multi-dimensional data fusion in situation assessment. However, these methods face the problem of poor adaptability, especially when dealing with real dynamic network environments.

As a new method, deep learning is also widely used in situation assessment methods. Zhang et al. [10] built the decision tree (DT) and long short-term memory (LSTM) network for network security situation awareness, which regarded attacks as possibilities and described the network situation by combining the occurrence probability and impact of attacks. Yang et al. [11] proposed an evaluation method based on adversarial learning and used an adversarial training model to evaluate security by taking attack impact in the Common Vulnerability Scoring System (CVSS) as an evaluation index.

From the situation assessment based on deep learning, it can be seen that the premise of a excellent situation assessment is that the performance of the intrusion detection model is excellent enough. Fortunately, the field of intrusion detection has developed very well in recent years[12]. Li et al. [13] proposed an intrusion detection model based on a deep learning framework of multi-layer Extreme Learning Machine (ELM). Lopes et al. [14] proposed multiple intrusion detection models based on temporal convolutional networks (TCN), which significantly improved detection performance. Wu et al. [15] proposed an efficient intrusion detection method by combining location embedding technology and a Transformer model to extract features from high-dimensional data.

In summary, network security situation assessment can be divided into two parts: intrusion detection and determination of attack indicators. Intrusion detection is ultimately feature extraction. Although many models have been proposed in previous studies, there are still some deficiencies in feature extraction. Similarly, many studies have been conducted on determining attack indicators, but they are not applicable to specific complex environments. Therefore, this study proposes a multidimensional situation assessment method based on deep learning, which provides new ideas for the specific application methods of situation assessment.

III. Intrusion Detection Model

In intrusion detection, this study proposes an intrusion detection model that combines TCN, TPA, and a gating mechanism to improve the performance of network intrusion detection. The model architecture is shown in Fig. 1.

A. Tcn

TCN extracts temporal features in data through

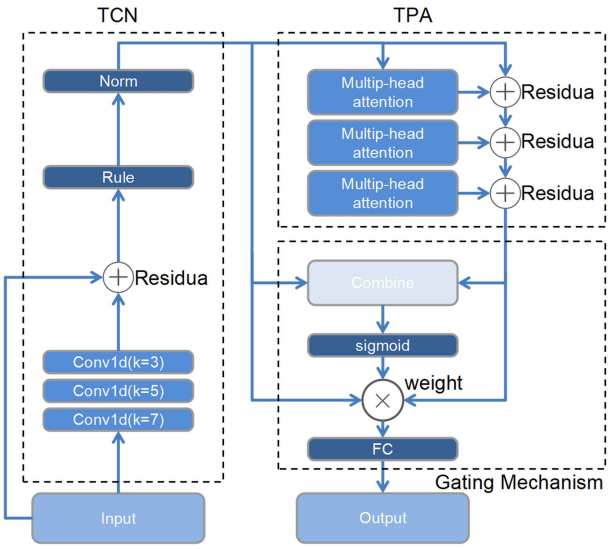


Fig. 1. Intrusion Detection Model architecture

convolution operations, solves the gradient vanishing and explosion problems in traditional Recurrent Neural Network (RNN) and LSTM, uses causal convolution to ensure that each output depends only on the current and previous inputs, and uses extended convolution to expand the receptive field to capture long-term dependencies. The specific design is as follows:

(1) Multi-scale convolution: Pass through multiple convolution layers; each convolution layer has a different kernel size to capture features of different time scales.

(2) Residual connection: Combine the output with the initial input through residual connection to alleviate the gradient vanishing and increase the stability of training.

(3) Nonlinear activation: The ReLU activation function introduces nonlinearity to enhance the expressiveness of the model.

(4) Layer normalization: Perform layer normalization on the output to make the model more stable.

In actual environments, attacks are accompanied by abnormal time series fluctuations in traffic. The TCN module can capture the characteristics of these anomalies at multiple scales. For example, a slow and long-term PostScan attack will appear to be more continuous, and TCN can capture the characteristics of this type of traffic.

B. TPA

The temporal pyramid attention combines the hierarchical attention mechanism of [16] and the self-attention mechanism of [17]. Through multi-head attention, the model can dynamically adjust the focus on different time periods and extract the characteristics of key time intervals. The specific design is as follows:

(1) Pyramid attention layer: three multi-head attention layers are used to build a pyramid shape to calculate the attention weights at different time scales.

(2) Residual connection: Similar to TCN, the output of each layer is added back to its input through residual connection to ensure that the focus is dynamically adjusted without losing the original information.

In actual environments, TCN captures global abnormal time series fluctuations, and TPA captures local abnormal time series in a short period of time. For example, in a DDoS attack, the traffic will increase sharply in a short period of time. TPA can flexibly capture the characteristics of this abnormal traffic.

C. Gating Mechanism

The adaptive gating mechanism is inspired by the gating mechanism proposed in [18] and aims to dynamically fuse the outputs of TCN and TPA. By adaptively adjusting the weights of feature fusion, the importance of each feature is optimized according to the input features, thereby improving the classification performance of the model. The specific design is as follows:

(1) Combine: Linearly concatenate the outputs of TCN and TPA along the feature dimension, and then use a fully connected layer to map them to a single dimension.

(2) Sigmoid activation: The sigmoid function is used to normalize the gates to between 0 and 1, and the weights of each of the two modules are determined.

(3) Weighted output: The outputs of TCN and TPA are weighted and summed according to the weights to obtain the final fused feature representation.

(4) FC: The feature representation is converted into a predicted value of the category using a fully connected layer.

In the above example of DDoS and PortScan, the gating mechanism will gradually learn the different weights of the two modules, thereby forming an adaptive gating.

IV. Network Security Situation Assessment Method

The situation assessment, as shown in Fig. 2, includes four parts: data preprocessing, intrusion detection, attack scoring, and network security situation assessment.

A. Data Preprocessing

This study uses the CICIDS2017 dataset [19], which was jointly developed by the Canadian Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC) in 2017. The dataset was collected in 5 days and contains normal traffic and 14 types of attacks. Normal traffic accounts for 80.30%, and the rare attack Heartbleed accounts for only 0.00039%. It simulates the real-world traffic situation and is one of the most commonly used datasets for intrusion detection.

The dataset contains some abnormal data. The cleaning steps are:

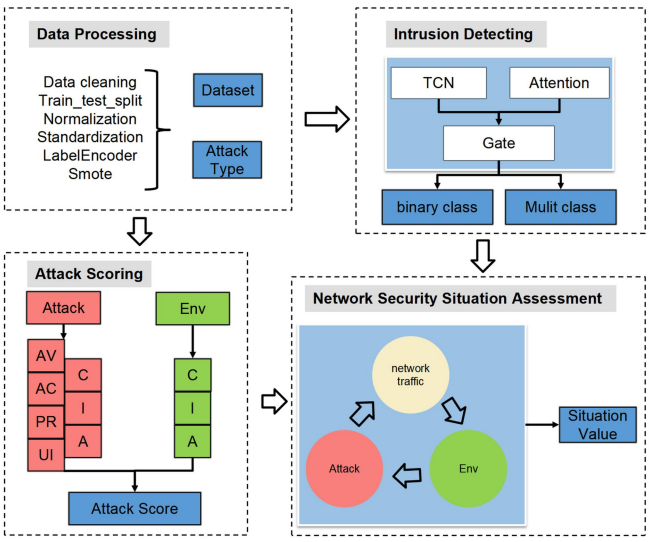


Fig. 2. Network Security Situation Assessment method structure

(1)Delete the meaningless feature column "Timestamp."

(2)Delete the duplicate feature column "Fwd Header Length.1."

(3)Replace missing values with 0.

(4)Replace Infinity with the maximum value of the current column plus 1.

To avoid data leakage, after processing abnormal data, the dataset is first divided into the training set, validation set, and test set, with a ratio of 7:1:2. The training set is used for model training, the validation set is used for hyperparameter tuning and model selection, and the test set is used to evaluate the final performance of the model. The division of the dataset is shown in Table I.

TABLE I
THE DATASET SPLIT

Class	Train set	Validation set	Test set
BENIGN	1591167	227310	454620
DoS Hulk	161751	23107	46215
PortScan	111251	15893	31786
DDoS	89618	12803	25606
DoS GoldenEye	7205	1029	2059
FTP-Patator	5556	794	1588
SSH-Patator	4128	590	1179
DoS slowloris	4057	580	1159
DoS Slowhttptest	3849	550	1100
Bot	1376	197	393
Web Attack-Brute Force	1055	151	301
Web Attack-XSS	457	65	130
Infiltration	26	3	7
Web Attack-Sql Injection	15	2	4
Heartbleed	8	1	2

Normalize all numerical features to the range of [0, 1] to improve the training efficiency of the model. Normalization not only speeds up the convergence of training but also improves the accuracy of the model.

Standardize the features to have zero mean and unit variance. Standardization helps to eliminate the scale differences between features, making the data more suitable for the optimization algorithm, thereby improving the performance of the model.

Label encode the categorical features and convert them into numerical values. In multi-classification, the category labels are encoded from 0 to 14; in binary classification, the labels are encoded as 0 and 1, where 0 represents normal traffic and other values represent attacks.

SMOTE [20] is used to oversample the minority classes in the training set. In this study, the categories with sample sizes lower than the average of all categories are increased to the average. This helps to improve the generalization ability of the model. By synthesizing minority class samples, the model's bias towards the majority class is effectively reduced, thereby improving its performance in real-world scenarios.

B. Intrusion Detecting

During the training phase, the preprocessed training data is first fed into the model. The TCN module captures multi-scale temporal features within the data, while the TPA module further extracts key features from different time periods. Finally, a gating mechanism adaptively fuses the outputs of TCN and TPA, optimizing feature weighting and producing the final classification result. After training, the model generates a set of optimized parameters that effectively capture the features and patterns in the data.

Subsequently, the trained model is applied to the test set to evaluate its classification performance on unseen data. During the testing phase, the model performs inference based on the parameters learned during training, producing both binary and multi-class classification results.

C. Attack Scoring

In this study, key metrics from CVSS4.0 are selected to score the attacks, including Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), User Interaction (UI), Confidentiality (C), Integrity (I), and Availability (A). Table II. presents the impact scores for these metrics. These indicators provide a comprehensive assessment of the potential impact and severity of different types of attacks. By combining these scores, the proposed method enables a more effective evaluation of the risks associated with each attack, thereby establishing a more informed and precise situational assessment method.

TABLE II
IMPACT SCORES OF INDICATORS

Indicator	Impact	Score
AV	Network(N)/Adjacent(A)/Local (L) /Physical (P)	0.85/0.62/0.55/0.2
AC	Low (L) / High (H)	0.77/0.44
PR	None (N)/Low(L)/High(H)	0.85/0.62/0.55/0.27
UI	None(N)/Passive(P)/Active (A)	0.85/0.56/0.2
C	None (N) /Low (L) /High (H)	0/0.22/0.56
I	None (N) /Low (L) /High (H)	0/0.22/0.56
A	None (N) /Low (L) /High (H)	0/0.22/0.56

In practical applications, evaluating attack impact solely using CVSS4.0 is insufficient. A customized assessment of security posture is required for different network environments. This study improves the evaluation method by incorporating the CIA metrics, which have the greatest impact on attacks. Based on discussions in the literature [21, 22, 23] about the security requirements for smart homes, smart cities, and smart agriculture, Table III. presents the CIA requirements for these three types of networks. In real-world scenarios, these metrics can be customized for specific applications to meet the particular security needs of each environment.

(1)Smart Home: Confidentiality is moderately important with a score of 0.6, reflecting the necessity of protecting personal data. Integrity has a score of 0.4, indicating it is less critical, as minor data inaccuracies may be tolerable. In this environment, Availability is the highest priority with a score of 0.8, as devices need to operate reliably and continuously.

(2)Smart City: Both Confidentiality and Integrity are highly important, each with a score of 0.8, reflecting the need to protect sensitive data and ensure the accurate functioning of urban infrastructure. Availability is the most

crucial factor, with a score of 1.0, due to the necessity for continuous service in critical urban systems.

(3)Smart Agriculture: Due to the relatively low sensitivity of agricultural data, Confidentiality is less important, with a score of 0.4. Integrity, at 0.6, is moderately important to ensure the accuracy of operational data. Availability is rated 0.5, reflecting a balance between the need for reliable service and the tolerance for occasional disruptions in agricultural operations.

TABLE III
CIA ENVIRONMENTAL REQUIREMENTS SCORE

Environment	Indicator	Score
Smart_Home	$C_{env}/I_{env}/A_{env}$	0.6/0.4/0.8
Smart_City	$C_{env}/I_{env}/A_{env}$	0.8/0.8/1.0
Smart_Agriculture	$C_{env}/I_{env}/A_{env}$	0.4/0.6/0.5

By adjusting these weights according to the specific environment, flexible adaptability is provided, ensuring that the situational awareness system can be fine-tuned to accommodate different security priorities across various domains.

The impact score for each attack is calculated using the following formula:

$$\text{Impact} = 1-(1-C\times \text{C}_{\text{env}})\times(1-I\times \text{I}_{\text{env}})\times(1-A\times \text{A}_{\text{env}}) \quad (1)$$

$$\text{Exploitability} = 8.22\times \text{AV}\times \text{AC}\times \text{PR}\times \text{UI} \quad (2)$$

$$\text{Score}_{\text{i}} = \min(\text{Impact}+\text{Exploitability},10) \quad (3)$$

where C, I, and A are classified as Impact indicators, C_{env}, I_{env}, and A_{env} represent the environmental requirements for these aspects, and AV, AC, PR, and UI fall under Exploitability indicators. The parameter 8.22 is a fixed value, and it is stipulated that the maximum score for each attack type is capped at 10.

D. Situational Assessment

The situational assessment in this study is based on the network security situational assessment method proposed in [11]. In their calculation, there is a small error in the classification of situational assessment levels: in their calculation, the maximum values of CIA are all 0.56, assuming only one flow, representing a single attack, and the CIA values for this attack are all 0.56. Using the formula proposed in their paper, the calculated situational value becomes 1.56, which exceeds 1. This study corrects this error by modifying the maximum value in high-risk scenarios. Table IV. shows the distribution of network security status levels.

TABLE IV
NETWORK SECURITY SITUATION LEVEL

V	NSSL
0.00~0.20	Safety
0.21~0.40	Low risk
0.41~0.60	Medium risk
0.61~0.80	High risk
0.81~10.00	Super risk

The situational value is calculated using the following formula:

$$\text{V} = \frac{\text{p}\times\sum_{i=1}^{n} S_i \times t_i}{N - t_{nml}} \quad (4)$$

where p represents the probability of binary classification, nnn denotes the attack type, S_i is the score obtained for each attack, and t_i indicates the frequency of each attack. Since the score for normal traffic is 0, normal traffic should be excluded from the calculations. N refers to the total number of flows, while t_{nml} signifies the count of normal traffic.

V. EXPERIMENTAL RESULTS

A. Evaluation Metrics

In this study, we used the following evaluation indicators to evaluate the performance of the model. Each indicator is calculated using the parameters of the confusion matrix [24], and the parameters are shown in Table V.

TABLE V
BASIC STRUCTURE OF CONFUSION MATRIX

	Predicted Positive	Predicted Negative
Actual Positive	True Positives (TP)	False Negatives (FN)
Actual Negative	False Positives (FP)	True Negatives (TN)

Accuracy: It indicates the proportion of correct predictions among all predictions, which measures the overall classification ability of the model. However, it cannot fully reflect the model performance in the case of imbalanced data. The calculation formula is as follows:

$$\text{Accuracy} = \frac{TP+\text{TN}}{TP+\text{TN}+\text{FP}+\text{FN}} \quad (5)$$

Precision: It indicates the proportion of true positive samples predicted as positive, which measures the classification ability of the model for positive samples. The calculation formula is as follows:

$$\text{Precision} = \frac{TP}{TP+\text{FP}} \quad (6)$$

Recall: It indicates the proportion of actual positive samples that are correctly predicted as positive, which measures the classification ability of the model for positive samples. It is suitable for imbalanced data sets. The calculation formula is as follows:

$$\text{Recall} = \frac{TP}{TP+\text{FN}} \quad (7)$$

F_1Score: It indicates the harmonic mean of recall and precision, which takes these two indicators into consideration in a balanced way and fully reflects the classification performance of the model. The calculation formula is as follows:

$$\text{F}_1 score = 2\times\frac{p\text{recision}\times\text{recall}}{p\text{recision}+\text{recall}} \quad (8)$$

ROC curve (Receiver Operating Characteristic Curve) [25]: It indicates the relationship between true positive rate (TPR) and false positive rate (FPR), which is used to evaluate the performance of the model under different classification thresholds. By calculating the area under the ROC curve (AUC), the overall classification ability of the model can be quantified. The closer the AUC is to 1, the better the performance of the model. The formulas for calculating TPR and FPR are as follows:

$$\text{T}PR = \frac{TP}{TP+\text{FN}} \quad (9)$$

$$FPR = \frac{FP}{FP + FN} \quad (10)$$

B. Hyperparameter Sensitivity Analysis

To verify the robustness and rationality of the model design, we conducted a series of multi-class classification experiments on the CICIDS2017 dataset. In each experiment, only one hyperparameter was varied at a time, while all other hyperparameters were fixed according to the final configuration listed in Table VI.

TABLE VI
MODEL HYPERPARAMETERS

Parameter	Value
Kernel_sizes	[3, 5, 7]
Activation	ReLU
num_heads	4
num_layers	3
loss_function	BCELoss or CrossEntropyLoss
optimizer	Adam
learning_rate	0.001
batch_size	128

As shown in Tables VII-X, various kernel size combinations in the TCN module were evaluated. The results indicate that when the kernel sizes of the three convolutional layers are set to 3, 5, and 7, respectively, the model achieves the best performance, with an Accuracy of 99.77%, a Recall of 99.76%, and an F_1score of 99.78%. Although the accuracy varies only slightly across other combinations, the recall drops significantly, suggesting that this specific configuration better enables the model to capture temporal features across multiple time scales.

TABLE VII
EFFECT OF KERNEL SIZES(%)

Kernel_sizes	[1,2,3]	[3,3,3]	[3,5,7]	[3,3,5]	[3,5,7]
Accurary	99.05	98.95	99.77	99.72	99.68
Recall	97.06	99.50	99.76	98.60	99.58
F_1score	98.10	99.44	99.78	98.66	99.51

TABLE VIII
EFFECT OF ATTENTION HEADS(%)

Attention_heads	2	4	6	8
Accurary	99.50	99.77	99.42	99.43
Recall	99.01	99.76	98.80	99.20
F_1score	99.38	99.78	98.99	99.33

TABLE IX
EFFECT OF NUM_LAYERS(%)

Num_layers	2	3	4	5
Accurary	99.60	99.77	99.65	99.41
Recall	99.55	99.76	98.13	99.45
F_1score	99.50	99.78	98.34	99.23

TABLE X
EFFECT OF LEARNING RATE(%)

Learning_rate	0.0001	0.001	0.01	0.1
Accurary	99.10	99.77	99.55	99.20
Recall	98.95	99.76	99.30	98.80
F_1score	99.02	99.78	99.42	98.99

C. Binary Classification

In binary classification, the intrusion detection model is compared with the classic models of RF [26], DT [27], CNN [28], and LSTM [29]. Fig. 3. shows the ROC curves of the five models in binary classification. Compared with other models, the AUC of our model is closest to 1, which indicates that it has the strongest ability to distinguish

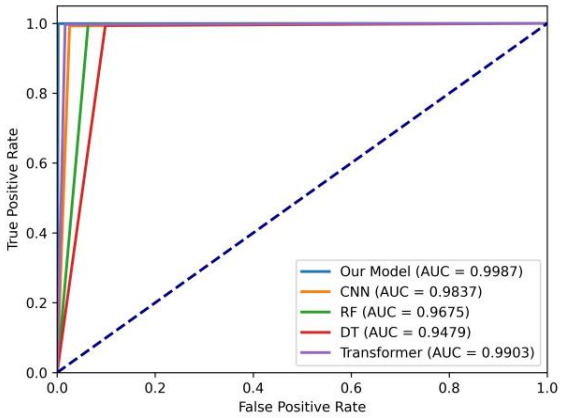


Fig. 3. The ROC curves.

normal traffic from attack traffic.

D. Multi-Class Classification

In multi-classification, the model is also compared with four classic models. Considering that attack samples are usually a minority in datasets and practical applications, resulting in unbalanced data distribution, recall is selected as the key evaluation indicator for each category. Table XI. shows the recall of various attacks, as well as the overall accuracy, recall, precision, and F_1score. The results show that compared with other models, our model has excellent classification capabilities in most categories. However, the recall of Sql Injection and Heartbleed is worse than that of some models. The reason is that the number of these categories is too small. Although oversampling is performed by SMOTE, the features learned by the model are still limited. Overall, our model is better than the classic model in all indicators.

To further verify the effectiveness of the proposed model, we compared it with the advanced models that did similar work. As shown in Table XII, our model has the best performance in all indicators. Compared with the models of [30, 31], our model achieves more efficient time series feature extraction through TCN and avoids the gradient vanishing problem that may exist in LSTM-type methods; compared with the models of [32, 33, 34], our model further extracts time series features through TPA and adaptively fuses the features of the first two modules through the gating mechanism to obtain higher performance.

E. Network Security Situation Assessment

To verify the effectiveness of the network security posture assessment method in multiple environments, we randomly selected 15 traffic groups from the test set, each containing 25 to 50 traffic at random, to simulate network traffic per minute.

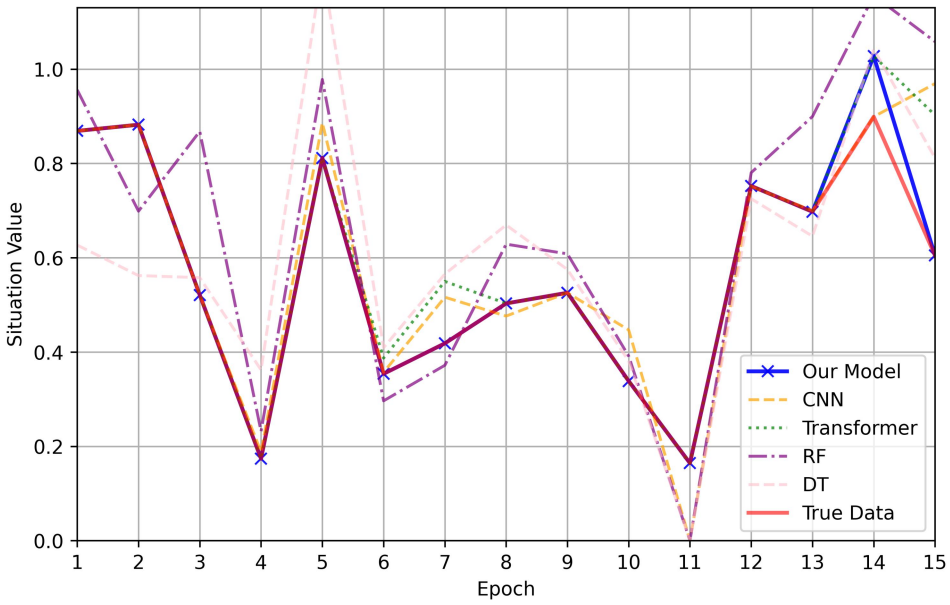
As shown in Fig. 4-6, compared with other models, the pro- posed situation assessment model is closer to the real value most of the time, and most of the situation values are even the same as the real data. In different environments, the change in situa- tion value is subtle, because the CIA is an important indicator in the environment; however, these

TABLE XI
RECALL RATES FOR EACH CLASS IN EACH MODEL AND OVERALL ACCURACY, RECALL, PRECISION, AND F_1SCORE

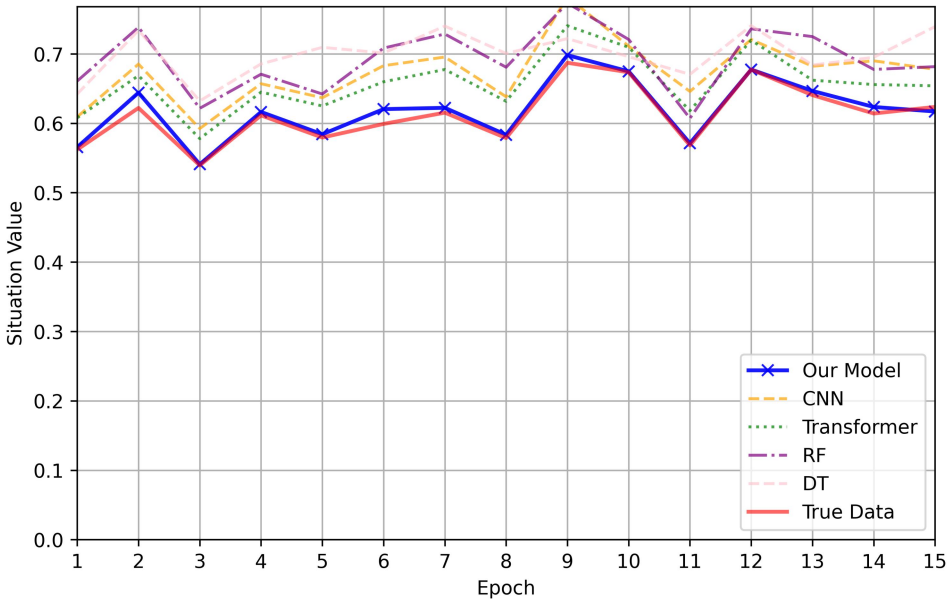
Model	DT	RF	CNN	Transform	Our Model
BENIGN	97.85	96.75	95.78	98.93	99.77
DoS Hulk	59.73	88.29	97.88	97.59	99.90
PortScan	99.10	99.89	97.57	99.92	100.00
DDoS	67.74	98.08	97.89	98.18	99.96
DoS GoldenEye	24.87	96.75	99.27	99.22	99.71
FTP-Patator	99.81	99.81	99.18	99.87	99.69
SSH-Patator	98.90	50.64	81.34	51.57	99.81
DoS slowloris	46.68	97.67	98.27	98.62	99.74
DoS Slowhttptest	69.27	97.45	99.09	99.09	99.09
Bot	2.29	97.20	97.71	94.15	98.98
Web Attack-Brute Force	0.00	33.55	11.63	33.55	40.86
Web Attack-XSS	90.77	79.23	95.38	80.77	86.15
Infiltration	71.43	71.43	71.43	71.43	71.43
Web Attack-Sql Injection	0.00	75.00	75.00	25.00	50.00
Heartbleed	50.00	100.00	100.00	100.00	50.00

TABLE XII
RECALL RATES FOR EACH CLASS IN EACH MODEL AND OVERALL ACCURACY, RECALL, PRECISION, AND F_1SCORE

Model	Accuracy	Recall	Precision	F_1score
[30](CNN+LSTM)	98.67	-	-	93.32
[31](CNN+BiLSTM)	97.70	97.80	97.70	97.70
[32](CNN+LSTM+Attention)	99.71	96.78	97.14	96.61
[33](CNN+BiLSTM+Attention)	95.67	95.90	95.82	95.86
[34](CNN+GRU+Attention)	99.65	99.63	99.65	99.64
Our Model	99.77	99.83	99.76	99.78

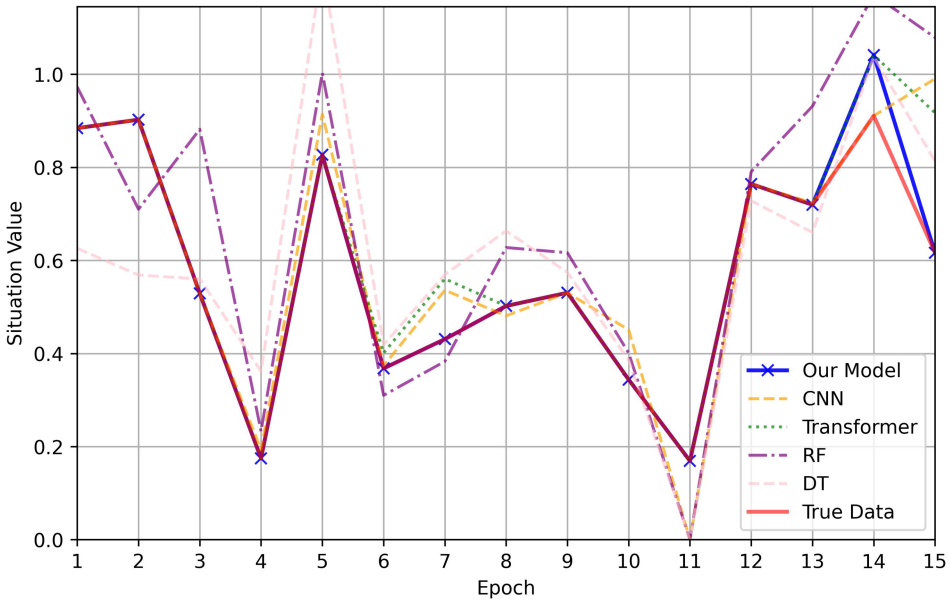


(a) Small-scale network

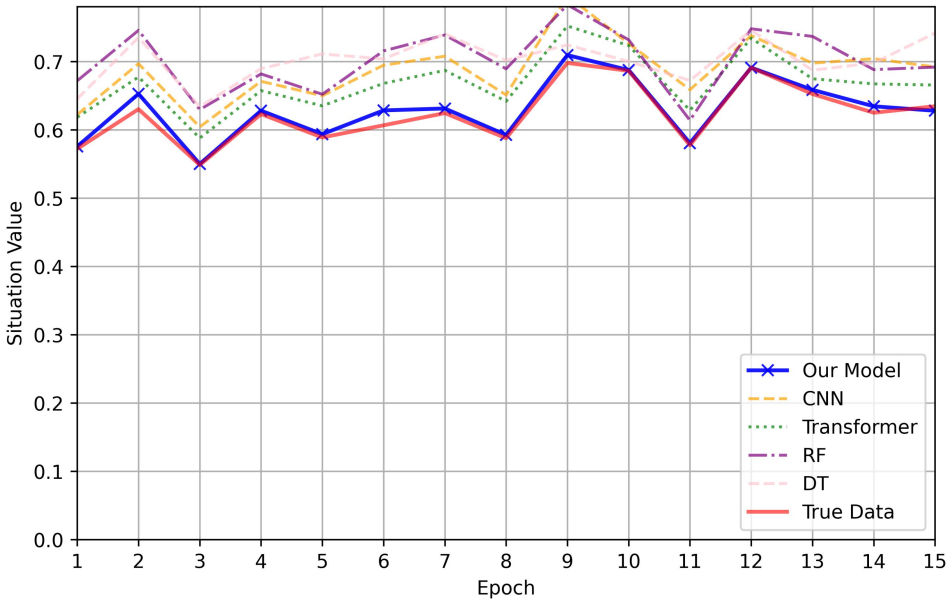


(b) Large-scale network

Fig. 4. The situation value of smart home in (a) small-scale and (b) large-scale networks.

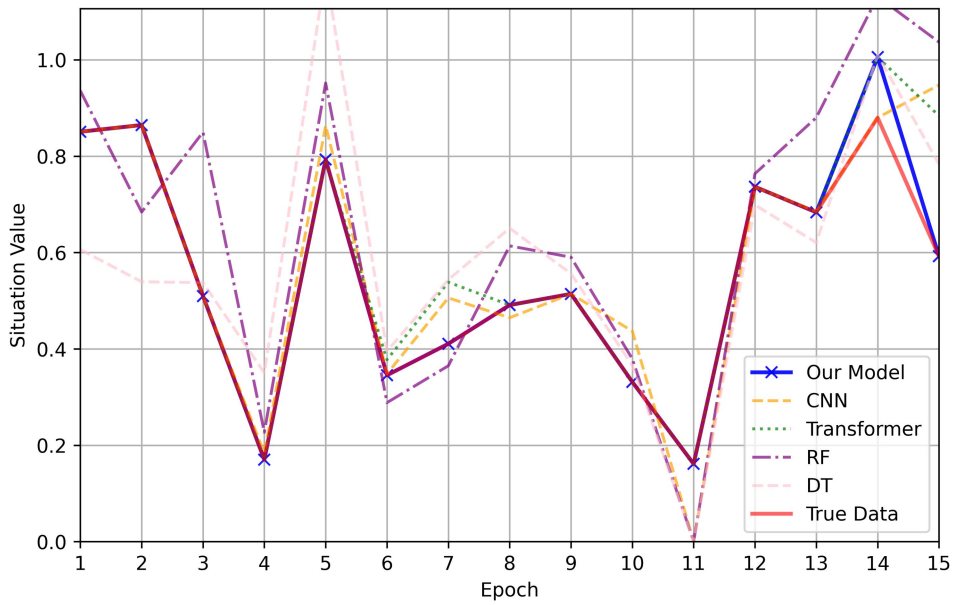


(a) Small-scale network

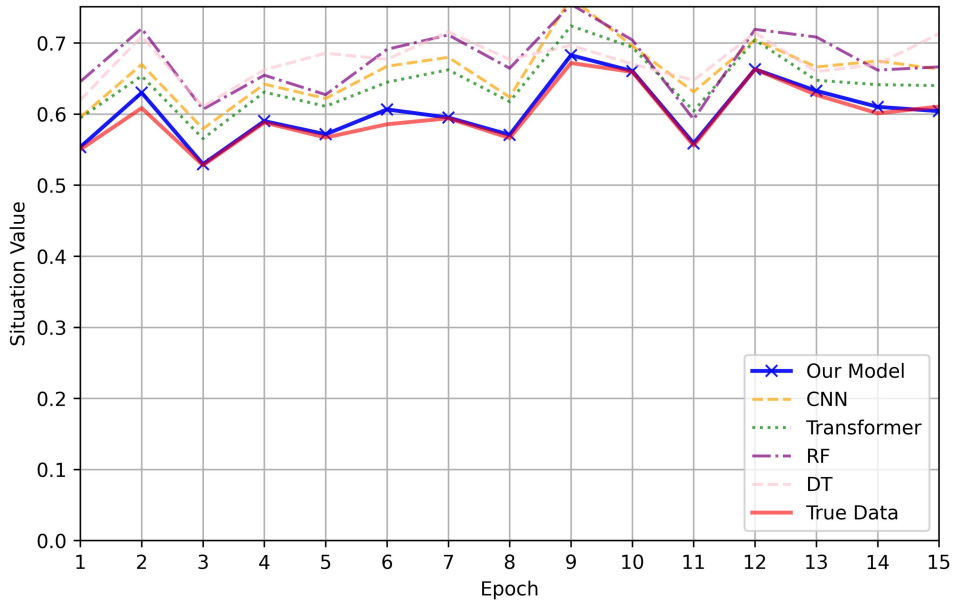


(b) Large-scale network

Fig. 6. The situation value of smart city in (a) small-scale and (b) large-scale networks.



(a) Small-scale network



(b) Large-scale network

Fig. 6. The situation value of smart agriculture in (a) small-scale and (b) large-scale networks.

TABLE XIII
RECALL RATES FOR EACH CLASS IN EACH MODEL AND OVERALL ACCURACY, RECALL, PRECISION, AND F_1SCORE

	Situation Assessment Strategy	Intrusion Detection	Adaptability	Scalability
[6]	Analytic Hierarchy Proccess	×（Not involved）	×（Poor adaptability）	×（No scalability）
[7]	Fuzzy Logic-Based Risk Assessment	×（Not involved）	×（Sensitive to environmental changes）	×（No scalability）
[8]	Hidden Markov Chain and Attack Graph	√（mRMR-Ig-based feature selection）	√（Attack path modeling）	√（Rule-based extensibility）
[9]	Multi-Dimensional Data Fusion	√（RBF neural network）	×（Weak generalization）	×（No scalability）
[10]	Fusion of Attack Probability and Impact	×（Not involved）	×（Limited expressiveness）	×（No scalability）
[11]	CIA Indicator-Based Assessment	√（Adversarial deep learning）	√（Certain adaptability）	×（Limited extensibility）
Our	Joint Modeling of Attacks, Traffic, and Impact	√（TCN + TPA + Gating Mechanism）	√（Adaptable to complex environments）	√（Modular design, easy to extend）

subtle changes can affect the se- curity level assessment to some extent. In Fig. 4–6(a), during the 5th round, the smart home and smart city environments were assessed as being in a "Super risk" state, while the smart agricul- ture environment was rated as "High risk." In the 15th round, the smart home and smart city were rated as "High risk," and smart agriculture as "Medium risk." In Fig. 4–6(b), during the 4th and 7th rounds, both the smart home and smart city were again as- sessed as "High risk," whereas the smart agriculture environment remained at "Medium risk." From an overall perspective, as the scale of the network increases, the system is subjected to more frequent and severe attacks. However, the fluctuation range of the situation value exhibits a decreasing trend. This phenomenon suggests that the proposed situation assessment method demonstrates strong adaptability and stability across different network environments. It effectively reflects the security posture of the system under high- load and high-risk conditions.

VI. Discussion

From the above experiment, it can be seen that our proposed net- work security situation assessment method has good applicability to the complex network environment and attacks in real life. How- ever, in practice, it is necessary to further extend the assessment method based on actual networks. For example, based on user be- havior analysis, a normal operation may be extremely similar to an attack behavior. After analysis by relevant security personnel, the impact of the attack should be further adjusted to achieve a customized situation assessment effect.

To further demonstrate the strengths of our method, we provide a comparative analysis with several representative existing ap- proaches in TABLE XIII. The comparison is based on four dimensions: the strategy for situation assessment, whether intrusion detection is involved, adaptability to dynamic environments, and scalability of the assessment framework. As shown in the table, traditional methods such as AHP and fuzzy logic focus primarily on static risk evaluation and lack both adaptability and extensibility. Although some recent methods incorporate machine learning or attack graph modeling, they still suffer from limited scalability or weak adapt- ability. In contrast, our method integrates attack characteristics, network traffic, and impact information through a modular deep learning architecture, achieving significant improvements in accu- racy, adaptability, and scalability.

VII. Conclusion

To address the limitations of traditional posture assessment methods, we propose a multi-dimensional network security posture assessment method based on deep learning. This approach combines intrusion detection, attack impact, and environmental requirements to form a customizable network security posture assessment method. The proposed model in intrusion detection significantly enhances the performance of the model by extracting the temporal features in the traffic through TCN, TPA extracts the local features in different time periods, and the gating mechanism adaptively adjusts the weights of both.

In future work, we plan to optimize the model with complementary optimization only to improve the model's ability to recognize a small number of classes to enhance the model's performance. We will also simulate the addition of other environmental requirements, such as user behavior analysis as mentioned above, to further extend the posture assessment methodology and promote the development of cybersecurity posture assessment in applications.

References

[1] W. S. Admass, Y. Y. Munaye, and A. A. Diro, "Cyber security: State of the art, challenges and future directions," Cyber Security and Applications, vol. 2, p. 100031, 2024.

[2] D. K. Alferidah and N. Z. Jhanjhi, "A review on security and privacy issues and challenges in internet of things," International Journal of Computer Science and Network Security, vol. 20, no. 4, pp. 263–286, 2020.

[3] W. Wu and C. Y. Yang, "An overview on network security situation awareness in internet," International Journal of Network Security, vol. 24, no. 3, pp. 450–456, 2022.

[4] H. Alavizadeh, J. Jang-Jaccard, S. Y. Enoch, et al., "A survey on cyber situation-awareness systems: Framework, techniques, and insights," ACM Computing Surveys, vol. 55, no. 5, pp. 1–37, 2022.

[5] J. Zhang, H. Feng, B. Liu, et al., "Survey of technology in network security situation awareness," Sensors, vol. 23, no. 5, p. 2608, 2023.

[6] H. Wang, Z. Chen, X. Feng, et al., "Research on network security situation assessment and quantification method based on analytic hierarchy process," Wireless Personal Communications, vol. 102, pp. 1401–1420, 2018.

[7] M. Alali, A. Almogren, M. M. Hassan, et al., "Improving risk assessment model of cyber security using fuzzy logic inference system," Computers & Security, vol. 74, pp. 323–339, 2018.

[8] Z. Li, H. Liu, and C. Wu, "Computer network security evaluation method based on improved attack graph," Journal of Cyber Security Technology, vol. 6, no. 4, pp. 201–215, 2022.

[9] Z. Chen, X. Yang, and Y. Zhu, "Research on hierarchical network security situation awareness data fusion method in big data environment," Journal of Cyber Security Technology, vol. 8, no. 1, pp. 31–52, 2024.

[10] H. Zhang, C. Kang, and Y. Xiao, "Research on network security situation awareness based on the LSTM-DT model," Sensors, vol. 21, no. 14, p. 4788, 2021.

[11] H. Yang, R. Zeng, G. Xu, et al., "A network security situation assessment method based on adversarial deep learning," Applied Soft Computing, vol. 102, p. 107096, 2021.

[12] H. Kamal Idrissi and A. Kartit, "Network intrusion detection using combined deep learning models: Literature survey and future research directions," IAENG International Journal of Computer Science, vol. 51, no. 8, pp. 998–1010, 2024.

[13] L. Wuke, Y. Guangluan, and C. Xiaoxiao, "Application of deep extreme learning machine in network intrusion detection systems," IAENG International Journal of Computer Science, vol. 47, no. 2, pp. 136–143, 2020.

[14] I. O. Lopes, D. Zou, I. H. Abdulqadder, et al., "Network intrusion detection based on the temporal convolutional model," Computers & Security, vol. 135, p. 103465, 2023.

[15] Z. Wu, H. Zhang, P. Wang, et al., "RTIDS: A robust transformer-based approach for intrusion detection system," IEEE Access, vol. 10, pp. 64375–64387, 2022.

[16] Z. Yang, D. Yang, C. Dyer, et al., "Hierarchical attention networks for document classification," Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489, 2016.

[17] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[18] K. Cho, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv:1406.1078, 2014.

[19] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," ICISSp, vol. 1, pp. 108–116, 2018.

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, et al., "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
[21] B. Hammi, S. Zeadally, R. Khatoun, et al., "Survey on smart homes: Vulnerabilities, risks, and countermeasures," Computers & Security, vol. 117, p. 102677, 2022.
[22] S. Sengan, V. Subramaniyaswamy, S. K. Nair, et al., "Enhancing cyber–physical systems with hybrid smart city cyber security architecture for secure public data-smart network," Future Generation Computer Systems, vol. 112, pp. 724–737, 2020.
[23] M. Gupta, M. Abdelsalam, S. Khorsandroo, et al., "Security and privacy in smart farming: Challenges and opportunities," IEEE Access, vol. 8, pp. 34564–34584, 2020.
[24] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," arXiv:2010.16061, 2011.
[25] T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.
[26] A. Liaw and M. Wiener, "Classification and regression by randomForest," R News, vol. 2, no. 3, pp. 18–22, 2002.
[27] W. Y. Loh, "Classification and regression trees," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 1, no. 1, pp. 14–23, 2011.
[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
[29] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[30] P. Sun, P. Liu, Q. Li, et al., "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," Security and Communication Networks, vol. 2020, Article ID 8890306, 2020.
[31] J. Wang, C. Si, Z. Wang, et al., "A new industrial intrusion detection method based on CNN-BiLSTM," Computers, Materials & Continua, vol. 79, no. 3, pp. 3897–3914, 2024.
[32] S. R. Dronadula, S. S., D. Gandhi, et al., "An innovative approach and evaluation of contemporary intrusion detection systems," Journal of Cyber Security Technology, vol. 8, no. 1, pp. 1–44, 2024.
[33] J. Zhao, Y. Liu, Q. Zhang, et al., "CNN-AttBiLSTM mechanism: A DDoS attack detection method based on attention mechanism and CNN-BiLSTM," IEEE Access, vol. 11, pp. 136308–136317, 2023.
[34] B. Cao, C. Li, Y. Song, et al., "Network intrusion detection model based on CNN and GRU," Applied Sciences, vol. 12, no. 9, p. 4184, 2022.