ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Network security situation prediction method based on attack defense tree

Xiaohui Wang[1,*], *Jun Xing*[1] *and Qianqian Shao*[1]

[1] School of Big Data, QINGDAO HUANGHAI University, Qingtao, Shandong, 266427, China

SUMMARY: *The network topology is reconstructed in real-time according to business requirements, and the data sources exhibit heterogeneity, making it difficult to handle the spatiotemporal dynamic coupling of data features, which increases the difficulty of predicting the network security situation. Therefore, this study proposes a network security situation prediction method based on attack defense tree. Firstly, security situation data is collected from multiple sources of network security data, including traffic data, log data, threat intelligence, and asset data. Through preprocessing steps such as cleaning, organizing, and standardizing, the quality and credibility of the data are improved. Then, the Lasso feature selection method is used to extract meaningful features from the preprocessed data and establish a situational assessment dataset. In the attack recognition stage, deep neural networks (DNNs) are used to accurately identify attack behaviors in the network, and Dropout regularization technology is introduced to enhance the model's generalization ability. Finally, combined with the attack defense tree model, based on the current network state and known attack information, the attack path is traversed and analyzed using reverse inference. Predict by calculating the overall vulnerability index of the system. The experimental results show that this method can effectively improve the accuracy, timeliness, and dynamism of predictions, providing strong support for network security management.*

KEYWORDS: *attack defense tree; Network security; Situation prediction; Lasso feature selection; Deep Neural Network*

1 Introduction

Against the backdrop of the continuous deepening of digitalization, the security threats faced by cyberspace are becoming increasingly complex and dynamically evolving. The successive emergence of new attack methods such as Advanced Persistent Threat (APT) and supply chain attacks has exposed the limitations of traditional defense paradigms that rely on rule matching or passive response in addressing such challenges. In this context, network security situational awareness technology has emerged, with its core being the use of multi-source data fusion and intelligent analysis methods to achieve real-time identification, dynamic evaluation, and active prediction of network threats [1, 2]. Therefore, the core goal of proposing network security situational awareness technology is to achieve real-time perception, dynamic evaluation, and active prediction of network threats through multi-source data fusion and intelligent analysis. However, existing situational awareness methods rely heavily on historical data or statistical patterns, lacking the ability for causal inference of attack paths and

*wangxh02@qdhhc.edu.cn
https://doi.org/10.65102/is2026734
Published: 30 April, 2026

dynamic optimization of defense strategies, resulting in significant limitations in prediction accuracy and defense effectiveness when facing unknown or complex attacks [3]. In view of this, comprehensive and in-depth research on network security situation prediction methods has become a core issue that urgently needs to be addressed in the current field of network security. The tackling of this issue has irreplaceable key value in enhancing the overall security protection effectiveness of cyberspace.

Currently, numerous scholars have conducted research on this. Reference [4] proposes an improved Particle Swarm Optimization Attention Bidirectional Long Short Term Memory (IPSO ABiLSTM) model for network security situation prediction. Firstly, based on the mechanism of the impact of attack behavior on the situational indicator system, generate the real situational assessment values corresponding to the original dataset, and introduce the sliding window method to reconstruct the situational values of the dataset; Next, use the IPSO algorithm to solve the problem of local optimal solutions. Finally, the IPSO ABiLSTM model was used to achieve situational prediction with different sliding window sizes. However, the network security situation is constantly changing, and this model may not be able to adapt to new situational characteristics in a timely manner when facing rapidly changing network environments, resulting in a decrease in prediction accuracy. Reference [5] proposes a targeted model based on the Internet of Things (IoT) to improve the accuracy of predicting the security status of school networks. Firstly, the data is reconstructed into a multidimensional time series, and then input into a support vector machine for training. Nonlinear methods are used to optimize the parameters of the training mode, in order to create a network security scenario prediction model. Finally, with the help of the Internet of Things, network security scenario prediction models are utilized to predict future network security. Although the simulation yields the optimal parameters for SVM, the optimal parameters may vary with changes in the network environment and data, requiring continuous re optimization. Reference [6] proposed a network security monitoring tool based on situational assessment and prediction. The evaluation module framework of this tool is based on convolutional neural networks and introduces an initial module that converts some large convolution kernels into concatenated small convolution kernels. This transformation, by concatenating multiple evaluators, can maximize the preservation of feature values and help reduce operating costs. In addition, the initial module is an optimized form of Elman neural network. The study also added delay operators to the model to respond to the temporal characteristics of network attacks. This model integrates multiple components with high complexity. Given its complex structure, it is highly likely that the model will overfit data during practical application, which will affect its generalization performance.

The attack defense tree, as an effective network security analysis tool, can intuitively describe the attacker's attack path and the defender's defense strategy. It decomposes the attack and defense process into a series of basic events and logical relationships, and displays the complete process of attack and defense by constructing a tree structure [7, 8]. The attack defense tree can not only clearly display the possible attack methods and steps that attackers may take, but also help defenders identify weak links in the system and develop targeted defense measures. Therefore, this study proposes a network security situation prediction method based on attack defense tree.

2 Design of Network Security Situation Prediction Method

To address the challenge of processing multi-source heterogeneous data in network security situation prediction, this study first collected raw information from traffic data, log data,

threat intelligence, and asset data. The data quality was improved through preprocessing steps such as cleaning, deduplication, and standard deviation normalization. The Lasso feature selection method was used to remove redundant features and construct a situation assessment dataset. On this basis, deep neural networks are used to identify network attack behaviors. Dropout regularization technology is introduced to suppress overfitting, and Softmax classifier is used to output attack types and intensities, thereby grasping the current attack situation in the network. Finally, the recognition results are integrated into the attack defense tree model, which adds defense measures to each leaf node on the basis of the traditional attack tree. Through reverse deduction, the attack path is traversed, and the vulnerability of leaf nodes and paths is quantified using multi-attribute utility theory and fuzzy analytic hierarchy process. The overall vulnerability index of the system is used to characterize the severity of the network security situation.

2.1 Data Collection and Preprocessing

This study collected security situation data from various network security data sources and gathered information on various elements related to network security. The data sources mainly include traffic data, log data, threat intelligence, and asset data. The following provides a brief explanation of the data source information.

① Traffic data: refers to the amount of data transmitted in the network during a specific time period. Usually refers to network traffic data, including network protocols, destination IP, and other information.

② Log data: refers to data that records system operational activities, events, and status information. Mainly including security devices, application logs, etc.

③ Threat intelligence: refers to the relevant information used to identify and evaluate network security threats, including threats and information about threat initiators. The main channels for obtaining threat intelligence include open source threat information and internal intelligence data.

④ Asset data refers to the information of various devices, applications, and data resources in a network system. Mainly including hardware, software, configuration, etc.

This article uses various technical methods to handle access, attacks, threats, and other data traffic inside and outside the network, including port mirroring and port splitting on switches to capture and record network traffic, as well as using traffic collection devices to perform protocol parsing, filtering, and reassembly operations on traffic packets to extract security events and device asset information and convert them into security log records and asset log records. Finally, extract network security data from security log records and asset log records.

The collected raw data often has problems such as missing values and inconsistencies, which require preprocessing [9]. Given that the collected data may contain a large amount of redundant content, duplicate entries, and irrelevant interference items, data cleaning and deduplication operations are needed to remove these noisy information and improve the quality and credibility of the data. In addition, there may be differences in format and content among data from different sources [10]. To ensure data consistency, it is necessary to classify and standardize the data. This article uses a commonly used method in data processing, namely the standard deviation normalization method, to scale all scaled data features to a unified dimension through formula (1).

$$Z = \frac{x - \mu}{\kappa} \tag{1}$$

In the formula, κ is the standard deviation and μ is the mean of all sample data.

By preprocessing the data and selecting appropriate indicators based on the indicator system, a situational assessment dataset is established. Next, extract meaningful features from the raw data to better describe the security status of the network. This article adopts the Lasso feature selection method. Lasso achieves parameter sparsity and feature selection by adding L1 regularization term to the loss function, minimizing the sum of error and regularization term. This regularization term prompts the model to select a small number of important features, reduce the coefficients of other features to 0, treat them as non significant variables, and subsequently eliminate them [11, 12]. Adjusting alpha can precisely control the complexity of the model and the degree of feature selection. The advantage of Lasso lies in its ability to automatically select features and build more concise and interpretable models.

Compared with other linear regression models, the Lasso regression model has a smaller root mean square error and can solve the problem of matrix irreversibility (such as multicollinearity) in multiple linear regression models, reducing model complexity. Assuming $X=(X^{(1)},X^{(2)},...,X^{(d)})$ is a covariate, for each $X^{(j)}$, $X^{(j)}=(x_1^{(j)},x_2^{(j)},...,x_n^{(j)})^T$, the general linear regression model is expressed as:

$$y=X\beta+\varepsilon \tag{2}$$

In the formula, y is the response vector of order $n\times 1$, $\beta\in R^d$ is the regression coefficient, and $\beta=(\beta_1,\beta_2,...,\beta_d)$. $\varepsilon\in R^d$ is the error vector.

The classic Lasso algorithm aims to minimize the sum of squared residuals as the optimization objective function and adopts L1 regularization penalty term. The mathematical expression for its optimization is:

$$J(\beta)=\sum_{i=1}^{m}(y-X\beta)^2+\lambda\sum_{j=1}^{n}\left|\beta_j\right| \tag{3}$$

In the formula, $\lambda\geq 0$ is the adjustment parameter. When the adjustment parameter is large enough, some β coefficients will be compressed to 0. $\sum_{j=1}^{n}\left|\beta_j\right|$ is the L1 regularization of the regression coefficient β. Geometrically, Lasso's optimization objective can be expressed as a constrained optimization problem. Assuming there is an n-dimensional feature space, each feature corresponds to a coordinate axis. The goal is to find a hyperplane that minimizes the data fitting error, and the L1 norm of the model parameter does not exceed the preset constant. This constraint forms a high-dimensional hypercube with the origin as the center in geometry (the boundary can be likened to a "sphere"), and the solution is at the intersection of the hypercube and the "sphere". Adjusting the constant can control the number of feature selections. If the constant is small, the constraint is strong, and Lasso tends to select fewer features, thereby achieving the effect of feature selection [13]. The geometric schematic diagram is shown in Figure 1.

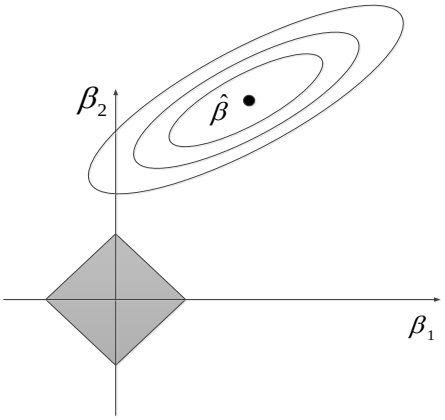

Figure 1: Lasso Regression

In the figure, β_1、β_2 is the model parameter to be optimized, the ellipse is the objective function, and the cubic region is the solution space. Each ellipse centered around $\hat{\beta}$ represents a residual sum of squares value, and each circle corresponds to the same RSS. The tangent point between the ellipse and the cube region is the optimal solution of the objective function. It is observed that the optimal solution of Lasso tends to produce tangent points on the coordinate axis, which enables Lasso regression to produce sparse solutions, with some coefficients being zero. In Lasso regression, the most commonly used model is LassoCV, which is a lasso linear model with iterative fitting along the regularization path. It has a particularly obvious advantage in finding the main features from high-dimensional features. Cross validation is used for hyperparameter α to ultimately select the optimal regularization parameter α. The processed data is then input into the above model to select representative features.

2.2 Network Attack Identification Based on DNN

After completing data collection and preprocessing, a feature dataset suitable for analysis was obtained. Next, use this data to identify attack behaviors in the network, as accurate identification of attacks is a key prerequisite for predicting network security situations. Only by understanding the types and strengths of attacks currently present in the network can we more accurately predict the future security situation. Deep neural networks (DNNs), with their powerful nonlinear modeling capabilities and self supervised feature representation learning paradigm, have constructed a complete mapping framework for high-dimensional complex function spaces and can recursively extract multi-level semantic features from raw data manifolds. Therefore, they have broad application prospects in network attack recognition [14, 15]. The structure of the DNN model is shown in Figure 2.

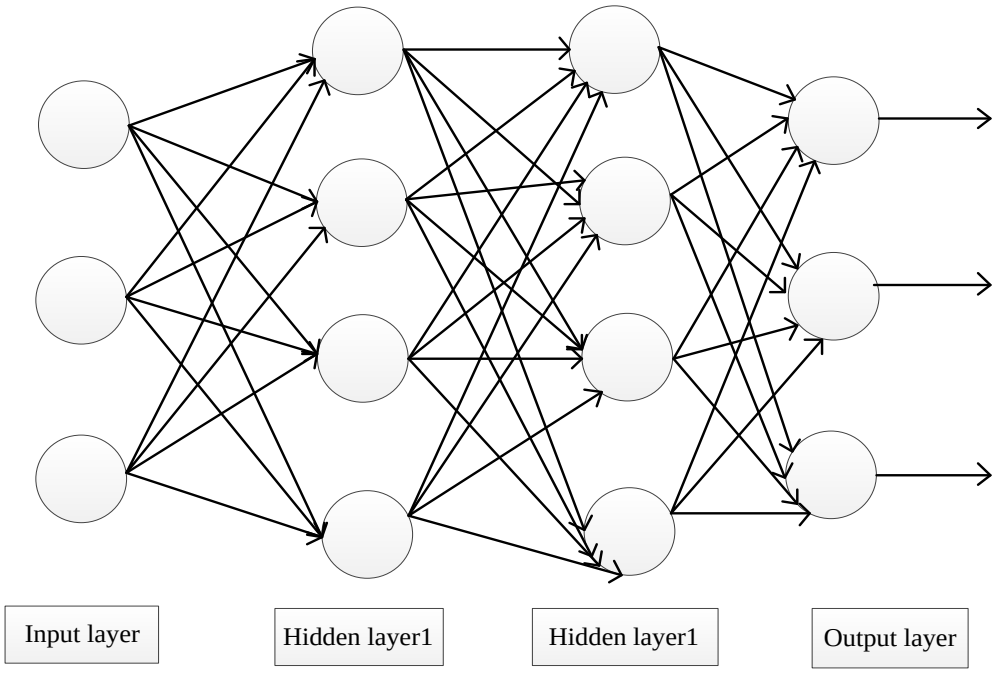

Figure 2: DNN Model Structure

In Figure 2, the input layer located at the bottom layer is responsible for the key task of exchanging information with the external environment, and it can capture and receive various types of feature information from the external environment. The number of neurons contained in the input layer is not arbitrarily set, but strictly consistent with the dimensions of the input data. This consistency is an important prerequisite for ensuring that the model can accurately and effectively process the input data; The number of hidden layers and neurons can be flexibly set according to the specific requirements of the task and the complexity of the model, which together determine the network's expressive and learning abilities; The number of output layer neurons in the last layer is directly related to the specific requirements of the task, and the final results correspond to their respective classifications. If σ is the activation function, the basic formula is as follows:

$$z = wx^T + b \tag{4}$$

$$a = \sigma(z) \tag{5}$$

In the formula, $x = [x_1, x_2, ..., x_n]$ is the input vector, and each x_i is an input data; w is the weight size corresponding to each input data; b is the bias value.

When the model complexity is too high, it is easy to fit the details and noise of the training set, and cannot generalize to new data, resulting in excellent performance of the model on the training set but significant performance decline on new or test data. Therefore, this article adopts Dropout to alleviate the overfitting problem of the model [16]. During each training iteration, randomly selecting the output of some neurons based on a preset probability p and setting it to 0 helps to weaken the dependency relationships between neurons, learn more robust and independent features, enhance generalization ability without changing the depth of the network. Its structure is shown in Figure 3.

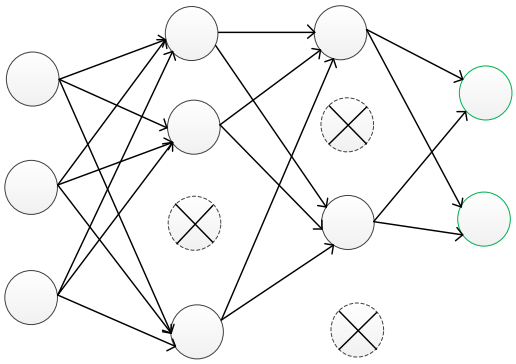

Figure 3: Dropout Structure Diagram

The forward propagation formula for neural networks without Dropout added is:

$$\begin{cases} z_i^{(n+1)} = w_i^{(n+1)} y^{(n)} + b_i^{(n+1)} \\ y_i^{(n+1)} = f(z_i^{(n+1)}) \end{cases} \tag{6}$$

The forward propagation formula of the neural network with Dropout added is formula (7), and compared to not adding Dropout, its output vector undergoes Bernoulli distribution.

$$\begin{cases} r_j^{(n)} \sim Bernoulli(p) \\ \tilde{y}^{(n)} = r^{(n)} * y^{(n)} \\ z_i^{(n+1)} = w_i^{(n+1)} \tilde{y}^{(n)} + b_i^{(n+1)} \\ y_i^{(n+1)} = f(z_i^{(n+1)}) \end{cases} \tag{7}$$

Softmax classifier can be regarded as a generalized form of logistic regression classifier when dealing with multi classification problems, with a concise calculation process and easy to understand results [17, 18]. The main function of the Softmax classifier is to map the neuron values of the output layer to real numbers in the (0,1) interval, and normalize them to a sum of 1, where the sum of probabilities for each category is also 1. During the training process, the Softmax classifier optimizes the classification performance of the model by adjusting parameters to maximize the probability of correct categories and minimize the probability of incorrect categories. The function that maps input to predicted results is the hypothesis function in machine learning algorithms. For linear classification, it is assumed that the function includes two parts: calculating the results through linear operations and converting the results into explanatory predictions. If P is the probability that x belongs to category N, then the assumption function of the Softmax classifier can be expressed as:

$$h_\theta(x) = \begin{bmatrix} P(y=1)\big|x;\theta \\ P(y=2)\big|x;\theta \\ \vdots \\ P(y=N)\big|x;\theta \end{bmatrix} = \frac{1}{\sum_{i=1}^{N} \exp(\theta_i^T x)} \begin{bmatrix} \exp(\theta_1^T x) \\ \exp(\theta_2^T x) \\ \vdots \\ \exp(\theta_i^T x) \end{bmatrix} \tag{8}$$

Select the category with the highest probability as the final classification result for the

sample.

Thus, complete the identification of network attacks based on DNN.

2.3 Network security situation prediction based on attack defense tree

By identifying network attacks based on DNN, the current attack situation in the network has been grasped. However, simply understanding the current attacks is not enough, it is also necessary to predict the future cybersecurity situation in order to take preventive measures in advance.

Attack defense tree is a graphical model used to describe attack scenarios and defense strategies, which can clearly display the paths and possible impacts of attacks, as well as corresponding defense measures. Therefore, in this study, the results of network attack identification based on DNN will be combined with attack defense trees for security situation prediction. Find the corresponding attack path and node in the attack defense tree for the identified attack type. Predict the severity of the current network security situation based on factors such as the complexity of attack paths, the intensity and frequency of attacks, and the effectiveness of defense measures.

The attack defense tree is an extension of the attack tree, which is a method of modeling different attacks that industrial control systems may face. It uses an inverted tree structure to represent various attack methods, with the root node being the attacker's target; The leaf node is at the end, indicating the attack method or event [19]. A complete attack refers to the process in which an attacker reaches the root node from a leaf node through other intermediate nodes in the attack tree. By traversing the attack leaf nodes to the root node, all attack paths can be obtained, where not all intermediate nodes can be propagated upwards by a single attack. In addition to leaf nodes, intermediate nodes in attack trees are generally classified as AND nodes, OR nodes, or SAND nodes.

The use of attack trees for network security situation analysis is a reverse deduction process. Firstly, identify an important security event in the network and use it as the root node of the attack tree; Then, establish an attack tree and analyze the necessary conditions for root node events to occur, that is, which attack method and path generate this security event; Finally, quantify the probability of root node occurrence, identify the most likely means for attackers to take, obtain the vulnerability of the network, and thus achieve network security situation prediction.

The attack defense tree extends the leaf nodes by adding one or several defense measures to each leaf node, as shown in Figure 4.

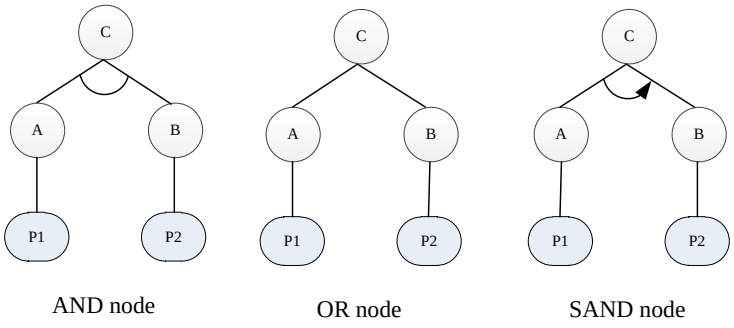

Figure 4: Attack Defense Tree Model

In the figure, node C represents the root node, which is the attack target, while nodes A and B are leaf nodes, referring to the attack method. Unlike traditional attack trees, P1 and P2

are added after the leaf nodes. P1 and P2 represent the most effective defense measures against the attack method, so that corresponding defense schemes can be found for each different attack method, which can better depict the network's attack and defense process.

When building an attack defense tree, the first step is to determine the attack target, then analyze the possible attack steps and methods that the attacker may take, and use these steps and methods as nodes of the tree, connecting them with branches to form an attack path. At the same time, label corresponding defense measures and defense effects on each node. When predicting network security situation based on attack defense tree, the attack defense tree can be traversed and analyzed according to the current network status and known attack information. Assess the security risks faced by the network by calculating the vulnerability of each attack path.

The vulnerability of leaf nodes mainly depends on three dimensions: the difficulty of the attack required by the attacker, the possibility of the attack behavior being detected by the detection system, and the severity of the consequences once the attack is successful. The greater the difficulty of the attack, the more obstacles the attacker faces in successfully implementing the attack, and the corresponding vulnerability of the leaf node is lower; The higher the probability of an attack being discovered, the more helpful it is to block the attack process in a timely manner, thereby reducing vulnerability; The more severe the consequences of the attack, the higher the vulnerability of the leaf nodes.

Based on the above relationship, this study adopts the multi-attribute utility theory to unify the three dimensions into the quantitative evaluation framework of leaf node vulnerability. Let $V(A_n)$ and $V'(A_n)$ be the vulnerability indicators of the n-th leaf node without and with defense measures, respectively. The formula for calculating vulnerability based on these factors is as follows:

$$V(A_n)= w_{dif}\times U(dif_n) + w_{poss}\times U(poss_n) + w_{imp}\times U(imp_n) \tag{9}$$

$$V'(A_n)= [w_{dif}\times U(dif_n) + w_{poss}\times U(poss_n) + w_{imp}\times U(imp_n)]\times (1-\frac{t_n}{m}) \tag{10}$$

In the equation, $U(\cdot)$ is the utility function, n is the number of leaf nodes, dif_n, $poss_n$ and imp_n are the difficulty of attacking the n-th leaf node, the likelihood of being discovered during the attack, and the severity of the consequences. w_{dif}, w_{poss} and w_{imp} are the weights of the three, and t_n is the number of defense measures at the n-th leaf node.

In the attack tree model, the destruction targets for industrial control systems often correspond to multiple different attack paths. To determine the overall vulnerability index of the system, this study calculates the vulnerability values of each attack path and selects the maximum value from them. The vulnerability of the entire system is characterized by the vulnerability of the path where the maximum value is located. On this basis, when recursively deducing the vulnerability of each attack sequence from bottom to top in the attack tree, the vulnerability data of leaf nodes is the basis for calculation. In this process, the nodes involved are usually divided into the following two categories:

① AND node: AND node indicates that all direct child node events must occur simultaneously to achieve the attack target. The absence of any child node cannot trigger the attack on the parent node, meaning that all steps on the attack path must be met and cannot be omitted. When the node type is AND, the vulnerability value of its upper layer attack sequence is equal to the product of the vulnerabilities of all direct sub branches of the node, as

shown in the following formula:

$$V(s_i) = V(A_{i1}) \times V(A_{i2}) \times \cdots \times V(A_{im}) \tag{11}$$

In the formula, s_i represents the i-th attack sequence, A_{im} represents a subset under the AND node.

② OR node: The OR node indicates that the occurrence of any direct child node event can lead to the target of the parent node's attack. The attack path is selective, and the attacker only needs to choose a feasible means to achieve the goal. When a node is of OR type, in the process of recursion to the upper layer, the vulnerability of the attack sequence is equal to the maximum vulnerability value among its subordinate branches, as shown in the following formula:

$$V(s_i) = \max\{V(A_{i1}), V(A_{i2}), \cdots, V(A_{im})\} \tag{12}$$

Attack trees usually only depict attack paths, while attack defense trees also consider defense measures based on the attack tree. After adding protective measures to the leaf node, the vulnerability of the node will decrease, thereby reducing the vulnerability of the entire system. To quantify the vulnerability of leaf nodes, it is necessary to clarify the specific values of three influencing factors in advance: the difficulty of the attacker's attack, the likelihood of the attack behavior being captured by the detection system, and the magnitude of the impact once the attack is successful. For these three dimensions, this study designed a set of grading rules and standards, as shown in Table 1.

Table 1: Rating Criteria

Difficulty of Attack	Rating	Probability of being discovered	Rating	The consequences generated	Rating
extremely difficult	5	extremely difficult	1	serious	1
difficult	4	difficult	2	strong	2
ordinary	3	ordinary	3	centre	3
simple	2	simple	4	weak	4
extremely easy	1	extremely easy	5	Very weak	5

When calculating the utility value of attributes, for the convenience of calculation, it can be assumed that they are inversely proportional, that is, $U(X) = \frac{1}{X}$. Then, as long as the weights of each of the three attributes are calculated, the vulnerability index of each leaf node can be obtained through formula (9) [20].

The calculation of weight values in this article adopts the fuzzy analytic hierarchy process, and the calculation steps are as follows:

Firstly, by using expert scoring, the importance of each two elements to the previous layer element is compared, and a fuzzy judgment matrix $R = (r_{ij})_{3\times3}$ is established for each leaf node.

Then, according to formulas (13) and (14), the fuzzy judgment matrix is uniformly transformed to obtain the fuzzy consistent judgment matrix.

$$r_i = \sum_{k=1}^{n} r_{ik} \tag{13}$$

$$f_{ij} = \frac{r_i - r_j}{2n} + 0.5 \tag{14}$$

In the formula, f_{ij} is the element in the i-th row and j-th column of the transformed fuzzy consistent matrix.

Finally, calculate each weight value in the fuzzy consensus judgment matrix according to formula (15), and verify whether the sum of the calculated weight values is equal to 1. If it is equal to 1, it is correct.

$$w_i = \frac{1}{n} - \left(\frac{1}{2} + \frac{1}{n}\right) \times \frac{1}{\alpha} \times \sum_{j=1}^{n} f_{ij} \tag{15}$$

Based on the vulnerability values of leaf nodes and the hierarchical structure of the attack defense tree, the vulnerability level of each attack path can be calculated one by one. Select the path with the highest vulnerability and use its value as the final measure of the network's vulnerability.

By calculating the overall vulnerability index of the system, it is possible to predict the network security situation. When the vulnerability index of the system is high, it indicates that the system is facing significant security threats and the network security situation is severe; When the vulnerability index of the system is low, it indicates that the security of the system is high and the network security situation is relatively good. Risk managers can develop corresponding protection plans based on the vulnerability indicators of the system, such as adding defense measures, strengthening security monitoring, etc., to reduce the vulnerability of the system and improve the level of network security.

3 Experiments and Results Analysis

3.1 Experimental Plan

The experimental operating platform and environment are shown in Table 2.

Table 2: Experimental Operation Platform and Environment

Operating Platform	Operating system	Windows10
	System memory	64G
	CPU	AMD Ryzen 7 6800H
	GPU	NVIDIA GeForce RTX 3060
Operating environment	Programming language	Python3.10
	Deep Learning Framework	PyTorch

In order to promote research in the field of network security, the experiment used the UNSW-NB15 dataset. This dataset contains 2540044 pieces of data, divided into 4 CSV files. The dataset simulates real network traffic and attack behavior, covering 9 types of network attacks and providing 49 features such as protocol type, target IP address, target port, duration, etc. The 9 types of attacks in the UNSW-NB15 dataset are as follows:

① Fuzzers: Attackers scan and exploit security vulnerabilities in the target system by sending random packets or sequences.

② Analysis: Attackers use scanning tools to perform port scans on the target system in order to obtain information about the target system.

③ Backdoors: Attackers install malicious programs on the target system and use them for remote control to obtain sensitive data of victims.

④ DoS: Denial of Service attack, where the attacker sends a large number of invalid requests, causing the target system to be unable to process legitimate requests.

⑤ Exploits: Attackers exploit known vulnerabilities to gain access to the target system.

⑥ Generic: An attack that can be carried out on all block ciphers, regardless of their structure.

⑦ Reconnaissance: an attack that uses probes to steal network and host information.

⑧ Shellcode: An attack in which attackers exploit vulnerabilities in the target system to inject malicious code.

⑨ Worms: Attacks that can self replicate and infect other computers through networks or email.

The experiment was conducted using the Reference [4] method, Reference [5] method, Reference [6] method, and the proposed method. The effectiveness of different methods was verified by comparing their accuracy, prediction time, and prediction mean square error in predicting network security situations. Among them:

a. Mean square error (MSE) of network security situation prediction: Quantify the size of prediction error by calculating the average square of the difference between the predicted value and the true value. The smaller its value, the closer it is to the true value. The calculation formula is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{16}$$

In the formula, n is the number of samples, y_i is the true value, and $\hat{y}_i$ is the predicted value.

b. Network security situation prediction time: Network security situation prediction time refers to the time required from input data to output prediction results. Real time is crucial in the field of cybersecurity. The shorter the prediction time, the faster the response to potential threats.

c. Attack Transition Probability Prediction Error (ATPE): This indicator can evaluate the predictive ability of different methods for attacker strategy adjustment, and is suitable for dynamic game scenarios. The smaller the ATPE, the stronger the adaptability of the method to changes in attacker behavior.

3.2 Experimental result

(A) Mean square error of network security situation prediction

The comparison results of the mean square error of network security situation prediction after applying different methods are shown in Figure 5.

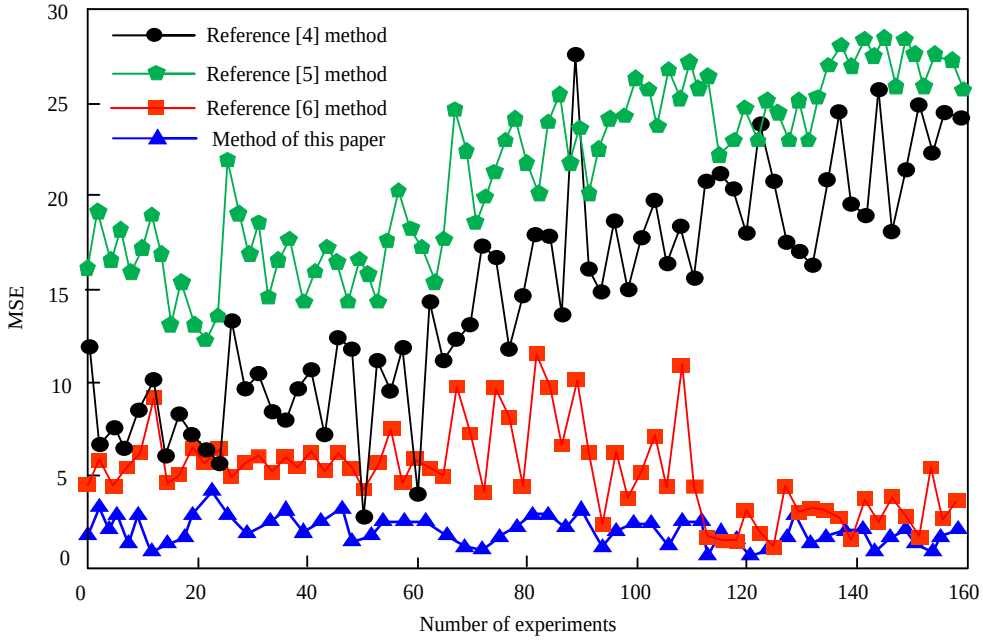

Figure 5: Test results of mean square error in network security situation prediction

From Figure 5, it can be seen that the MSE value of the network security situation prediction results obtained by applying the method proposed in this paper is below 5, while the MSE values of the comparison methods are higher than those of the method proposed in this paper. This indicates that the predicted values of the method proposed in this paper are closer to the true values and have better prediction performance. This is mainly due to the synergistic effect of the anti deduction mechanism of the attack defense tree model in this article's method and the recognition of DNN attacks. The attack defense tree dynamically traverses the attack path based on the current attack state, quantifies vulnerability using multi-attribute utility theory, and accurately reflects real-time threat evolution; DNN improves the robustness of attack recognition through Dropout regularization and feature selection optimization. The synergistic effect of the two effectively captures the spatiotemporal dynamic characteristics of the network state, thereby greatly improving prediction accuracy.

(B) Prediction time of network security situation

The test results of the prediction time of network security situation after applying different methods are shown in Table 3.

Table 3: Test results of network security situation prediction time/s

Number of experiments	Reference [4] method	Reference [5] method	Reference [6] method	Proposed method
10	3.2	4.1	5.3	2.5
20	4,3	5.1	6.0	3.2
30	5.4	6.3	7.5	3.7
40	6.2	5.4	8.6	2.9
50	5.8	8.5	7.3	4.3
60	7.8	6.7	6.0	3.8
70	3.8	4.0	5.3	2.4
80	4,9	5.3	6.1	3.1
90	5.7	5.3	7.6	2.6
100	6.6	6.4	8.3	2.9

According to the analysis of the data in Table 3, the prediction time of the network security situation using the method in reference [4] is between 3.2 and 6.6 seconds; The prediction time of the method in reference [5] is between 4.1 and 8.5 seconds; The prediction time of the method in reference [6] is between 5.3 and 8.6 seconds; And the prediction time of the method in this article is less than 5 seconds. After comparison, it can be seen that the proposed method has a shorter prediction time for network security situation, indicating that this method can detect potential threats faster. This is because the Lasso feature selection method in this article eliminates a large number of redundant features, reducing the computational complexity of redundant data. Meanwhile, with the introduction of Dropout in DNN, the model training becomes more stable and can quickly converge without the need for repeated parameter tuning. The attack defense tree only traverses the relevant branches, avoiding global search and reducing overall computational complexity.

(C) Prediction error of attack transition probability

The test results of the prediction error of attack transition probability after applying different methods are shown in Figure 6.

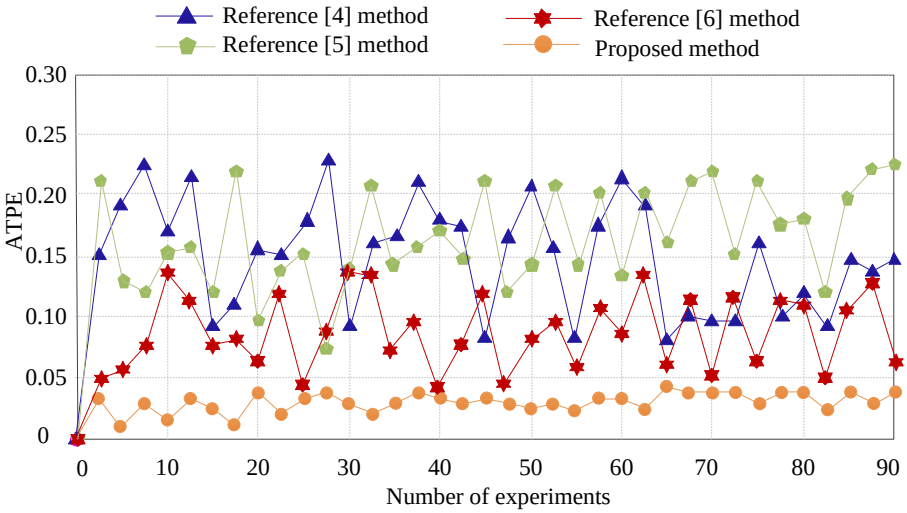

Figure 6: Test results of prediction error for attack transition probability

From the results in Figure 6, it can be seen that the minimum prediction error of attack transition probability for Reference [4] method is 0.08, the minimum prediction error of attack transition probability for Reference [5] method is 0.07, and the minimum prediction error of attack transition probability for Reference [6] method is 0.04. However, after applying the proposed method, the prediction error of attack transition probability always exceeds 0.04. After comparison, it can be seen that the proposed method has a lower prediction error for attack transition probability, indicating that this method has stronger adaptability to changes in attacker behavior. The core of the proposed method's lower prediction error for attack transfer behavior lies in the dynamic adaptability of the attack defense tree. The defense tree quantifies and adjusts path vulnerability in real-time through leaf node defense measures, and combines fuzzy analytic hierarchy process to dynamically update attribute weights, which can quickly respond to changes in attack strategies. The recognition results of DNN are continuously input into the defense tree model, forming a closed-loop feedback mechanism that closely matches the evolution of attacker behavior, significantly better than static model driven comparison methods.

4 Conclusion

The network security situation prediction method based on attack defense tree proposed in this article, after collecting and preprocessing multi-source data, combines Lasso feature selection to improve data quality and provide reliable support for subsequent analysis. Then, the powerful feature learning and nonlinear fitting capabilities of DNN are utilized to achieve efficient network attack recognition, and Dropout regularization and Softmax classifier are used to optimize model performance, enhancing generalization and classification accuracy. Based on this, an attack defense tree model is constructed to combine attack paths with defense measures. Multi attribute utility theory and fuzzy analytic hierarchy process are used to quantitatively evaluate the vulnerability of leaf nodes and attack paths, achieving accurate quantitative prediction of network security situation.

This method integrates DNN attack recognition results with attack defense tree analysis, comprehensively and dynamically evaluates the network security situation, provides scientific decision-making basis for risk managers, and helps to formulate protection strategies in advance. The experiment shows that it has significant advantages in improving the real-time, accuracy, and dynamism of situational awareness. In the future, models and data processing strategies can be further optimized to adapt to complex network security environments and continuously improve protection levels.

Funding

Research Project on Undergraduate Teaching Reform of Shandong Provincial Department of Education: Research on the "Mentorship + Project-Based" Talent Cultivation Model for Computer-Related Majors in Applied Universities under the Credit System M2022328

References

[1] G Z Zhang. Simulation of network confusion attack defense based on reinforcement learning algorithm [J]. Computer Simulation, 2024, 41 (12): 462-466.

[2] H Gao, L Guo. Research on network security situation prediction algorithm combining intuitionistic fuzzy sets and deep neural networks[J].SAE International Journal of Connected and Automated Vehicles, 2024,7(3):341-353.

[3] Y Wu, C Shen, S Xiao, et al. A novel research on network security situation prediction based on iteratively optimized RBF-NN[J].Scientific Reports, 2025,15(1):1-17.

[4] Y X Wu, D M Zhao. Build IPSO-ABiLSTM model for network security situation prediction[J].Journal of Information Science & Engineering, 2024, 40(1):71-88.

[5] W Yan, L Qiao, S Krishnapriya, et al. Research on prediction of school computer network security situation based on IOT[J].International Journal of System Assurance Engineering and Management, 2022, 13(Mar. Suppl.1):S488-S495.

[6] L Zhang, Y Liu. Network security prediction and situational assessment using neural network-based method[J].Journal of Cyber Security And Mobility, 2023, 12(4):547-568.

[7] Abolfathi M, Inturi S, Jafarian B K H .Toward enhancing web privacy on HTTPS traffic:

A novel SuperLearner attack model and an efficient defense approach with adversarial examples[J].Computers & Security, 2024, 139(Apr.):1-14.

[8] Lopuhaa-Zwakenberg M, Budde E C, Stoelinga M. Efficient and Generic Algorithms for Quantitative Attack Tree Analysis[J].IEEE Transactions on Dependable and Secure Computing,2023,20(5):4169-4187.

[9] Wentao L. Construction and analysis of QPSO-LSTM model in network security situation prediction[J].Journal of Cyber Security And Mobility, 2024, 13(3):417-438.

[10] Xu M, Liu S, Li X. Network security situation assessment and prediction method based on multimodal transformation in edge computing[J].Computer Communications, 2024, 215(Feb.):103-111.

[11] C J Li, Y Wang, X H Yuan. Sparse estimation of functional logistic additive models based on adaptive group Lasso[J].Statistics & Decision,2025,41(24):64-69.

[12] Z Y Lai, T H Wang, X Zhang. Feature weighted support vector machine based on HSIC Lasso[J].Computer and Modernization,2025(7):119-126.

[13] Yang J, Xu Y, Wang B, et al. Group Lasso based redundancy-controlled feature selection for fuzzy neural network[J].Optoelectronics Letters, 2023,19(5):284-289.

[14] Alsaidi A A A S, Mohammed J H, Ogaili A N R R, et al. HawkPhish-DNN cybersecurity model: adaptive hybrid optimization and deep learning for enhanced multi-objective phishing URL detection[J].International Journal of Information Technology, 2025, 17(7): 3859-3875.

[15] Gao T, Yang J, Wang C, et al. A smoothing Group Lasso based interval type-2 fuzzy neural network for simultaneous feature selection and system identification[J]. Knowledge-Based Systems, 2023, 280(Nov.25):1-11.

[16] Poornima C L, Rao C S, Varma D N. Predicting weld quality in duplex stainless steel butt joints during laser beam welding: a hybrid DNN-HEVA approach[J].Journal of Advanced Manufacturing Systems, 2024, 23(4):801-836.

[17] Li R, Wang H, Dai H, et al. Accurate state of charge prediction for real-world battery systems using a novel dual-dropout-based neural network[J].Energy, 2022, 250(Jul.1): 1-13.

[18] Qi Z, Zhoupu W. A Network Intrusion Detection Method for Information Systems Using Federated Learning and Improved Transformer[J].International Journal on Semantic Web and Information Systems (IJSWIS),2023,20(1):1-20.

[19] Xu J, Zhang F, Jin W, et al. A deep investigation on stealthy DVFS fault injection attacks at DNN hardware accelerators[J].Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 2025, 44(1):39-51.

[20] Aijaz M, Nazir M. Modelling and analysis of social engineering threats using the attack tree and the Markov model[J]. International Journal of Information Technology, 2024, 16(2): 1231-1238.