Campus Network Security Situation Awareness Based on AHP and Nadam Algorithm

Liwen Xu
Business School, Jiangsu Open University, China
xuliwen920@126.com

Abstract: *Background: the campus network serves as a basic communication and management platform of colleges and universities, providing convenience for campus life. However, it also faces network security issues. Aiming at the security problems brought by network threat attacks, a security situational awareness model of campus network based on Analytic Hierarchy Process (AHP) and Nadam algorithm was proposed. Methods: firstly, the improved AHP was used to build the Network Security Situation* (*NSS) assessment mode. Then, the Nadam algorithm and the improved Long Short-Term Memory (LSTM) network were used to build the NSS prediction model. Results: the results showed that the improved AHP had a good consistency in the Judgment Matrix (JM). The Fuzzy Neural Network (FNN) evaluation method, based on the improved Gravity Search Algorithm (GSA), began to converge around the 69th iteration, with a small output error of 0.0107. After 20 iterations, the fitness value stabilized at 0.13. The NSS assessment model, based on the improved AHP, achieved a high security situation value of 0.425. The mean square error of the Look ahead method, combined with the Nadam algorithm, flattened out after 80 iterations, which could increase the convergence speed of LSTM networks. The accuracy of the NSS prediction model using Nadam algorithm and improved LSTM network was the highest, up to 98%. The false positive rate and false negative rate were the lowest, at 2.64% and 11.03%, respectively. Additionally, the predicted NSS value was closest to the true value, with a Mean Absolute Percentage Error (MAPE) of 0.039 and a mean square error of 0.01. Conclusion: in summary, the constructed model in this study has good application effects in NSS awareness, and has certain positive significance for maintaining the campus network security.*

Keywords: *Network security, situation awareness, analytic hierarchy process, nadam algorithm, long and short-term memory network.*

Received April 2, 2025; accepted July 28, 2025
https://doi.org/10.34028/iajit/22/6/15

1. Introduction

As the growth of technology and the rapid popularization of campus networks on campus, the level of information technology in higher education institutions has been improved. Although the Internet has improved the quality of life for students, it has also brought various security issues, including hacker intrusion and network attacks [11]. Therefore, network security is increasingly valued. Network security means the protection of the software and hardware of a network system, and data from damage, alteration, or unauthorized access, while ensuring continuous and reliable operation. Security assessment is an important management tool that spans the entire life-cycle of information systems, serving as the foundation and prerequisite for formulating and adjusting security strategies. Only by fully identifying system security risks can effective security prevention measures be taken in a targeted manner. To evaluate the security situation of campus networks effectively, a comprehensive evaluation indicator system must be established first. Analytic Hierarchy Process (AHP) is a decision-making method that allows for qualitative and quantitative analysis. It is suitable for target systems with layered and staggered evaluation indicators [6, 22]. The Long Short-Term Memory (LSTM) network is a type of Recurrent Neural Network (RNN) that consists of input, output, and hidden layers. It is developed to solve the issue of long-term dependencies in RNNs and is used for predicting temporal data, translating text, and recognizing speech. As a result, LSTM is being researched for its potential in predicting Network Security Situations (NSS) and improving network convergence speed [1, 28]. The improved LSTM is optimized using the Look ahead method in combination with the Nadam algorithm. With the rapid development of network attack technology, network security issues have become increasingly severe. The existing methods of intrusion detection computation, firewalls, virus killing, etc. can only take preventive and protective measures against threat attacks that have already occurred, and cannot grasp the trend of network systems. Therefore, NSS awareness has emerged. At present, there are NSS awareness technologies that use network definition as the foundation, as well as those that use fuzzy reasoning as the foundation. Although they can perceive the NSS, there are significant errors and high computational complexity compared to the actual network state. Furthermore, existing network security technologies struggle to effectively respond to the constant network attacks. In this context, this study

utilizes the advantages of deep neural networks in processing high complexity network environments and non-linear data to study situation assessment and prediction. The study builds an NSS assessment model with improved AHP and an NSS prediction model with improved Nadam-LSTM. The research aims to improve the accuracy and effectiveness of NSS assessment and prediction by improving traditional AHP and LSTM models, providing new methods and technologies for research and practice in the field of network security. The main structure of the study is structured into four sections. The first section analyzes the current relevant research status. The second section builds an NSS assessment model based on improved AHP and an NSS prediction model based on improved Nadam-LSTM. The third section analyzes the application effect of the proposed model. The final section summarizes the entire study.

The novelty of this study mainly consists of two points. The first point is to address the subjectivity of AHP, which relies on expert experience. Linear programming is used to analyze the influencing factors in the system network and obtain a weight matrix, which helps to effectively handle complex data in the network system. The second point is to use LSTM to handle the time series problem of uncertain and nonlinear NSS prediction, and combine Look ahead method and Nadam algorithm to optimize LSTM. The online update mechanism is integrated into the parameter update of LSTM to improve the accuracy and convergence speed of the situation prediction model.

The contribution of the research lies in using AHP and Nadam algorithm to construct a campus NSS awareness model, and combining linear programming method to optimize and improve AHP. It achieves rapid perception of abnormal situations such as network attacks and hacker intrusions, provides guarantees for campus network security and has important application value for campus network security protection. The contribution of research to the knowledge system lies in the NSS through linear programming and improved Gravity Search Algorithm (GSA) optimized Fuzzy Neural Network (FNN), which enhances the objectivity and accuracy of the NSS assessment model. Secondly, by combining LSTM and Nadam algorithms for optimization, the online update mechanism is integrated into parameter updates, which improves the accuracy and convergence speed of the NSS prediction model. Furthermore, it provides new ideas and methods for research and practice in the field of network security, and adds new content and technical means to the knowledge system of NSS assessment and prediction.

2. Related Works

NSS awareness denotes to obtain security elements, and understand them to predict future trends in network security development. Chen [5] stated that NSS awareness is a core hot topic in network information security, which is receiving increasing attention. To explore the application effects of intelligent learning algorithms, a Radial Basis Function (RBF) prediction model on the basis of simulated annealing algorithm and hybrid hierarchical genetic algorithm optimization was constructed, and relevant experiments were conducted. The results indicated that the proposed model had good prediction performance. Tan *et al*. [21] proposed a HoneyNet method that includes threat detection and situational awareness to address the security threats and attacks on intelligent objects, gateways, and edge nodes in the Internet of Things (IoT), but traditional network security methods are not fully applicable. This approach enhanced the security and resilience of the IoT. The outcomes indicated that the proposed method had certain feasibility and effectiveness. Chen *et al*. [4] proposed a security awareness and protection system that utilizes a zero-trust architecture based on the 5G intelligent medical platform to construct a trustworthy dynamic access control model and achieve real-time NSS awareness. The results indicated that the system met the data and end-to-end security enforcement requirements involved in 5G-based intelligent medical systems. Liu and Zeng [14] used wavelet packet FNN and chaotic Particle Swarm Optimization (PSO) algorithm to monitor the NSS to ensure the security of the intelligent city's IoT and its normal operation. They analyzed the basic theory of the security situation of the intelligent city's IoT network and designed a framework for the perception of the security situation of the IoT. When conducting Multi-Criteria Decision Analysis (MCDA) for network security, it is important to consider special factors such as complexity, time sensitivity, uncertainty, and data integration. This requires decision analysis methods to make effective decisions in constantly changing network environments to protect network security. Sonal and Ghosh [19] studied a detailed multi-level framework to analyze the resilience performance of active distribution networks. A language scale-based MCDA method was used for situational awareness-based resilience level estimation of active distribution networks considering different high impact events. Bouramdane [3] proposed an MCDA method using AHP to address various network security threats and attacks faced by smart grids, and also studied the integration of artificial intelligence technology in smart grid security. Madhavi *et al*. [15] proposed using the MCDA model to evaluate the impact of each energy consuming sensor node on the network cooperation process in response to resource consumption attacks in wireless sensor networks. This helped to offset the impact of resource consumption attacks and improve the quality of service in the network.

The Nadam algorithm is an extension of the gradient descent optimization algorithm and can improve its performance. It is a deep learning optimization algorithm that has been extensively researched by many scholars.

Zhu and Hou [31] put forward an improved Nadam algorithm with the discussion and comparison of various optimizers in neural networks with Adam algorithm. The findings indicated that the proposed algorithm performed well in terms of loss and accuracy between output and actual results when classifying data using a three-layer neural network. Gui *et al*. [10] proposed a Nadam algorithm to address the problem of weak generalization ability and even inability to converge in extreme cases of the Adam algorithm, to achieve faster convergence speed and higher training accuracy. The outcomes expressed that the proposed algorithm had good training performance on deep learning tasks. Zhang *et al*. [30] put forward a bidirectional gate recurrent unit neural network with attention mechanism for the problem of charging state estimation in battery management. They also developed an adaptive momentum optimization algorithm combined with Nadam algorithm to update the weight matrix of the neural network. The findings denoted that the proposed algorithm could capture charging dynamics well and had a fast convergence speed, with a certain degree of effectiveness. Darvishi *et al*. [7] proposed a universal sensor validation architecture based on Nadam algorithm, which is built on a series of neural network estimators and classifiers, to detect anomalies in sensor measurements, identify faulty sensors, and provide appropriate estimation data. The outcomes indicated that the raised architecture had certain effectiveness in both hard and soft fault detection. Wozniak *et al*. [24] proposed an evaluation model with Nadam algorithm to analyze the network traffic of various IoT solutions, addressing the issue of pre-evaluation of information entering the network in network physical systems. The findings expressed that the proposed model had high accuracy in identifying potential even when the number of evaluated network features decreased. Iqbal *et al*. [12] expressed that deep learning is becoming increasingly popular in various research fields and widely used to solve image classification problems. To compare and find better learning algorithms in image classification tasks on small datasets, hyper-parameters related to optimizers and models were adjusted, and eight learning algorithms, including Nadam algorithm, were used for experiments.

Xie [26] proposed a network perception risk perception and prevention model based on deep reinforcement learning. By using the deep reinforcement learning algorithm to monitor and evaluate the network status in real time, it improved the ability to identify advanced attack methods and reduced the false positive rate. The experimental results showed that this model increased the anomaly detection rate by 98.3%, enhanced the accuracy of attack prediction by 97.4%, increased the accuracy of network risk assessment by 96.4%, and simultaneously reduced the false positive rate by 11.2%. Manaa *et al*. [16] adopted machine learning and deep learning methods, including random forest, Support Vector Machine (SVM), logistic regression, multi-layer perceptron, deep artificial neural network and LSTM network, etc., to detect and mitigate Distributed Denial of Service (DDoS) attacks. The results showed that the random forest exhibited a 100% accuracy rate and the lowest false positive rate on the UNSWNB 15 and UNSW2018 IoT Botnet datasets. However, on the actual edge Industrial Internet of Things (IIoT) datasets, the accuracy rate of the random forest was 98.79%, and the accuracy rate of the LSTM method in deep learning reached 99.36%. Moreover, it took the least time among multiple types of detection. Wu *et al*. [25] proposed a graph neural network based on the attention mechanism for detecting cross-level and cross-departmental network attacks. By constructing a graph structure based on log density to organize network traffic information and using the federated graph attention network model to evaluate the interactivity between graph nodes, the accuracy of internal network interaction was improved. The experimental results showed that this method achieved comparable accuracy and robustness to traditional detection methods while protecting privacy and data security. Dehkordi *et al*. [8] used a Transformer-based autoencoder model to conduct network intrusion detection on the Network Security Laboratory-Knowledge Discovery in Databases (NSL-KDD), and evaluated the anomaly detection performance of the model on the test set through key metrics such as Mean Squared Error (MSE), Area Under Curve (AUC), and Root Mean Squared Error (RMSE). The results showed that the model performed excellently in anomaly detection, with low MSE, high AUC and relatively low RMSE values, proving the potential of the model in the field of network security.

In summary, the current trend in NSS awareness research is towards data-driven methods, utilizing machine learning, deep learning, and other technologies to detect, predict, and respond to network security incidents. But with the increasing complexity and frequency of network attacks, the real-time requirements for NSS awareness are becoming higher and higher. Network security data typically has high-dimensional and high-frequency characteristics, and how to effectively process and utilize this data for situational awareness remains a challenge. Therefore, research established an NSS assessment model based on improved AHP and an NSS prediction model based on improved Nadam-LSTM, which can help improve the accuracy and efficiency of NSS awareness.

3. A Campus NSS Awareness Model Based on AHP and Nadam Algorithm

Campus network security is an important component of campus security. Network attacks pose a significant threat to students and teachers who rely on the internet for their daily activities. In response to the issue of campus NSS awareness, an NSS assessment model

based on improved AHP and an NSS prediction model based on improved Nadam-LSTM are studied and built to jointly carry out campus NSS awareness.

3.1. Building an NSS Assessment Model Based on Improved AHP

NSS assessment refers to detecting computer systems or network facilities, identifying security vulnerabilities, and taking effective measures as soon as possible to protect the security of network systems. The current NSS assessment methods mainly include AHP, fuzzy-AHP, Delphi method, and comprehensive analysis method. However, traditional analysis methods rely too heavily on expert experience. Additionally, data obtained through situational elements is complex and nonlinear, which can negatively impact the accuracy and reliability of evaluations, potentially increasing the time and cost of evaluation. Therefore, this study proposes an AHP NSS evaluation model based on linear programming, which uses linear programming to calculate the comparison matrix, effectively avoiding the subjectivity problem of traditional methods relying on expert experience to calculate indicator weights. Additionally, an improved GSA was introduced to improve convergence speed. Network node measurement involves multiple factors. To objectively evaluate the impact of attack nodes in complex networks, it is necessary to build a comprehensive evaluation indicators system. AHP decomposes various elements related to decision-making into three levels: objectives, criteria, and plans, and organizes quantitative and qualitative analysis to achieve the final decision. The specific steps are shown in Figure 1.

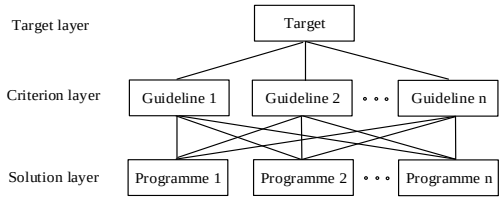


Figure 1. Specific steps of AHP.

In decision-making problems, ranking similarity measurement methods such as Ranking-wise Similarity and Weighted Similarity can be incorporated into MCDA. Ranking-wise Similarity is used to compare the similarity between two rankings, while Weighted Similarity is a measurement based on weights. The ranking similarity measurement method provides decision-makers with an effective tool to compare different decision options or evaluate the ranking of different attributes, facilitating the analysis of multi-criteria decision-making problems. When determining the weights of various levels and factors, to reduce the hardness of comparing many factors with different properties and raise accuracy, the consistency matrix method is used to compare the two factors with each other and rate them based on their importance. The matrix formed by the comparison results is the Judgment Matrix (JM), as denoted in Equation (1).

$$A = (a_{ij})_{n\times n} = \begin{pmatrix} a_{11}, a_{12}, \dots, a_{1n} \\ a_{21}, a_{22}, \dots, a_{2n} \\ \dots \\ a_{n1}, a_{n2}, \dots, a_{nn} \end{pmatrix} \quad (1)$$

In Equation (1), a_{ij} means the comparison outcome between the *i*-th and the *j*-th factors. The row vectors of the JM can obtain feature vectors are geometrically averaged, and then it normalizes each feature vector, as indicated in Equation (2).

$$\bar{W}_i = \frac{\bar{W}_i}{\sum_{i=1}^{n} \bar{W}_i}, (i = 1,2, \dots, n) \quad (2)$$

Then the feature vector is $W[W_1, W_2, \dots, W_N]^T$. The *max* eigenvalue calculation of the JM is shown in Equation (3).

$$\lambda_{max} = \frac{1}{n}\sum_{i=1}^{n} \frac{[AW]_i}{W_i}, (i = 1,2, \dots, n) \quad (3)$$

To avoid interference from other factors, consistency testing of the JM is required, as expressed in Equation (4).

$$\begin{cases} CR = \dfrac{CI}{RI} \\ CI = \dfrac{\lambda_{max} - n}{n - 1} \end{cases} \quad (4)$$

In Equation (4), *CR* denotes the Consistency Ratio (CR). When it is less than 0.1, the JM passes the one-time test. *RI* denotes the average random consistency indicator, and *CI* means the deviation from the consistency indicator. Traditional AHP requires expert experience to obtain a JM. Here, experts refer to those who have domain knowledge and experience in decision-making and can provide valuable opinions for decision-making. They can be subject experts, industry practitioners, managers, or individuals with relevant knowledge and experience. Therefore, the obtained JM has a certain degree of subjectivity. To address this issue, linear programming is used to analyze the influencing factors in the system network and improve the consistency of the JM. In response to the traditional single indicator decision system network model, the attribute characteristics of the system network are studied and analyzed. A hierarchical situation assessment is proposed, and a first level indicator combining threat, vulnerability, stability, and disaster tolerance of the network is established. The specific network situation indicator system is denoted in Table 1.

Table 1. Network situation indicator system.

Primary indicators	Secondary indicators
Threatening	Attack type, attack success probability, attack frequency, consequences of the attack.
Vulnerability	Network vulnerability level, number of vulnerabilities, network self-protection situation, storage media situation.
Stability	CPU occupancy rate, network traffic change rate, total data flow, bandwidth usage rate.
Disaster recovery	Host operating system, network bandwidth, device version, service type.

Linear programming is a quantitative analysis

technique used to address issues with linear objective functions and linear constraint objectives. It is commonly used for quantitative analysis and performs well in solving problems with linear constraint objectives, making it suitable for problems with measurable objectives. Linear programming was used to replace expert judgment in traditional AHP. The weight of AHP indicators was calculated through linear programming, and the decision matrix of AHP was obtained. The optimal weight decision comparison matrix was selected using the established NSS assessment index system and the quantitative values of the indicators. It mainly consists of four components: decision variables, variable boundaries, constraint conditions, and objective functions. The optimal comparison matrix of the first level indicator can be obtained by quantifying the influence factors of the second level indicator, and the optimal value of the decision variable can be solved as shown in Equation (5).

$$\begin{cases} S_{opt} = \sum_{i=1}^{k} \alpha_i S_i, M_{opt} = \sum_{i=1}^{k} \beta_i M_i \\ F_{opt} = \sum_{i=1}^{k} \gamma_i F_i, R_{opt} = \sum_{i=1}^{k} \delta_i R_i \end{cases} \tag{5}$$

In Equation (5), S_i, M_i, F_i, and R_i represent the quantitative values of the secondary indicators of stability, threat, vulnerability, and disaster tolerance, respectively, while α_i, β_i, and δ_i represent the constants of the decision variables assigned to each secondary indicator. Linear programming can be used to quantify the impact factors of secondary indicators and obtain the optimal comparison matrix of primary indicators. In AHP, the influencing factors of these secondary indicators are used as priorities for situational assessment. The improved AHP based on linear programming first constructs a comparison matrix using the results of the linear programming objective function and matrix permutation, which is transposed to obtain the comparison matrix as shown in Equation (6).

$$\begin{bmatrix} S_i \\ M_i \\ F_i \\ R_i \end{bmatrix}^T = \begin{bmatrix} S_1, M_1, F_1, R_1 \\ S_2, M_2, F_2, R_2 \\ S_3, M_3, F_3, R_3 \\ S_4, M_4, F_4, R_4 \end{bmatrix} \tag{6}$$

Secondly, based on the relative relationship between various influencing factors, a general matrix is constructed as shown in Equation (1). Then, the general matrix is normalized. Finally, the eigenvectors corresponding to the eigenvalues in the JM are used as the relative weights of the influencing factors to determine the weights of the optimal matrix, and consistency checks are performed on the results.

In summary, in the process of improving the AHP algorithm, an NSS index system is first constructed. The first-level indicators are divided into threat, vulnerability, stability, and disaster tolerance, and 4 to 5 second-level indicators are set under each first-level indicator (such as threats including attack types, success probabilities, etc.). Then, linear programming is adopted to replace the traditional expert scoring. Taking the quantified values of the secondary indicators as the input, an objective function is constructed to maximize the rationality of the indicator weights. The constraint conditions ensure that the sum of the weights is 1 and non-negative, and the optimal weight matrix is solved by the simplex method. Finally, a JM is generated based on the weight matrix to calculate the maximum eigenvalue, Consistency Index (CI) and CR. If the ratio is less than 0.1, the consistency test is passed; otherwise, the linear programming parameters are adjusted and the solution is re-solved.

FNN integrates the advantages of fuzzy theory and neural networks, and can handle high complexity and nonlinear data. However, it also has the issues of slow convergence speed and being prone to falling into local optima. Therefore, the study combines GSA to optimize FNN. GSA is a stochastic heuristic optimization algorithm that searches for the optimal solution by moving the particle positions of the population. It assumes that there are N individuals in a search space where individual particles are randomly placed, and the gravitational force of individual j on individual i at t time is shown in Equation (7).

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t)M_{aj}(t)}{R_{ij}(t)+\varepsilon}\left(x_j^d(t) - x_i^d(t)\right) \tag{7}$$

In Equation (7), $G(t)$ indicates the gravitational constant at the moment of t. M_{pi} and M_{aj} and $R_{ij}(t)$ express the masses and the Euclidean distance of individuals i and j, respectively. ε means the constant. x_i and x_j refer to the positions of individuals i and j, respectively. The goal of the fitness function is to minimize errors, and the best and worst fitness are shown in Equation (8).

$$\begin{cases} best(g) = \min\ fit_j(g),\ j \in 1,...,I \\ worst(g) = \max\ fit_j(g),\ j \in 1,...,I \end{cases} \tag{8}$$

In Equation (8), *max* $fit_j(g)$ and *min* $fit_j(g)$ represent the *max* and mini fitness values of the j-th particle at the g-th iteration, respectively. According to the law of motion, the acceleration of individual i during the iteration of t times is shown in Equation (9).

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{9}$$

In Equation (9), $M_i(t)$ indicates the mass of the individual i. The position and speed updates of individual i are shown in Equation (10).

$$\begin{cases} v_i^d(t+1) = rand_i v_i^d(t) + a_i^d(t) \\ x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \end{cases} \tag{10}$$

In Equation (10), $rand_i$ means the inertia weight, with a value of [0, 1]. To promote the convergence and

performance of GSA and avoid the algorithm falling into local optima, the study first uses the average acceleration of all particles in the same one-dimensional dimension to replace a single particle, improves the acceleration of a single particle in the velocity update formula, and then optimizes the inertia weight to enhance the algorithm's self-learning ability. The process of improved GSA is expressed in Figure 2.

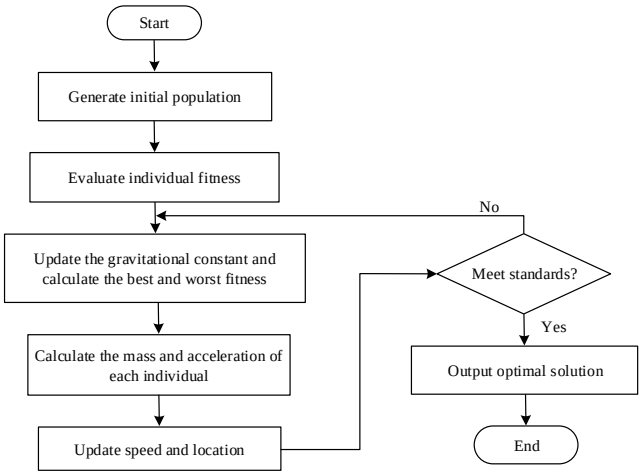


Figure 2. Flow chart of improved GSA.

Because $rand_i$ is a random constant, a self-adaptive inertia weight update formula is proposed. As the iteration times increases, the inertia weight decreases. To further enhance the global cognitive ability of each particle and improve their convergence ability, the new inertia weight update equation is shown in Equation (11).

$$rand_i^{'} = \left[rand_{\max} - \frac{t(rand_{\max} - rand_{\min})}{T} \right] \frac{1}{\sqrt{\ln 2}(1+e^t)} \quad (11)$$

In Equation (11), t serves as the current iteration times, and T represents the given *max* amount of iterations. The formula for updating the speed and position of individual i in the improved GSA is shown in Equation (12).

$$\begin{cases} v_i^d(t+1) = rand_i^{'} v_i^d(t) + \bar{a}_i^d(t) \\ x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \end{cases} \quad (12)$$

In the process of the GSA-FNN algorithm, the structure of the FNN is initialized first, and the positions of the particle swarm (corresponding to the network weights) are randomly generated. Particle fitness is computed (based on output error minimization), global optimal positions and individual optimal positions are updated, and particle positions are updated iteratively by a modified GSA, where inertia weights are adaptively decayed with the number of iterations. When the fitness converges or reaches the maximum number of iterations, the position of the optimal particle is assigned to the FNN weight to complete the model training.

In summary, the NSS estimation model with improved AHP is shown in Figure 3. The study first uses linear programming to optimize AHP, establishes an AHP NSS evaluation model based on linear programming, and obtains safety situation evaluation indicators and their weights [20]. Then, the study combines GSA to improve the FNN and conduct NSS assessment.

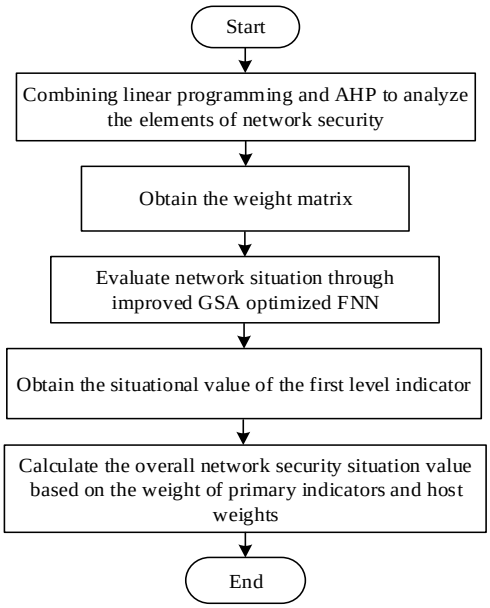


Figure 3. NSS estimation model based on improved AHP.

3.2. Building an NSS Prediction Model with Improved NADAM-LSTM

To accurately predict the growth trend of network security from various aspects, it is important to quickly and accurately predict the NSS. In the field of situation awareness, situation prediction is located in the middle layer. Firstly, the model is trained by extracting conventional network feature parameters, and then the NSS prediction results are obtained. The efficiency of the network is compared through error analysis. In NSS awareness, intrusion prediction methods contain two categories: prediction models and intrusion detection enhancement. Prediction models mainly consist of qualitative prediction methods, time series, and causal prediction methods. Classic prediction models include SVM-based prediction models and gray theory-based prediction models. The prediction of NSS is a time series problem that requires consideration of the impact of time series on network security development trends. However, due to the uncertainty and non-linearity of time series, the accuracy of traditional situation prediction methods is often affected [29]. To address this issue, this study utilizes LSTM, which is suitable for processing long sequence data, for modeling. LSTM is a temporal RNN with four interaction layers. The problem of gradient back-propagation during the training process of RNNs can be solved by introducing memory units. To prevent gradient vanishing and explosion, LSTM's recurrent units include forgetting gates, input gates, and output gates. The structure of LSTM's recurrent units is expressed in Figure 4.

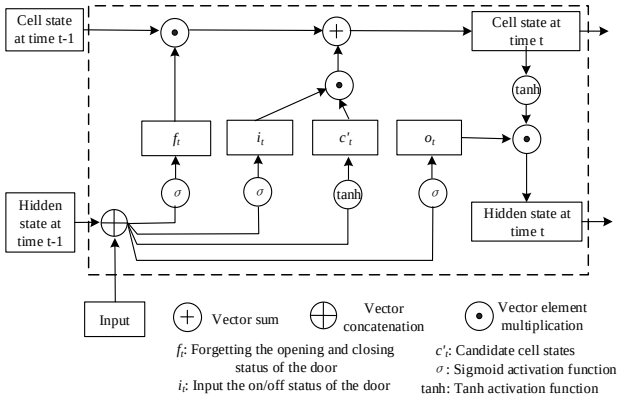


Figure 4. The cyclic unit structure of LSTM.

To optimize network parameters using real-time received network situation values, an LSTM learning online update mechanism using a minimum cost function is studied to establish an NSS prediction model. With the practical problems of network systems, it is proposed to update network parameters using the situation time series data transmitted online, to establish a more effective LSTM situation prediction model under the condition of real-time observation of network situation data. Firstly, an improved LSTM prediction model is established by using existing historical data. Then, the established prediction model is utilized to obtain the predicted value, and the new observation data is utilized as the true value of the previous sampling time. Finally, the model parameters are iteratively updated. By proposing methods for improving the LSTM model to update parameters in real-time, as online data is increasingly used, the predicted values obtained by the model can become more accurate, which is beneficial for administrators to monitor network security online. The update for improving LSTM is shown in Equation (13).

$$\begin{cases} f_t = \sigma\left[W_f(h_{t-1}, x_t) + b_f\right] \\ i_t = \sigma\left[W_i(h_{t-1}, x_t) + b_i\right] \\ \overline{c}_t = \tanh\left[W_c(h_{t-1}, x_t) + b_c\right] \\ c_t = f_t * c_{t-1} + i_t * \overline{c}_t \\ o_t = \sigma\left[W_o(h_{t-1}, x_t) + b_o\right] \\ h_t = o_t * \tanh(c_t) \end{cases} \tag{13}$$

In Equation (13), σ represents the sigmoid activation function, h_t-1 represents the input at the previous time, x_t represents the input at the current time, W and b represent learnable parameters, i_t represents the state of the input gate, $\bar{c}_t$ represents candidate cell cells, *tanh* represents the activation function, c_t represents the cell state at the current time, and c_{t-1} represents the cell state at the previous time. To raise the convergence speed of the improved LSTM, the Look ahead method combined with the Nadam algorithm is studied to optimize the improved LSTM. The improved LSTM can train the weights of four matrices and connect different layers as inputs to the Nadam algorithm [27]. The current situation prediction based on situational awareness is the development trend of NSS awareness. NSS prediction is a proactive defense mechanism. It analyzes and understands current and past network elements, and predicts future network trends. NSS awareness is the basis for current prediction, which is a critical component of cyberspace situational awareness. Figure 5 shows the specific process of the NSS awareness model.

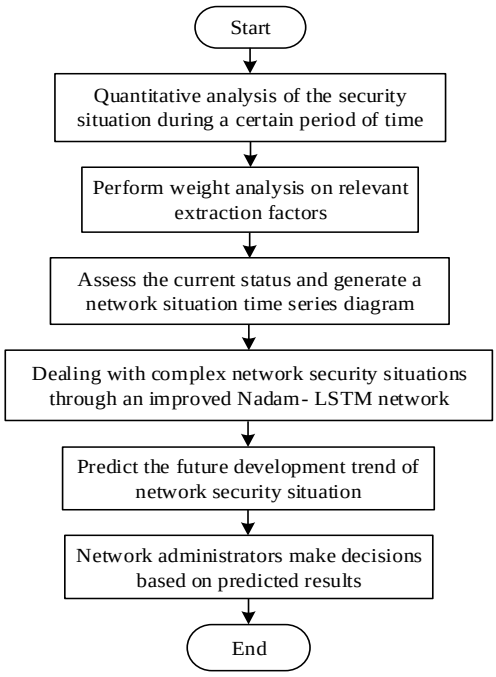


Figure 5. NSS awareness model.

Nadam algorithm is an efficient stochastic optimization method with adaptive learning rate, which has been widely used in deep learning algorithm optimization [13]. The Nadam algorithm is an optimization algorithm of the Adam algorithm, which combines NGA and Adam to expand the actual time series into a matrix. The standardized time series E is shown in Equation (14).

$$E = \frac{e_i}{\sqrt{e_i^2 + e_{i+1}^2 + ... + e_{i-k+1}^2}}, i = 1, 2, ..., x - k + 1 \tag{14}$$

In Equation (14), x severs as the length of the time series, and k denotes the number of samples. It initializes network parameters while setting parameters, as shown in Equation (15).

$$\begin{cases} U_f = rand(H, G) \\ q_f = rand(1, G) \\ ... \\ Max\_iter = P_1 \\ Error\_Cost = P_2 \end{cases} \tag{15}$$

In Equation (15), U_f refers to the weight of the forgetting gate. q_f expresses the forgetting gate bias. H denotes the number of LSTM cell units. G means the number of neuron layers. P_1 severs as the *max* amount of iterations. P_2 indicates the error threshold. In the forward propagation of LSTM, the cell unit state information $\hat{r}_t$

that needs to be forgotten, the amount of information $\hat{i}_t$ that can be saved in the cell unit state at t time, the cell unit state V_t, and the predicted value s_t at t time are calculated as shown in Equation (16).

$$\begin{cases} \hat{r}_t = \omega\left(U_f[s_{t-1}, e] + q_f\right)V_{t-1} \\ \hat{i}_t = \omega\left(U_i[s_{t-1}, e_t] + q_i\right)\tan s\left(U_V[s_{t-1}, e_t] + q_V\right) \\ V_t = \hat{r}_t + \hat{i}_t \\ s_t = \omega\left(U_\alpha[s_{t-1}, e_t] + q_\alpha\right)\tan s(V_t) \end{cases} \quad (16)$$

In Equation (16), $\omega(U_f[s_{t-1}, e]+q_f)$ severs as the forgetting gate's output. V_{t-1} means the previous moment's unit state. $\omega(U_i[s_{t-1}, e_t]+q_i)$ denotes the inputting gate's output, which determines the value that the cell unit needs to be updated. $tan\ s(U_V[s_{t-1}, e_t]+q_V)$ means a brand-new candidate vector constructed using the function $tan\ s$. $\omega(U_\alpha[s_{t-1}, e_t]+q_\alpha)$ indicates the outputting gate's output, and tan $s(V_t)$ represents the unit state at the current time. It repeatedly calculates Equation (16) until all training samples' predicted values are obtained, and then the total error between all predicted and true values is obtained, as shown in Equation (17).

$$J_\theta(f, s, U, q) = \frac{1}{2}\|f - s\|^2 \quad (17)$$

If the total error does not arrive the error threshold or the *max* amount of iterations, the iteration times are increased by 1 and the network parameters are updated using the back propagation through time algorithm [2]. The Look ahead method combined with the Nadam algorithm is used to train LSTM, and it inputs the weight matrix $\zeta_0=[U_r, U_i, U_v, U_\alpha]$ to be updated. It updates parameters in real-time based on observation data, and adds ζ_0 and $E_{n+1}(e_{n-k+2}, \ldots, e_{n+1})$ for LSTM forward propagation. The predicted values of the new samples are shown in Equation (18).

$$error = error + \frac{1}{2}(s_{n+1} - e_{n+2})^2 \quad (18)$$

At the next sampling time, if the predicted value arrives the point of network attack, the administrator will issue a attack alarm to avoid further attacks on the network. In summary, the specific process of the NSS prediction model with improved Nadam-LSTM designed in the research is indicated in Figure 6.

In the process of the Nadam-LSTM algorithm, the time series of the NSS is standardized first, and the training set, verification set and test set are proportionally divided. Then, the LSTM network is constructed. The Nadam optimizer is used to update the parameters, the Nesterov momentum term is integrated to adjust the gradient direction, and the Look ahead method is introduced to globally correct the parameters every 5 iterations. Forward propagation calculates the states of the forget gate, input gate and output gate, updates the cell state, and optimizes the network weights through the backpropagation time algorithm until the mean square error converges.

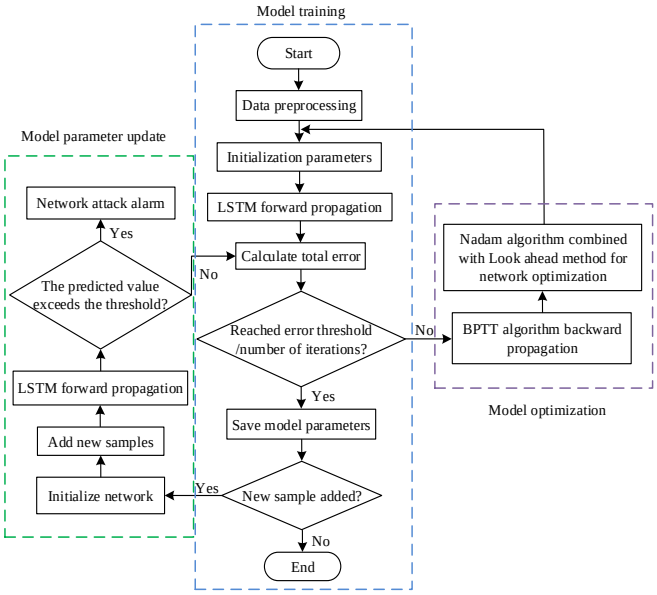


Figure 6. Flow chart of network situation prediction model with improved Nadam-LSTM.

4. Analysis of the Effectiveness of Campus NSS Awareness Model with AHP and Nadam Algorithm

The study proposed an NSS assessment and a prediction model with improved AHP and Nadam-LSTM, respectively. These models are beneficial for maintaining campus network security, but their effectiveness requires further verification. The research mainly analyzed from two aspects. The first part analyzed the performance of the improved AHP-based NSS assessment model, and the second part analyzed the effectiveness of the improved Nadam-LSTM-based NSS prediction model.

4.1. Effectiveness Analysis of NSS Assessment Model with Improved AHP

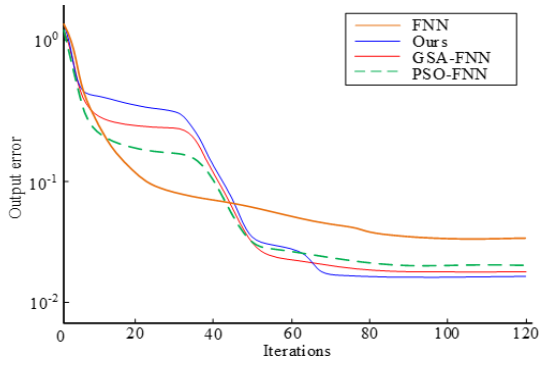
In terms of dataset processing, the DARPA2000 dataset removed redundant data packets through traffic cleaning, processed numerical features using Z-Score standardization, and was divided into the training set, validation set, and test set in a ratio of 7:1.5:1.5. To address the problem of unbalanced attack categories, the SMOTE oversampling technique was adopted to balance the sample distribution. After the HoneyNet dataset was parsed by Snort alert data, the time series features were extracted through the sliding window, the dataset was segmented at 8:1:1, and the rare attack categories were processed using category-weighted cross-entropy. The campus network dataset collected 90-day traffic logs. Invalid logs were filtered through regular expressions, and Min-Max standardization was adopted. The 75-day data was used as the training set and the 15-day data as the test set. The simulation scenarios included DoS attacks (200 times per hour), port scans (500 times per day), and internal threats (three privilege escalation

times per week). The attack frequency was set based on the threat statistics of the real campus network.

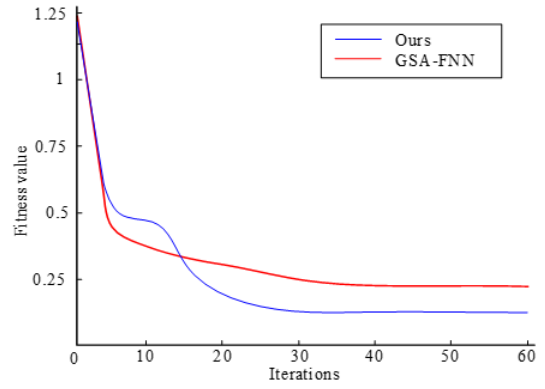
Table 2. Comparison results of consistency test analysis between two methods.

Algorithm	AHP	Improved AHP
λmax	5.09	4.89
CR	0.0028	0.0017
CI	0.004	0.001

To verify the effectiveness of AHP based on linear programming optimization, the CR of improved AHP was obtained through simulation experiments and compared with traditional AHP. The two methods' comparison outcomes are denoted in Table 2. From the table, the CR of improved AHP was less than 0.1, and the JM's inconsistency was within a reasonable range, indicating good consistency. In addition, the CR and CI of the improved AHP were both smaller than those of the traditional AHP, with a CR of 0.0017 and a CI of 0.001, indicating that the improved AHP methods proposed in the study had better matrix consistency and were effective.



a) Output error result.



b) Fitness value result.

Figure 7. Comparison results of output error and fitness value.

Simulation experiments were conducted to demonstrate the effectiveness of the proposed GSA-FNN evaluation method. The input and output sample dimensions were set to 5 and 1, respectively, with 8 hidden nodes, a maximum iteration of 180, and a learning rate of 0.35. The output errors were compared with those of traditional FNN, GSA-FNN, and PSO-FNN models. To further demonstrate the superiority of the improved GSA, fitness values were compared with the GSA-FNN model. The comparison findings are shown in Figure 7. From Figure 7-a), the proposed method, GSA-FNN and PSO-FNN models showed good performance in early iterations, avoiding FNN falling into local optima before the number of iterations was 30. In addition, the proposed method decreased again at 60 iterations and entered full optimization. It began to converge at around 69 iterations, resulting in the smallest output error of 0.0107 when compared to the other three models. From Figure 7-b), the fitness value of the GSA-FNN model rapidly decreased before 5 iterations, then tended to flatten out, and finally stabilized at around 0.24. The fitness value of the method proposed in this study gradually stabilized after 20 iterations, and finally stabilized at around 0.13, which was lower than the fitness value of the GSA-FNN model, indicating that the improved GSA had certain effectiveness.

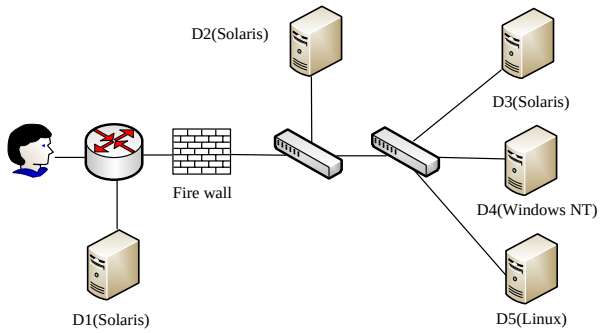
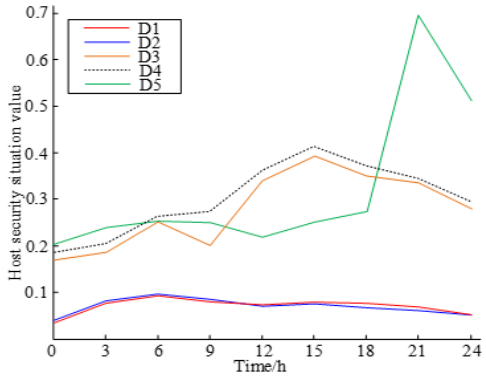


Figure 8. Set network scenarios.

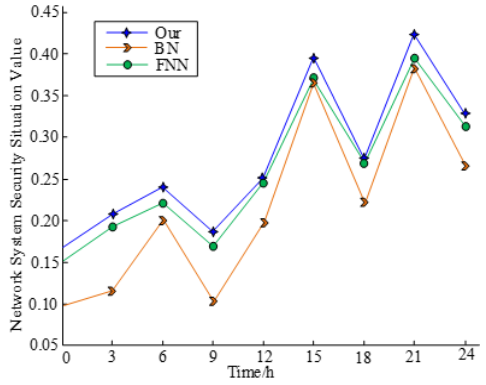
To verify the effectiveness of the improved AHP-based NSS assessment model, the study first used the improved AHP to obtain the weights of four first level indicators. The weights of stability, threat, vulnerability, and disaster tolerance were obtained as 0.18, 0.44, 0.20, and 0.18, respectively. Then, using the DARPA2000 dataset, the dataset was placed in a set network scenario for 24 hours, and threat attack quantification data was used as experimental data. The network was subjected to five major types of attacks every hour, and the changes in situation values were recorded. The network situation values were then calculated every three hours. The set network scenario is shown in Figure 8.

The host security situation values were obtained using the improved AHP-based NSS assessment model. Finally, the NSS was calculated by combining the host weight values and situation values, and compared with the Bayesian Network (BN) method and traditional FNN method. The results are shown in Figure 9. From Figure 9-a), hosts D1 and D2 were not threatened by network attacks, and the situation values did not change much. Therefore, it can be inferred that the host situation values were mainly caused by external issues. Host D5 suddenly received a threat attack during the 18–21-hour period, and the security situation value rapidly increased to 0.7, indicating a more dangerous network state. From Figure 9-b), compared to the other two methods, the network system security situation value obtained by this research method was higher, with the highest value of

0.425 at 21h. The results demonstrated the feasibility and performance of the NSS assessment model based on improved AHP.



a) Host security situation value.



b) Network system security situation value.

Figure 9. Comparison results of host security situation value and NSS.

To further validate the effectiveness and superiority of the proposed model, the study used a hacker attack Honeynet dataset collected by the HoneyNet organization for testing. This dataset contains Snort alarm data from 2000 to 2011 [23]. The network situational values obtained from this study model were compared with four methods: Stable Preference Ordering Towards Ideal Solution (SPOTIS), COrollation-based ModEl Tracking (COMET), Sequential Interactive Model for Urban Systems (SIMUS), and Risk Assessment Network COMponent (RANCOM) [9, 17, 18]. The SPOTIS model is a widely used model for NSS awareness. It provides comprehensive NSS awareness through real-time monitoring and analysis of network data traffic. The model was employed in threat detection and prevention, NSS, and other related fields. The COMET model is an NSS awareness method based on event correlation and model tracking, which can automatically detect potential threat behaviors, generate alerts, and respond. It has been widely applied in the field of network security. This study used COMET-II for comparison. The SIMUS model is based on simulation technology, which helps analyze, predict, and respond to network security threats by simulating the occurrence and evolution of network security events. It is widely used in NSS awareness. The RANCOM model is an NSS awareness method based on risk assessment, which evaluates the security risks of network systems and data, and provides security recommendations and measures. It is widely used in network security management by enterprises and government departments to discover potential security vulnerabilities and threats and improve their network security defense level. The results are shown in Figure 10. From Figure 10, among the four methods, the NSS value obtained by the research model was the highest overall, and the NSS was the best. The average network system security status value of this research model was 0.287, which was higher than SPOTIS's 0.281, COMET's 0.277, SIMUS's 0.269, and RANCOM's 0.272. The results indicates that the NSS assessment model based on improved AHP has good effectiveness and certain superiority.

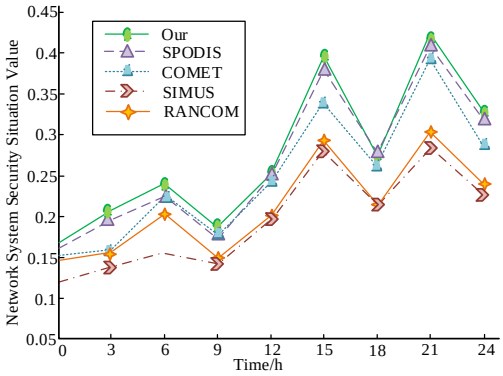


Figure 10. Comparison of network situation values for four methods.

In summary, the AHP based on linear programming optimization solved the subjective problem that required expert experience. Compared to traditional AHP, the JM's consistency was better, and the weights of the first level indicators were more objective and reasonable, which had a certain degree of effectiveness. With the improved GSA algorithm, the FNN evaluation model had a fast convergence speed and was not easily trapped in local optima. The NSS assessment model with improved AHP could also obtain high network system security situation values, which had certain feasibility and effectiveness.

4.2. Analysis of the Effectiveness of NSS Prediction Model with Improved NADAM-LSTM

To verify the effectiveness of the NSS prediction model based on improved Nadam-LSTM, the study used the system network attacks' historical log information collected by a certain network company for 90 days as raw data. The training set consisted of the first 75 days, while the test set consisted of the last 15 days. The model was configured with 28 input and 128 hidden layers, a batch size of 128, and 27 training steps. The NSS time series after standardizing the original data is shown in Figure 11.

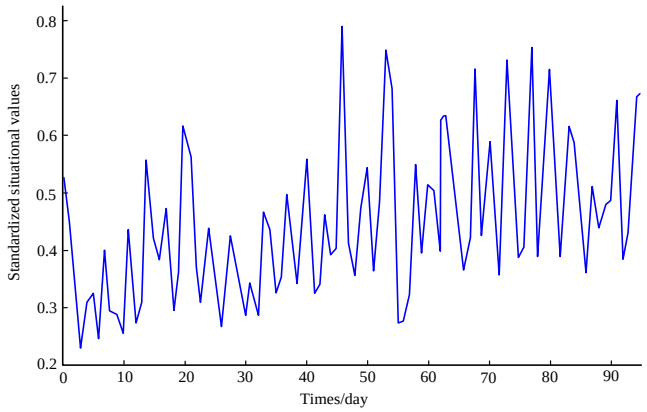


Figure 11. Standardized NSS time series.

To evidence the convergence of the Look ahead method combined with the Nadam algorithm, the MSE curve with respect to the number of iterations was obtained and compared with the Nadam, Adam, and RMSProp algorithms. The comparison results are shown in Figure 12. As the iteration times increased, the MSE of all four algorithms gradually decreased. In addition, among the four algorithms, the MSE of the raised algorithm was always smaller than that of the other three algorithms. At 10 iterations, the MSE was 0.41, which gradually decreased and tended to flatten out after 80 iterations. The results indicates that the combination of the Look ahead method and the Nadam algorithm has a certain effect on increasing the convergence speed of LSTM.

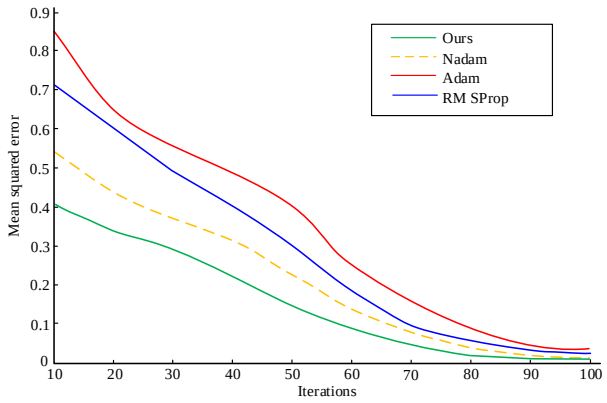


Figure 12. Curve of mean square error with iteration times.

To assess the performance of the NSS prediction model based on improved Nadam-LSTM, the accuracy, false positive rate and false negative rate of detecting network definition were compared with the fuzzy inference perception technology and the network definition perception technology. The outcomes are shown in Figure 13. Compared to the other two perception technologies, the NSS prediction model based on improved Nadam-LSTM had the highest accuracy, consistently maintaining over 90% and reaching a maximum of 98%. The study found that the accuracy of fuzzy inference perception technology was 84.61%. The false positive rate and false negative rate of the research model were both the lowest, with values of 2.64% and 11.03%, respectively, when the data size was 4000. The results indicates that the model constructed in the study has high accuracy in network definition, and had some feasibility and effectiveness.

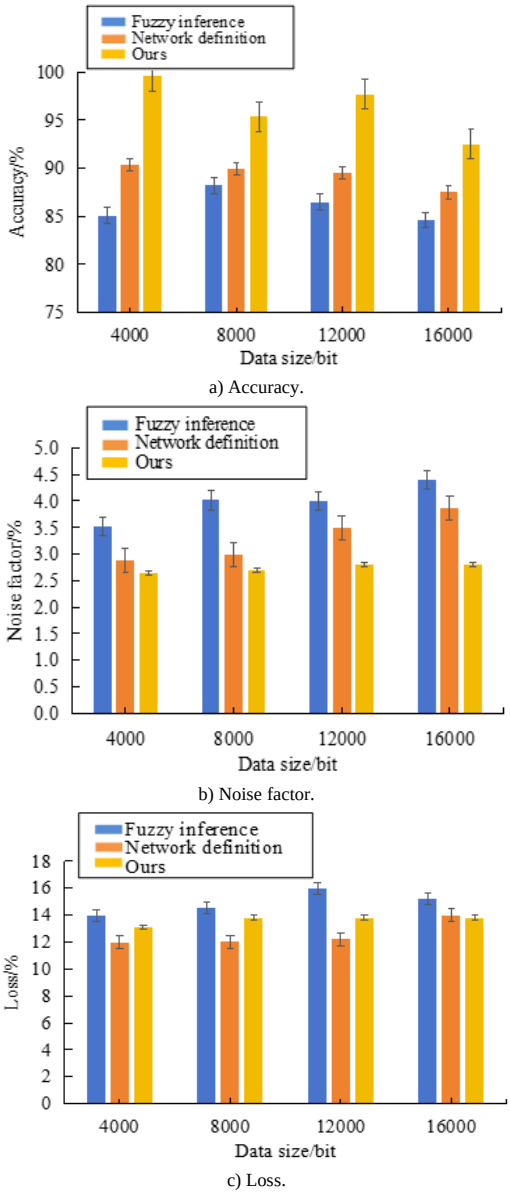
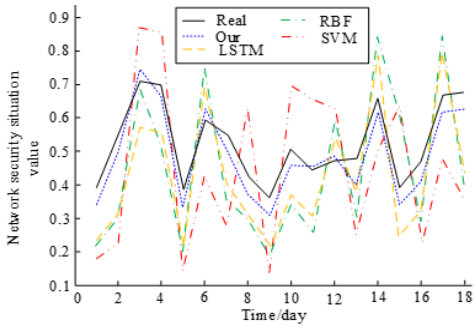


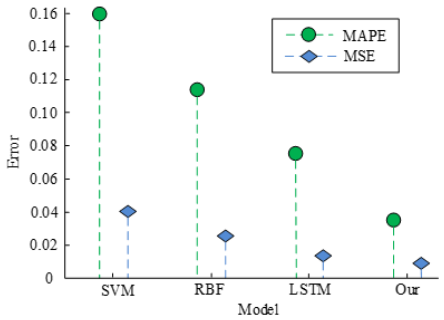
Figure 13. Comparison of three methods for network behavior detection.

To demonstrate the superiority of the NSS prediction model based on improved Nadam-LSTM, the study selected SVM, RBF, and traditional LSTM as controls, and compared the prediction accuracy of the four methods under the same conditions using NSS values. Mean Absolute Percentage Error (MAPE), and MSE were used as evaluation indicators. The comparison results are shown in Figure 14. Based on the graph, the model constructed in this study predicted the NSS value closest to the true value among the four prediction methods. It had a relatively small prediction error, with the lowest MAPE and MSE values of 0.039 and 0.01, respectively. The results indicates that the NSS prediction model with improved Nadam-LSTM can effectively utilize online data to improve prediction

accuracy, with high prediction accuracy and certain feasibility and effectiveness.



a) Network security situation value.



a) Comparison between MAPE and MSE.

Figure 14. Comparison of prediction accuracy results of four methods.

All the experimental results were based on five independent repeated tests and the two-tailed t-test (α=0.05) was used. In the comparison of NSS prediction models, the accuracy rate of the improved Nadam-LSTM (98.0%±1.2%) was significantly higher than that of the traditional LSTM (92.5%±2.3%), t (8)=5.62, p<0.001. MAPE (0.039±0.005) was significantly lower than SVM (0.072±0.008), t (8)=6.31, p<0.001. The confidence interval was calculated by the Bootstrap method. The average safety value of the model under the HoneyNet dataset was 0.287 (95% CI: 0.279-0.295), which was superior to the upper limit of the confidence interval of the comparison method.

To explore the practical application effect of an NSS prediction model based on improved Nadam-LSTM, a study was conducted on the campus network of a certain university, randomly launching Denial of Service Attacks (DoSA), internal threats, and key cracking attacks on the campus network. The prediction accuracy of the proposed model was compared with the currently advanced Bidirectional-LSTM (Bi-LSTM), the Extreme Learning Machine combined with fruit Fly Optimization Algorithm (FOA-ELM), and the LSTM-Informer model. The results are shown in Table 3. From Table 3, compared with the other three models, the NSS prediction model based on the improved Nadam-LSTM had the highest prediction accuracy under different types of network attacks, with 96.42%, 93.64%, and 95.81%, respectively. The results showed that the NSS prediction model based on the improved Nadam-LSTM had also demonstrated good application effects in real campus networks, and had certain feasibility and superiority.

Table 3. Comparison of prediction accuracy among four models.

Model	DoSA	Insider threat	Key cracking attack
Bi-LSTM	90.84%	88.98%	90.32%
FOA-ELM	93.64%	91.86%	92.76%
LSTM-Informer	92.84%	90.59%	92.17%
Improved Nadam-LSTM	96.42%	93.64%	95.81%

In summary, the MSE of the Look ahead method combined with the Nadam algorithm is always smaller than that of the Nadam, Adam, and RMSProp algorithms, indicating that the proposed algorithm has effect on increasing the convergence speed of LSTM. The NSS prediction model based on improved Nadam-LSTM has high accuracy, low false positive rate and false negative rate, and the predicted NSS value is close to the true value. Therefore, it has high accuracy in detecting network behavior and has some feasibility and effectiveness.

To verify the contributions of each component, ablation experiments were designed and studied, and the results are shown in Table 4. The results showed that linear programming increased the accuracy of AHP by 2.7% and reduced MSE by 0.007, verifying that it reduces subjective deviation through quantifying weights. The Look ahead mechanism accelerated the convergence of Nadam-LSTM by 15 steps and improved the accuracy rate by 3.3%, proving that it avoids local optimum through prospective update.

Table 4. Ablation experiment results.

Model variant	Accuracy rate (%)	MSE	Number of convergence iterations
Improve AHP (linear programming)	85.3±2.1	0.032±0.004	120
Nadam-LSTM (no lookahead)	94.7±1.8	0.015±0.003	95
Complete model	98.0±1.2	0.010±0.002	80

The comparison of model efficiency is shown in Table 5. In Table 5, the lightweight model was achieved through feature selection (retaining the top 20% of important features) and channel pruning. The results showed that the computational cost of the complete model increased by 62%, mainly due to the matrix inversion of linear programming and the momentum calculation of Nadam. However, the computational cost could be reduced by 40% through model compression, which was still better than the 23% improvement in accuracy of the baseline model.

Table 5. Model efficiency comparison.

Model	Training time (90-day data)	Parameter quantity (in millions)	Memory usage (GB)
Traditional AHP+LSTM	4.2h	1.2	2.8
Improved model (complete)	6.8h	2.7	5.3
Lightweight improvement model	5.1h	1.8	3.9

The applicability of the proposed method can be

expanded from three aspects. First of all, it is the flexibility of the indicator system. The linear programming framework allows for the dynamic adjustment of index weights based on the threat characteristics of different campus networks (for example, university research networks pay more attention to data leakage, while vocational education colleges focus on ransomware attacks), such as incorporating "the encryption strength of research data" into the disaster recovery index. The second is the scalability of the model structure. The time series modeling capability of Nadam-LSTM does not rely on a specific network topology. When the network expands from a star structure to a Mesh structure, only the input feature dimensions (such as increasing parameters like link load and routing hop count) need to be adjusted, and the time series features can be re-extracted through the sliding window for adaptation. The last point is the potential for cross-scenario migration. The weight quantification method of improving AHP can be extended to scenarios such as enterprise networks and government networks. Only the secondary indicators need to be replaced (such as adding "VPN access security" in enterprise networks), and the fuzzy reasoning mechanism of GSA-FNN is also effective for the real-time situation assessment of industrial control networks because the timing characteristics of industrial protocols have similar nonlinear characteristics to the traffic of campus networks.

5. Conclusions

As the advancement of technology, the Internet has gradually been applied to all aspects of life, bringing convenience to users while also posing network security risks. In response to the issue of campus NSS awareness, an NSS assessment model with improved AHP and an NSS prediction model with improved Nadam-LSTM were studied and built. The outcomes denoted that the CR and CI of the improved AHP were smaller than those of the traditional AHP, with a CR of 0.0017 and a CI of 0.001, indicating a certain degree of effectiveness. The FNN evaluation method optimized based on the improved GSA started to converge at around 69 iterations, with a small output error of 0.0107. The fitness value gradually stabilized after 20 iterations, and finally stabilized at around 0.13. The NSS evaluation model based on improved AHP obtained a higher value of 0.425 at 21h. The MSE of the Look ahead method combined with the Nadam algorithm was 0.41 at 10 iterations, and tended to flatten out after 80 iterations. The NSS prediction model based on improved Nadam-LSTM had the highest accuracy, consistently maintaining over 90%, with a maximum of 98%. The false positive rate and false negative rate were the lowest, with 2.64% and 11.03%, respectively. The predicted NSS value was closest to the true value, with relatively small prediction errors, with MAPE and MSE values of 0.039 and 0.01, respectively. In summary, the model constructed by the research has certain feasibility and effectiveness. However, there are three limitations in the current research. The first is that the computational complexity is relatively high. The improved linear programming solution of AHP and the bidirectional parameter update of Nadam-LSTM have led to a 62% increase in the model training time compared with the traditional methods, and are limited by Central Processing Unit (CPU) computing power when deploying edge computing devices (such as campus network access layer routers). The second is the insufficient generalization ability of new types of threats. The model relies on historical attack patterns for training, and the false negative rate for zero-day attacks (such as exploitation of unknown vulnerabilities) reaches 18.7%, because the fuzzy rule base of GSA-FNN does not contain the characteristics of unknown attacks. The third is the strong dependence on data collection. In scenarios where the proportion of network traffic encryption exceeds 60% (such as teaching platforms enabled with HTTPS), the accuracy of feature extraction decreases by 12-15%, resulting in deviations in the situation assessment values. Future research will focus on developing a lightweight model architecture to reduce the parameters of Nadam-LSTM through channel pruning and adapt to edge nodes. Then, the meta-learning mechanism is introduced to construct a fespars-shot zero-day attack detection module, and new types of threats are inferred by using the meta-characteristics of historical attacks (such as traffic mutation patterns). Meanwhile, a decoupling algorithm for encrypted traffic features is designed. Combined with the generative adversarial network, potential attack features are restored from encrypted traffic to improve its applicability in complex network environments.

Author Contributions

Liwen Xu made all the contributions.

Data Availability Statement

All data generated or analyzed during this study are included in this article. Further enquiries can be directed to the corresponding author.

References

[1] Aljadani E., Assiri F., and Alshutayri A., "Detecting Spam Reviews in Arabic by Deep Learning," *The International Arab Journal of Information Technology*, vol. 21, no. 3, pp. 495-505, 2024. https://doi.org/10.34028/iajit/21/3/12

[2] Alosaimi A. and Elloumi M., "Back Propogation Neural Network Based Cybersecurity Information Retrival from Repository," *Turcomat*, vol. 12, no. 10, pp. 1197-1204, 2021. https://turcomat.org/index.php/turkbilmat/article/

view/4312/3679

[3] Bouramdane A., "Cyberattacks in Smart Grids: Challenges and Solving the Multi-Criteria Decision-Making for Cybersecurity Options, Including Ones that Incorporate Artificial Intelligence, Using an Analytical Hierarchy Process," *Journal of Cybersecurity and Privacy*, vol. 3, no. 4, pp. 662-705, 2023. DOI: 10.3390/jcp3040031

[4] Chen B., Qiao S., Zhao J., Liu D., and et al., "A Security Awareness and Protection System for 5G Smart Healthcare Based on Zero-Trust Architecture," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10248-10263, 2021. DOI: 10.1109/JIOT.2020.3041042

[5] Chen Z., "Research on Internet Security Situation Awareness Prediction Technology Based on Improved RBF Neural Network Algorithm," *Journal of Computational and Cognitive Engineering*, vol. 1, no. 3, pp. 103-108, 2022. DOI: 10.47852/bonviewJCCE149145205514

[6] Dar T., Rai N., and Bhat A., "Delineation of Potential Groundwater Recharge Zones Using Analytical Hierarchy Process (AHP)," *Geology, Ecology, and Landscapes*, vol. 5, no. 4, pp. 292-307, 2020. https://doi.org/10.1080/24749508.2020.1726562

[7] Darvishi H., Ciuonzo D., Eide E., and Rossi P., "Sensor-Fault Detection, Isolation and Accommodation for Digital Twins via Modular Data-Driven Architecture," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827-4838, 2021. DOI: 10.1109/JSEN.2020.3029459

[8] Dehkordi S., Nasri S., and Dami S., "Unveiling Anomalies: Transformative Insights from Transformer-based Autoencoder Models," *International Journal of Computers and Applications*, vol. 47, no. 1, pp. 29-44, 2025. DOI: 10.1080/1206212X.2024.2441147

[9] Dezert J., Tchamova A., Han D., and Tacnet J., "The SPOTIS Rank Reversal Free Method for Multi-Criteria Decision-Making Support," *in Proceedings of the IEEE 23rd International Conference on Information Fusion*, Rustenburg, pp. 1-8, 2020. DOI: 10.23919/FUSION45008.2020.9190347

[10] Gui Y., Li D., and Fang R., "A Fast Adaptive Algorithm for Training Deep Neural Networks," *Applied Intelligence*, vol. 53, no. 4, pp. 4099-4108, 2023. DOI: 10.1007/s10489-022-03629-7

[11] Guo H., Li J., Liu J., Tian N., and Kato N., "A Survey on Space-Air-Ground-Sea Integrated Network Security in 6G," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 1, pp. 53-87, 2022. https://doi.org/10.1109/COMST.2021.3131332

[12] Iqbal I., Odesanmi G., Wang J., and Liu L., "Comparative Investigation of Learning Algorithms for Image Classification with Small Dataset," *Applied Artificial Intelligence*, vol. 35, no. 10, pp. 697-716, 2021. DOI: 10.1080/08839514.2021.1922841

[13] Ishtaiwi A., Ali A., Al-Qerem A., Alsmadi Y., and et al., "Impact of Data-Augmentation on Brain Tumor Detection Using Different YOLO Versions Models," *The International Arab Journal of Information Technology*, vol. 21, no. 3, pp. 466-482, 2024. https://doi.org/10.34028/iajit/21/3/10

[14] Liu Q. and Zeng M., "Network Security Situation Detection of Internet of Things for Smart City Based on Fuzzy Neural Network," *International Journal of Reasoning-based Intelligent Systems*, vol. 12, no. 3, pp. 222-227, 2020. https://doi.org/10.1504/IJRIS.2020.109650

[15] Madhavi S., Santhosh N., Rajkumar R., and Praveen R., "Pythagorean Fuzzy Sets-based VIKOR and TOPSIS-based Multi-Criteria Decision-Making Model for Mitigating Resource Deletion Attacks in WSNs," *Journal of Intelligent and Fuzzy Systems: Applications in Engineering and Technology*, vol. 44, no. 6, pp. 9441-9459, 2023. DOI: 10.3233/JIFS-224141

[16] Manaa M., Hussain S., Alasadi S., and Al-Khamees H., "DDoS Attacks Detection Based on Machine Learning Algorithms in IoT Environments," *Inteligencia Artificial*, vol. 27, no. 74, pp. 152-165, 2024. DOI: 10.4114/intartif.vol27iss74pp152-165

[17] Munier N., "A New Approach to the Rank Reversal Phenomenon in MCDM with the SIMUS Method," *Multiple Criteria Decision Making*, vol. 11, pp. 137-152, 2016. DOI: 10.22367/mcdm.2016.11.09

[18] Sałabun W. and Piegat A., "Comparative Analysis of MCDM Methods for the Assessment of Mortality in Patients with Acute Coronary Syndrome," *Artificial Intelligence Review*, vol. 48, no. 4, pp. 557-571, 2017. DOI: 10.1007/s10462-016-9511-9

[19] Sonal. and Ghosh D., "Impact of Situational Awareness Attributes for Resilience Assessment of Active Distribution Networks Using Hybrid Dynamic Bayesian Multi Criteria Decision-Making Approach," *Reliability Engineering and System Safety*, vol. 228, pp. 108772-108796, 2022. https://doi.org/10.1016/j.ress.2022.108772

[20] Swathi T., Kasiviswanath N., and Rao A., "An Optimal Deep Learning-based LSTM for Stock Price Prediction Using Twitter Sentiment Analysis," *Applied Intelligence*, vol. 52, no. 12, pp. 13675-13688, 2022. DOI: 10.1007/s10489-022-03175-2

[21] Tan L., Yu K., Ming F., Cheng X., and Srivastava G., "Secure and Resilient Artificial Intelligence of Things: A Honeynet Approach for Threat Detection and Situational Awareness," *IEEE*

Consumer Electronics Magazine, vol. 11, no. 3, pp. 69-78, 2022. DOI: 10.1109/MCE.2021.3081874

[22] Tavana M., Soltanifar M., and Santos-Arteaga F., "Analytical Hierarchy Process: Revolution and Evolution," *European Journal of Operational Research*, vol. 326, no. 2, pp. 879-907, 2023. https://doi.org/10.1007/s10479-021-04432-2

[23] Wieckowski J., Kizielewicz B., Shekhovtsov A., and Sałabun W., "RANCOM: A Novel Approach to Identifying Criteria Relevance Based on Inaccuracy Expert Judgments," *Engineering Applications of Artificial Intelligence*, vol. 122, pp. 106114, 2023. https://doi.org/10.1016/j.engappai.2023.106114

[24] Wozniak M., Silka J., Wieczorek M., and Alrashoud M., "Recurrent Neural Network Model for IoT and Networking Malware Threat Detection," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5583-5594, 2021. DOI: 10.1109/TII.2020.3021689

[25] Wu J., Qiu G., Jiang W., and Jin J., "Federated Learning for Network Attack Detection Using Attention-based Graph Neural Networks," *Scientific Reports*, vol. 14, no. 1, pp. 1-16, 2024. DOI: 10.1038/s41598-024-70032-2

[26] Xie J., "Application Study on the Reinforcement Learning Strategies in the Network Awareness Risk Perception and Prevention," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, pp. 1-12, 2024. DOI: 10.1007/s44196-024-00492-x

[27] Xu F. and Shen T., "Look-Ahead Prediction-based Real-time Optimal Energy Management for Connected HEVs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2537-2551, 2020. DOI: 10.1109/TVT.2020.2965163

[28] Xu Y., Lu X., Cetiner B., and Taciroglu E., "Real-Time Regional Seismic Damage Assessment Framework Based on Long Short-Term Memory Neural Network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 4, pp. 504-521, 2021. https://doi.org/10.1111/mice.12628

[29] Yang H., Zhang Z., Xie L., and Zhang L., "Network Security Situation Assessment with Network Attack Behavior Classification," *International Journal of Intelligent Systems*, vol. 37, no. 10, pp. 6909-6927, 2022. DOI: 10.1002/int.22867

[30] Zhang Y., Chen J., Wang D., Hu M., and Chen L., "The Bidirectional Gate Recurrent Unit Based Attention Mechanism Network for State of Charge Estimation," *Journal of the Electrochemical Society*, vol. 169, no. 11, pp. 110503, 2022. DOI: 10.1149/1945-7111/ac9d09

[31] Zhu Z. and Hou Z., "Research and Application of Rectified-NAdam Optimization Algorithm in Data Classification," *American Journal of Computer Science and Technology*, vol. 4, no. 4, pp. 106-110, 2021. DOI: 10.11648/j.ajcst.20210404.13

Liwen Xu obtained a Bachelor's degree in Law from Nanjing University of Finance and Economics in 2009 and a Master's degree in Law from Southeast University in 2018. Her research interests include computer science, educational informatization, and practical teaching. Work experience: From 2009 to present, employed as a teacher at the Business School, Jiangsu Open University. Academic situation: Since 2009, she has published 20 provincial-level journal articles as the first author, led 1 general project in philosophy and social sciences in Jiangsu Province, 1 key project in education science planning in Jiangsu Province, participated in 4 general projects in philosophy and social sciences in Jiangsu Province, and 3 university level projects.