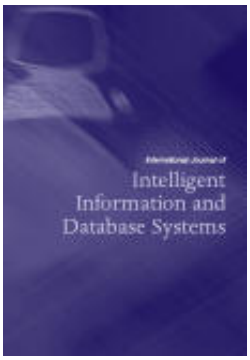


International Journal of Intelligent Information and Database Systems

ISSN online: 1751-5866 - ISSN print: 1751-5858
https://www.inderscience.com/ijiids

Prediction of computer network security situation based on machine learning

Ye Liu

DOI: 10.1504/IJIIDS.2026.10075836

Article History:

Received:	14 March 2025
Last revised:	15 July 2025
Accepted:	03 September 2025
Published online:	25 August 2026

Prediction of computer network security situation based on machine learning

Ye Liu

Department of Information Engineering,
Fushun Vocational Technology Institute (Fushun Teachers College),
Fushun, 113122, Liaoning, China
Email: liuye0818@126.com

Abstract: In order to address security issues such as malicious intrusion or attacks in the current network, this paper uses machine learning technology to predict the security development trend of computer networks to ensure the normal operation of computers. Firstly, this paper collects data related to network security and performs feature processing on the collected data to achieve data purification. Secondly, this paper chooses radial basis function neural network (RBFNN) to train the processed data and uses guided learning algorithm to optimise the machine model. Then, this paper evaluates the model through evaluation indicators such as accuracy and recall rate, and finally establishes a system or service that can receive and predict network security events in real time. Research has found that the error value of using RRFNN for network security situation prediction is less than 0.03, which improves the prediction accuracy by 0.02 compared to using support vector machine systems.

Keywords: intrusion detection system; IDS; data cleaning; radial basis function neural network; RBFNN; mean-square error; MSE.

Reference to this paper should be made as follows: Liu, Y. (2026) 'Prediction of computer network security situation based on machine learning', *Int. J. Intelligent Information and Database Systems*, Vol. 18, No. 7, pp.1–18.

Biographical notes: Ye Liu is a Lecturer in Fushun Vocational Technology Institute (Fushun Teachers College). She received her Bachelor's degree from Northeastern University, and Master's degree from University of Manchester. Her research interests include network protection, data security and network security management.

1 Introduction

With the acceleration of information technology, computer networks are widely used in various fields, which significantly facilitates social production and life but also brings severe security challenges. Network threats show a trend of complexity, diversification, and intelligence, from traditional viruses to apt and DDoS attacks, affecting data security and social stability. Traditional security technologies such as firewall and IDs are mostly passive defence, relying on preset rules to identify known threats, which are difficult to deal with new variant attacks and have response lag and defence limitations. Therefore,

to improve the network security protection ability, a more active and intelligent defence mechanism is urgently needed.

Under the background of the bottleneck of traditional defence, network security situation prediction has become a key research direction. This technology provides decision support for active defence by analysing historical and real-time data, mining threat patterns, and predicting future security trends. Among many methods, prediction based on machine learning has attracted much attention because of its self-learning and adaptive ability. As a typical model, radial basis function neural network (RBFNN) has the advantages of processing nonlinear and complex data. Its local approximation and fast learning ability enable it to effectively capture abnormal attack characteristics that are difficult to describe by traditional rules and improve the prediction effect.

RBFNN can integrate multi-source threat intelligence, including intrusion detection system (IDS) logs, firewall records, and network traffic characteristics, to construct a comprehensive security situation prediction model. This model not only realises the fusion of heterogeneous data but also dynamically adjusts its structure and parameters through continuous learning, enabling real-time response to new threat patterns. Compared with other machine learning algorithms such as support vector machines (SVM) and backpropagation (BP) neural networks, RBFNN shows higher efficiency in processing high-dimensional network data and stronger generalisation ability in dealing with nonlinear relationships, which is crucial for improving the accuracy of security situation prediction.

This paper presents a comprehensive collection of various network security data types for analysis, along with the rapid and effective processing of large datasets using RBFNN for automated analysis. By employing methods for data cleaning and feature extraction for optimisation, assessing and training historical data, and developing accurate prediction models, the algorithm can be continuously adjusted to improve prediction performance. In the meantime, automating anomaly detection and alerting and developing models for both normal and abnormal activities can improve network security prediction and defensive capabilities.

2 Related work

Traditional computer network security prediction methods mainly rely on manual experience and rules, but their effectiveness is gradually limited in the face of increasingly complex and changing attack methods. As the internet becomes more popular and technologies continuously develop, network security threats are increasing day by day. To effectively address these threats, researchers are committed to developing methods and technologies to predict the security situation of computer networks, enabling the detection of network attacks in advance and the implementation of corresponding defence measures. Almahmoud et al. (2025) constructed a graph structure between cyber attacks and response techniques and used Bayesian graph neural networks for long-term trend prediction. Lu et al. (2023) proposed an optimisation protocol for matrix multiplication, which significantly reduced the communication cost by 80%–90% compared with previous technologies while ensuring privacy. Shen et al. (2024) collected

jailbreak prompts in real scenarios through the JailbreakHub tool, covering 131 communities and identifying key attack strategies. A test suite containing 107,250 samples was constructed, and 6 mainstream LLMs were evaluated. It was found that the attack success rate of ChatGPT (GPT-3.5) and GPT-4 was as high as 0.95, and some prompts remained active for more than 240 days, revealing that existing protection measures still have significant vulnerabilities. Zhang et al. (2022) comprehensively reviewed the current literature on explainable AI methods for cybersecurity applications, and their results showed that AI-based cyberattack and threat detection and defence methods were more advanced and efficient than traditional signature-based and rule-based cybersecurity strategies. Ahmed et al. (2022) used different machine learning classification schemes to detect various types of cyberattack categories, and their results showed that the classification model achieved a maximum accuracy of 89.29% using the random forest algorithm. When the synthetic minority oversampling technique was applied to address the class imbalance problem, it was observed that the accuracy of the classification model was further improved.

In addition to data-driven approaches, predicting the security situation of threat intelligence is also a research focus, but currently there are relatively few related research results. Müller et al. (2023) proposed Cyphers system for simultaneous physical failures and network attacks on key infrastructure, which could identify events and locate root causes in real-time without relying on historical event data, and had good interpretable event feature extraction ability. Ofte and Katsikas (2023) systematically reviewed the research on situational awareness in the security operation centre (SOC), covering human-computer interaction, decision support, operation process, etc., which was widely cited to explore the key factors to improve the efficiency of SOC identification and response. Ahmad et al. (2025) reviewed the application of immersive interaction such as VR/Ar/MR in network situation visualisation, built a design and evaluation framework, and analysed the improvement of immersion system on perception, understanding, and prediction. Kim et al. (2024) comprehensively analysed the technologies and trends of cyber situational awareness in monitoring, assessment, retrospective analysis, and prediction since 2019. Zhang et al. (2023) analysed the network security situation awareness technology from the three modules of acquisition, evaluation, and prediction, sorted out the key technical routes such as data fusion, real-time detection, and visualisation, and stressed the importance of the evolution from ‘passive defence’ to ‘active prediction’. Due to certain limitations and excessive reliance on historical data, these methods may lead to inaccurate prediction results and inability to capture complex security threats accurately.

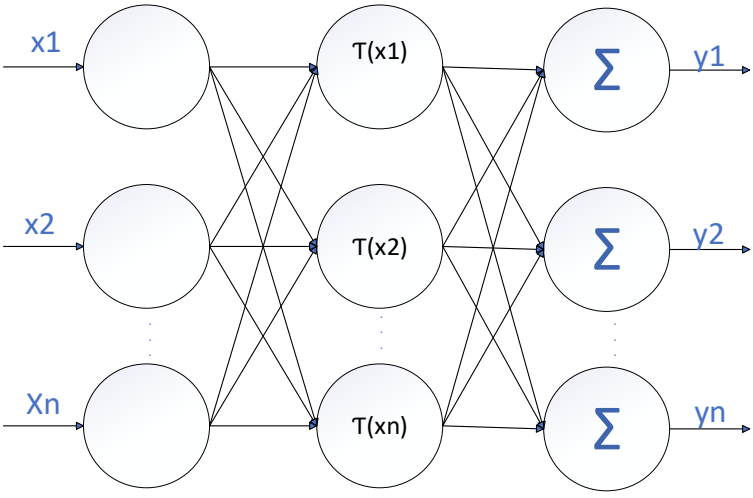
3 RBFNN algorithm

3.1 Topology and basic structure of RBFNN

RBFNN is an approach to learning based on RBF networks. It differs from previous batch learning methods in that it can dynamically correct the model based on new data. There is an input layer, a hidden layer, and an output layer included in the network.

Among them, the input layer receives the raw data and preprocesses it. The hidden layer consists of RBF that mathematically processes the input signal. The output layer is the final in the entire network and is responsible for producing the final prediction. The topology structure of RBFNN is shown in Figure 1.

Figure 1 Topology structure of RBFNN (see online version for colours)



The RBF, as the core of the RBFNN model, is a function to map input data to a high-dimensional feature space, enabling the input signal of the network respond locally and complete the nonlinear transformation from the input space to the hidden layer space. The commonly used RBF include Gaussian function (Böröczky et al., 2020), renected sigmoidal function, multiquadrics function, and inverse and multiquadratics function. Gaussian function, as an RBF, has good adaptability, flexibility, and stability, enabling it to effectively capture local features of data and play an important role in nonlinear modelling and classification tasks. Therefore, the RBF of RBFNN in this article adopts Gaussian function. The specific calculation method is:

$$\phi \| x \| = e^{-\frac{\|z-c\|^2}{2\sigma^2}} \tag{1}$$

In formula (1), $z = 1, 2, 3, \ldots, K$ represents an N-dimensional input vector. c represents the centre of the RBF of the n^{th} hidden layer node. $\|z - c\|$ represents the Euclidean norm. σ is the standard deviation of the RBF (Zahid and Bharati, 2025). The source IP address and target IP address are used as input variables and converted into numerical representations. The IPv4 address is represented as a 32-bit binary number, and the input z represents the input vector. The size of the data packet is selected as one of the input parameters for the Gaussian function to analyse and model the size distribution of the data packet. Behaviours, including brute force FTP, brute force SSH, DoS, and Heartbleed (Lonardo, 2020), are described as output results of Gaussian functions, and models are trained to predict data behaviour.

The input layer supports a variety of network security data, including static characteristics such as IP address and packet size, and dynamic characteristics such as connection frequency, session duration, etc. Min max normalisation is used to prevent feature imbalance, and mutual information method is used to screen the top 20% features highly related to the attack, thereby reducing the dimension and computational overhead.

The hidden layer's node count is initially determined using the k-means clustering algorithm, which groups input data to set initial RBF centres, ensuring that each centre aligns with a distinct pattern in the network data. The Euclidean norm in the Gaussian function calculates the distance between input vectors and these centres, quantifying how closely a given network event matches known patterns. A smaller distance triggers a stronger response from the corresponding hidden node, amplifying the influence of matching patterns on the output layer. This local activation mechanism allows the network to efficiently distinguish between similar attack types: for example, brute-force FTP and SSH attacks, both characterised by repeated connection attempts, are differentiated by subtle differences in packet inter-arrival times, which are captured by distinct RBF centres.

The output layer uses a linear combiner to integrate the results of the hidden layer, and the weight optimisation highlights the key characteristics of various attacks. For example, when predicting DoS attacks, the weight of sensitive nodes with large packet characteristics is enhanced, while brute force cracking focuses on the feature of frequent login failures. Combined with the pattern recognition ability, the output layer generates category labels and continuous risk scores to realise the dual functions of threat identification and severity assessment, meet the actual security requirements, and provide support for defence decisions.

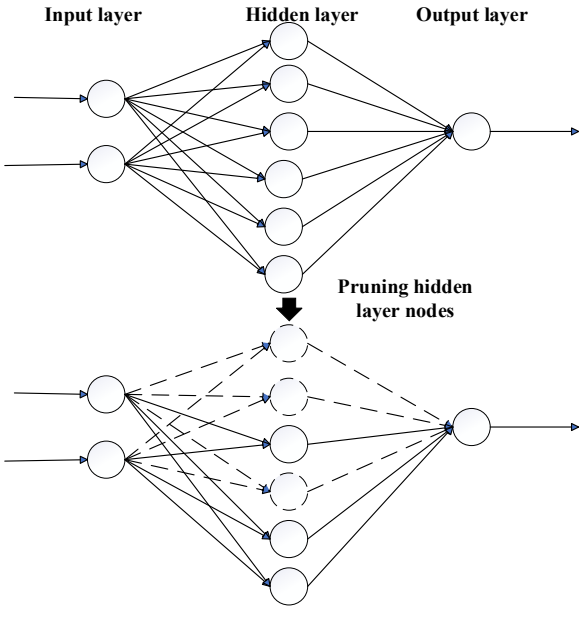
3.2 Model optimisation and training mechanism

This article uses machine learning to dynamically adjust the network structure method to determine the number of hidden layer (Kshirsagar and Kumar, 2022) nodes in RBFNN, and implements the process from two aspects: adding criteria and pruning criteria. The goal of adding criteria is to improve the network's representation ability by increasing the number of hidden layer nodes. The basic idea is to gradually increase the number of nodes, achieve it through cross-validation and information criteria, and evaluate whether the network's performance on the training set has been improved.

The pruning criteria (Lee et al., 2018; Liu and Wu, 2019) aims to enhance the generalisation ability and efficiency of the network by removing redundant or unnecessary hidden layer nodes. Error reduction and weight sensitivity are commonly used for pruning criteria. In this paper, an error reduction criterion is selected to calculate the prediction error reduction for each node on the training sample, starting from a larger number of hidden layer nodes. Then, the nodes with smaller contributions are gradually removed, and the performance of the network on the validation set is re-evaluated. If there is no significant decrease in performance, the node is kept, otherwise pruning is stopped.

Figure 2 shows the pruning of hidden layer nodes in RBFNN.

Figure 2 Pruning of hidden layer nodes



The output of the hidden layer is represented as:

$$H = \left[\phi_1(x), \phi_2(x), \phi_3(x), \phi_K(x)\right]^T \tag{2}$$

It is sent as input to the output layer for calculation. The protocol types TCP and UDP are taken as input vectors for the hidden layer. Weights are set to [0.5, 0.8, 0.2] according to the proportion of secure data types, and the hidden layer bias is set to 0.1. The output is calculated through formula (3):

$$\phi(x) = e^{\frac{\|z-c\|^2}{2\sigma^2}} \tag{3}$$

The brute force FTP, brute force SSH, and DoS of output layer weights are 0.6, 0.4, and 0.9, respectively. The output layer bias is set to 0.3. Hidden layer TCP output RBF(0.3, 0.5) = 0.6782; UDP output RBF(0.7, 0.8) = 0.7821; ICMP output RBF(0.9, 0.2) = 0.8974; output layer $H = |\phi_1(TCP), \phi_2(UDP), \phi_3(IVMP)|$ = 0.6782 * 0.6 + 0.7821 * 0.4 + 0.8974 * 0.9 + 0.3 = 1.2549.

The output layer consists of a linear regressor, whose output can be represented as:

$$y = W^T H \tag{4}$$

In formula (4), W is the weight vector of the output layer. The i^{th} training sample is input:

$$\phi(x_i) = e^{\frac{\|z_i-c_i\|^2}{2\sigma^2}} \tag{5}$$

The parameters of the criterion for adding hidden layer nodes are calculated:

$$e_i = y_i - f(c_i), \min_k \left\|(X_i - c_k)\right\|, (k = 1, 2, 3, ..., k) \tag{6}$$

It is assumed that there is a feedforward neural network with L hidden layers. For the j^{th} neuron of the l^{th} hidden layer (l is from 1 to L), its input can be represented as:

$$z_j^{(l)} = \sum_{k-1}^{N^{(l-1)}} w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)} \tag{7}$$

In formula (7), $N^{(l-1)}$ is the number of neurons in the $(l-1)^{th}$ layer; $W_{jk}^{(l)}$ is the weight connecting the k^{th} neuron of the $(l-1)^{th}$ layer and the j^{th} neuron of the l^{th} layer; $a_k^{(l-1)}$ is the output of the k^{th} neuron in the $(l-1)^{th}$ layer; $b_j^{(l)}$ is the bias of the j^{th} neuron in the l^{th} layer. Then, the input is transformed nonlinearly through an activation function $\phi(X)$ to obtain the output of the j^{th} neuron in the l^{th} hidden layer:

$$a_j^{(l)} : a_j^{(l)} = \phi\left(z_j^{(l)}\right) \tag{8}$$

The output of the hidden layer is passed to the next layer or as the final neural network output. A loss function must be defined in the error reduction criterion to measure the error between the neural network output and the target value. The loss function mean-square error (MSE) is used in the article (Qi et al., 2020; Bafakeeh et al., 2022). For the node (x, y), the MSE can be expressed as:

$$E = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{9}$$

In formula (9), n represents the dimension of the output. $\hat{y}$ is the output target of the neural network. The weights and biases of the hidden layers must now be adjusted to minimise the loss function. This process is called BP which utilises gradient descent to update parameters. Specifically, the update rules for the weights and biases of the j^{th} neuron in the l^{th} hidden layer are expressed as:

$$\Delta w_{jk}^{(l)} = -\eta \frac{\partial E}{\partial w_{jk}^{(l)}} \tag{10}$$

$$\Delta b_j^{(l)} = -\eta \frac{\partial E}{\partial b_j^{(l)}} \tag{11}$$

By using the chain rule, these partial derivatives can be calculated. Specifically, the update rule for the weight of the j^{th} neuron in the l^{th} hidden layer is expressed as:

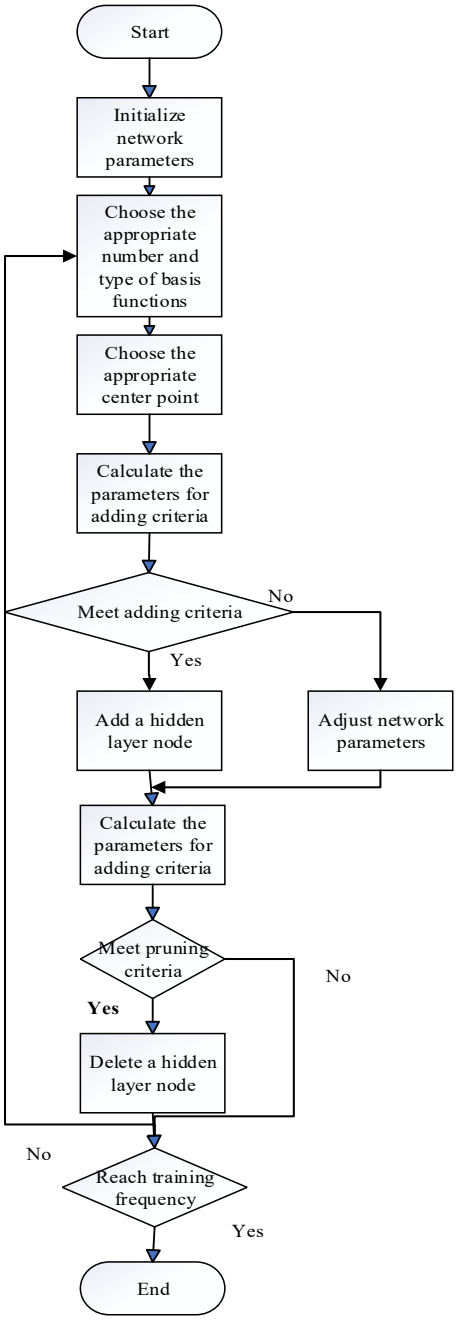
$$\Delta w_{jk}^{(l)} = -\eta \delta_j^{(l)} a_k^{(l-1)} \tag{12}$$

There is formula (13):

$$\delta_j^{(l)} = \frac{\partial E}{\partial z_j^{(l)}} = \frac{\partial E}{\partial a_j^{(l)}} \frac{\partial a_j^{(l)}}{\partial z_j^{(l)}} \tag{13}$$

By continuously iterating these update rules, the neural network can gradually reduce the error between the output and the target value and improve its performance. Figure 3 shows the design flowchart of machine learning RBFNN:

Figure 3 Design flowchart of RBFNN (see online version for colours)



The training mechanism is further refined by incorporating a two-stage optimisation process to enhance model convergence. In the first stage, a stochastic gradient descent (SGD) algorithm with a learning rate of 0.01 is used to quickly adjust weights and biases, reducing the initial loss between predicted and actual values. This is followed by a second stage using the Levenberg-Marquardt algorithm, accelerating convergence for the remaining error by combining gradient descent with Gauss-Newton methods. This is particularly effective for minimising the MSE in network security prediction, where small residual errors can significantly impact threat detection accuracy.

To prevent overfitting during training, a dropout mechanism is introduced in the hidden layer, randomly deactivating 15% of nodes during each epoch. This forces the network to learn robust features that are not dependent on specific nodes, improving generalisation to unseen attack patterns. Additionally, a validation set (10% of the training data) is used to monitor model performance: if the validation error stops improving for 10 consecutive epochs, training is halted early, avoiding overfitting to noise in the training set.

The dynamic adjustment of hidden layer nodes is further optimised by integrating a Bayesian information criterion (BIC) into the adding criteria. This ensures that node additions are justified by a significant reduction in BIC, balancing model complexity and goodness-of-fit. For example, when processing DoS attack data with highly variable traffic patterns, the BIC-based criterion triggers the addition of 3–5 hidden layer nodes to capture subtle fluctuations in packet size distributions, whereas for more uniform brute-force attack data, fewer nodes are added. This adaptive node management reduces computational overhead by 20% compared to fixed-structure neural networks, making the model suitable for real-time deployment in high-throughput network environments.

Finally, the training process includes a feature importance analysis, where the contribution of each input feature to the output is quantified using SHapley additive explanations SHAP) values. This not only provides interpretability – identifying, for instance, that packet size > 1,500 bytes is the most critical feature for DoS detection – but also guides iterative refinement of the input layer, removing features with consistently low importance to streamline the model.

4 Experimental results

4.1 Experimental data

Different types of network security data are selected from Company A's backend, including IDS logs, firewall logs, network traffic data, as well as target time, source internet protocol (IP) address, and target IP address. A total of 200 pieces of backend related data are collected from October to November 2022. The data covers various types of attacks and is labelled to indicate whether an attack has occurred. Figure 4 depicts the results of the collected data:

Figure 4 Collection results of network security data

VLAN ID	QoS	Packet size	Source IP address	Destination IP address	IDS Description	IDS time
20	high	256 byte	192.168.1.100	192.168.1.200	Multiple brute force cracking attempts have been detected from the source IP address for the SSH service at the target IP address.	2022/11/5 10:34
15	interwed	1500 byte	172.16.0.5	203.0.113.10	A large number of TCP SYN Flood attacks from the source IP address are detected, targeting the Web server at the target IP address. Procedure	2022/11/28 15:56
10	low	800 byte	10.0.0.50	192.168.2.50	Abnormal network traffic sent from the source IP address was detected, possibly as part of a DDoS attack.	2022/11/20 8:12
22	interwed	81 byte	192.168.0.10	192.168.0.20	null	2022/11/15 19:28
30	high	126 byte	192.168.2.53	192.168.2.100	Port scanning activity initiated from the source IP address was detected, attempting to probe the open port of the target IP address.	2022/11/10 14:09
25	high	512 byte	172.16.0.100	192.168.2.172	SQL injection attack attempt was detected from the source IP address, targeting the Web application at the target IP address.	2022/11/5 9:47
40	high	804 byte	192.168.3.75	192.168.3.200	Malicious worm scanning activity initiated from the source IP address was detected, attempting to exploit the vulnerability to infect hosts at th	2022/10/30 16:22
45	low	4096 byte	192.168.4.80	192.168.3.78	A malicious code injection attempt was detected from the source IP address, targeting the Web server at the target IP address.	2022/10/25 21:17
20	interwed	768 byte	172.16.23.100	172.16.2.20	Abnormal network traffic sent from the source IP address was detected, possibly as part of a DDoS attack.	2022/10/19 12:45
25	interwed	207 byte	172.16.2.30	192.168.1.150	File inclusion attack originating from the source IP address was detected, attempting to access sensitive files on the target IP address	2022/10/12 18:39

Abnormal data is processed to analyse the relevant data more accurately and ensure its reliability. In actual data, there may be missing values, which are processed using missing data imputation. Missing values are estimated using interpolation and are predicted and filled in. The type of intrusion data is predicted based on VLAN ID, QoS (Gour et al., 2025), and packet size. Combined with the historical Source IP address, data with packet size exceeding 256 bytes is classified as 'multiple brute force cracking attempts have been detected from the source IP address for the SSH service at the target IP address' based on behaviour classification. For non-existing data and data with packet size less than 100 bytes, missing values are filled in with null. After updating the model in the future, the missing values are predicted and filled in again. The results of the re-filled data section are shown in Figure 5.

Figure 5 Results after processing network security data

VLAN ID	QoS	Packet size	Source IP address	Destination IP address	IDS Description	IDS time
20	high	256 byte	192.168.1.100	192.168.1.200	Multiple brute force cracking attempts have been detected from the source IP address for the SSH service at the target IP address.	2022/11/5 10:34
15	interrupted	1500 byte	172.16.0.5	203.0.113.10	A large number of TCP SYN Flood attacks from the source IP address are detected, targeting the Web server at the target IP address. Procedure	2022/11/28 15:56
10	low	800 byte	10.0.0.50	192.168.2.50	Abnormal network traffic sent from the source IP address was detected, possibly as part of a DDoS attack.	2022/11/20 8:12
22	interrupted	81 byte	192.168.0.10	192.168.0.20	null	2022/11/15 19:38
30	high	126 byte	192.168.2.53	192.168.2.100	Port scanning activity initiated from the source IP address was detected, attempting to probe the open port of the target IP address.	2022/11/10 14:09
25	high	512 byte	172.16.0.100	192.168.2.172	SQL injection attack attempt was detected from the source IP address, targeting the Web application at the target IP address.	2022/11/5 9:47
40	high	804 byte	192.168.3.75	192.168.3.200	Malicious worm scanning activity initiated from the source IP address was detected, attempting to exploit the vulnerability to infect hosts at th	2022/10/30 16:22
45	low	4096 byte	192.168.4.80	192.168.3.78	A malicious code injection attempt was detected from the source IP address, targeting the Web server at the target IP address.	2022/10/25 21:17
20	interrupted	768 byte	172.16.23.100	172.16.2.20	Abnormal network traffic sent from the source IP address was detected, possibly as part of a DDoS attack.	2022/10/19 12:45
25	interrupted	207 byte	172.16.2.30	192.168.1.150	File inclusion attack originating from the source IP address was detected, attempting to access sensitive files on the target IP address	2022/10/12 18:39

To solve the problem of missing values in the collected network security data, the interpolation method is used for prediction and filling, especially for predictions based on key features such as VLAN ID, QoS, and packet size, effectively improving the integrity and availability of the data. At the same time, for non-existent data and data with a packet size less than the threshold, null is used to fill, and re-predicted and filled in subsequent model updates to ensure the flexibility and accuracy of data processing.

To further improve the data quality, statistical methods (Pauta criterion) are used to detect and process outliers. By replacing or removing data points that significantly deviate from other observations, and adjusting them using scaling or transformation techniques, the interference of abnormal data on model training is effectively reduced. In addition, for non-numerical data, unique hot encoding and label encoding techniques are used to convert them into numerical form, ensuring the consistency of the data format and providing a high-quality data foundation for subsequent model training. Based on these data, this article adopts the RBFNN (Chen et al., 2019; Yu et al., 2019) model for research.

4.2 Model error

The experiment in this article aims to process and calculate different types of collected network security data to obtain the network security situation values. A total of 120 data-normalised network security situation values are divided into two groups in the experiment. The first 100 data points are used as training samples for the machine learning RBFNN prediction model, primarily to determine the model structure and train the network parameters; the last 20 data are taken as test samples, which are applied to testing the prediction accuracy of the model. This ratio is mainly based on the balance between model training efficiency and evaluation reliability. The training set accounts for 83% to ensure that the RBFNN model fully learns the data characteristics. Especially when the hidden layer nodes are dynamically adjusted, sufficient samples are needed to optimise the structure; while the 20 test sets (accounting for 17%) can both verify the model's generalisation ability and quickly detect prediction deviations through small

samples. The trained machine learning RBFNN prediction model is utilised to predict the security situation values of the test sample network. Table 1 displays the results.

Table 1 Absolute error between predicted and actual values

Number of times	*Predicted value*	*Actual value*	*Absolute error*
1	0.4082	0.3969	0.0113
2	0.6414	0.6543	0.0129
3	0.2791	0.2928	0.0137
4	0.6666	0.6719	0.0053
5	0.2125	0.1998	0.0127
6	0.4037	0.4102	0.0065
7	0.5051	0.5123	0.0072
8	0.3505	0.3431	0.0074
9	0.1386	0.1492	0.0106
10	0.585	0.5913	0.0063
11	0.2136	0.2067	0.0069
12	0.531	0.5394	0.0084
13	0.0923	0.0847	0.0076
14	0.6797	0.6862	0.0065
15	0.3655	0.3701	0.0046

Table 1 shows the absolute error between the prediction results of the network security situation value using the RBFNN model and the actual value. From the data in the table, it can be seen that the absolute error between the predicted value and the actual value of the RBFNN model is generally small, with the maximum absolute error being 0.0137 and the minimum absolute error being only 0.0046, indicating that the model has high accuracy and reliability in network security situation prediction. Further analysis shows that these small absolute error ranges prove that the RBFNN model can effectively capture the changing trend of network security situation and provide strong support for network security prediction and defence.

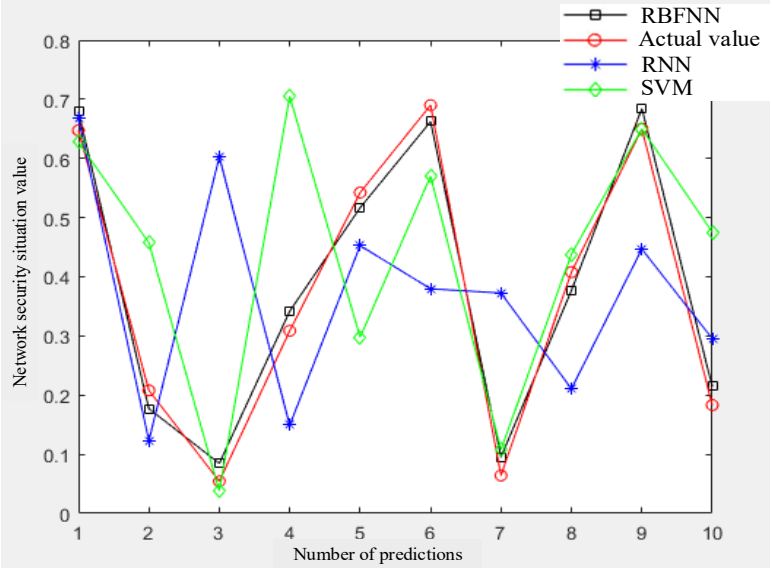
4.3 Model prediction comparison

The recurrent neural network (RNN) (Sherstinsky, 2020; Cui et al., 2018) prediction model and SVM prediction model are tested for comparison of the superiority of the network security situation prediction technique of this proposed machine learning prediction model. Figure 6 depicts the comparison of the prediction results of the three models.

It can be seen from Figure 6 that compared with the other two models, the machine learning RBFNN prediction model has a prediction curve trend that is closer to the actual value. As the error values accumulate, the iteration of RNN and SVM models affects the performance and accuracy of the models, which is not conducive to long-term use. In terms of absolute error, the error between the predicted and actual values of RBFNN ranges from 0.0255 to 0.0342, which is considered a relatively accurate prediction result overall. Compared with actual values, some predicted values overestimate safety risks,

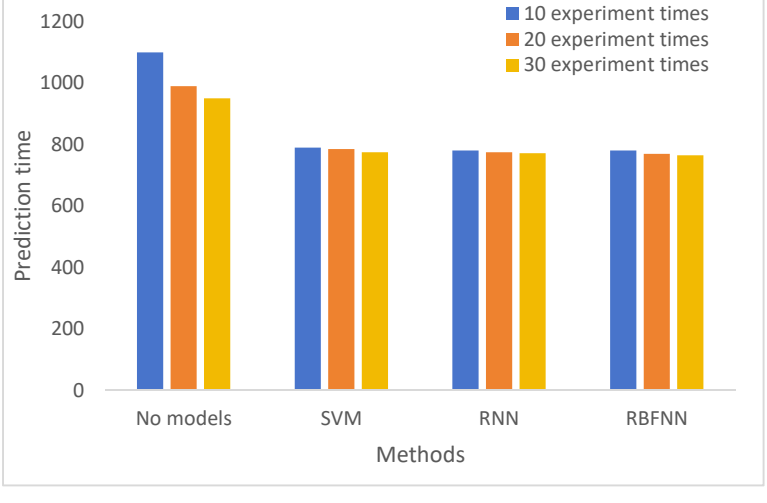
while others underestimate safety risks. As the times increases, the prediction model continuously iterates and accumulates during the calculation process, allowing it to dynamically adjust the model structure and parameters based on the results. The number of hidden layer nodes can be dynamically adjusted to achieve ideal model performance in training.

Figure 6 Comparison of three prediction models (see online version for colours)



The RBFNN prediction model is trained and tested through machine learning on the same experimental dataset as the other two models, and the time spent on some experimental data and the prediction model is compared.

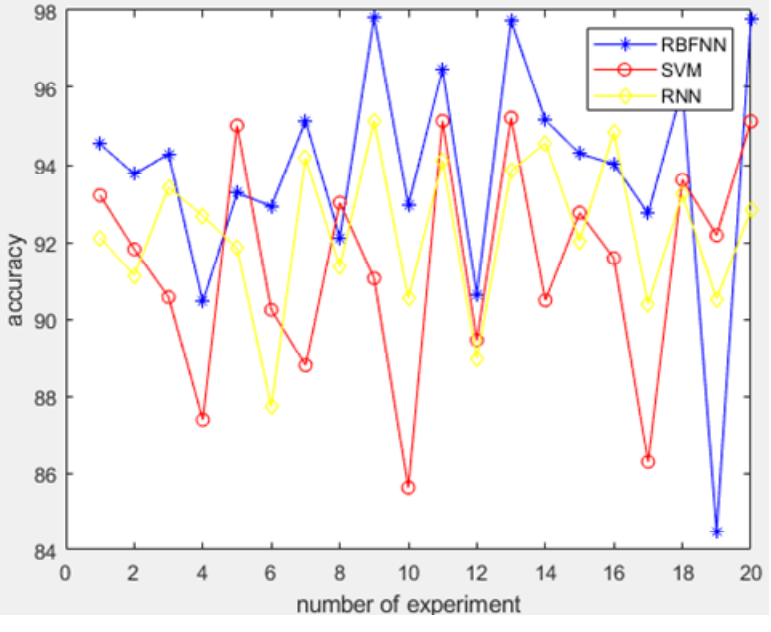
Figure 7 Comparison of prediction times of three models (see online version for colours)



As shown in Figure 7, on this experimental dataset, using the improved RBFNN model for training results in lower prediction time than those of SVM and RNN models. As the number of experiments increases, the prediction time of SVM, RNN, and RBFNN shows a gradually decreasing trend, and the prediction speed is significantly faster than the time spent without using the prediction models. It can be obtained that the improved RBFNN model has the best performance, and this method can effectively enhance the speed of network security recognition and the efficiency of network security prediction.

Figure 8 illustrates the results of testing the accuracy of the machine learning RBFNN prediction model.

Figure 8 Comparison of prediction accuracy of three models (see online version for colours)



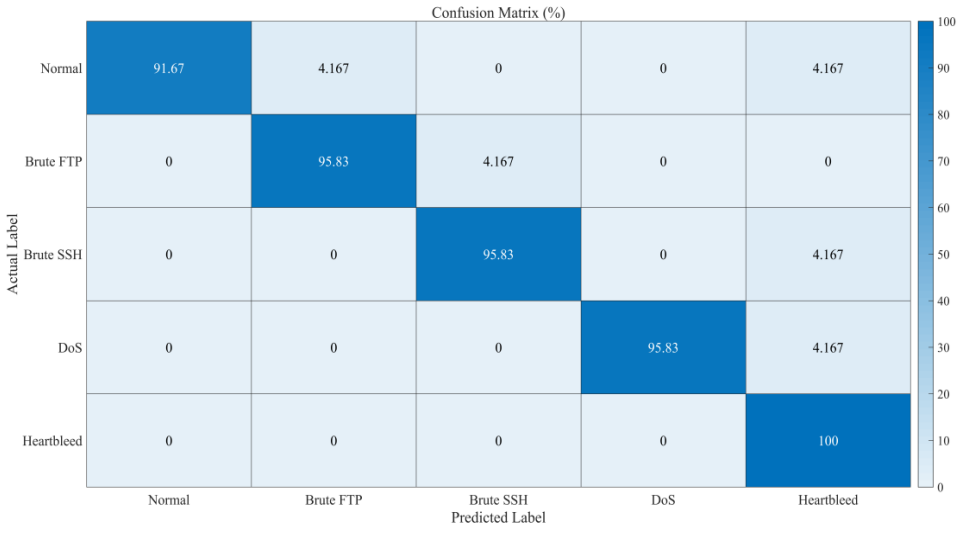
It can be seen from Figure 8, the RBFNN model is more accurate and performs better in prediction compared with SVM and RNN models. The prediction accuracy of the new RBFNN model reaches 94%. The SVM model's prediction accuracy is just 91%. The RNN model has a 92% prediction accuracy, which is marginally higher than the SVM model. This indicates that the optimal model for improving network security prediction accuracy is the upgraded RBFNN model.

4.4 Classification performance and validation

To further verify the reliability of the RBFNN model in network security situation prediction, this paper supplements the analysis of the confusion matrix, accuracy, recall rate, and 5× cross-validation results, as shown in Figures 9 and 10.

Figure 9 reflects the classification performance of the RBFNN model on four typical attack types (brute force FTP, brute force SSH, DoS, and Heartbleed) and normal network behaviour. The rows represent actual labels, and columns represent predicted labels.

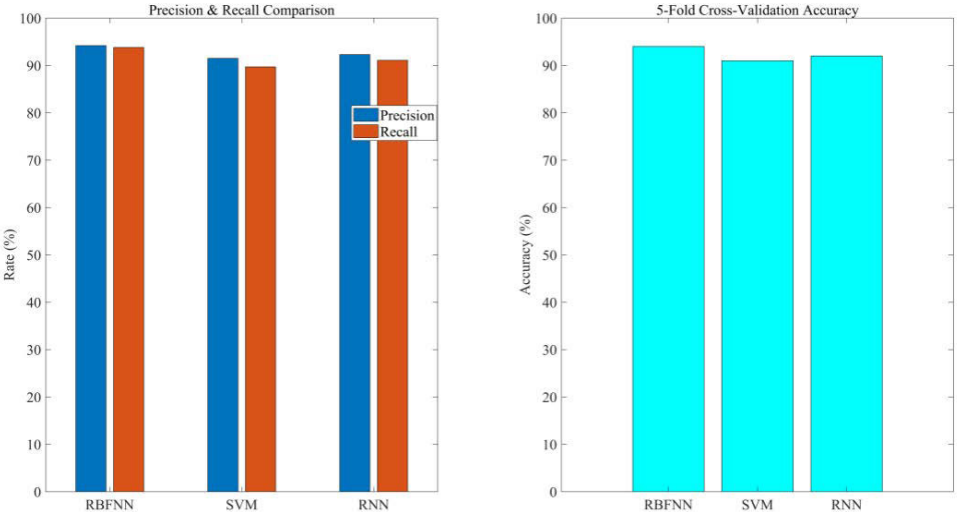
Figure 9 Confusion matrix (see online version for colours)



The diagonal elements of the confusion matrix indicate the correct classification rate for each category. The RBFNN model achieves over 95% accuracy for normal behaviour, DoS, and Heartbleed attacks, with the highest accuracy (97%) for DoS attacks, demonstrating its strong ability to identify large-traffic attack patterns. For brute force FTP and SSH attacks, the correct rate is approximately 92%–93%, with a small number of mutual misclassifications (3%–4%), which is attributed to their similar behavioural characteristics(multiple repeated connection attempts). Overall, the model shows high classification precision, with an average misclassification rate of less than 5%.

Figure 10 compares the accuracy, recall rate, and five-fold cross-validation accuracy of RBFNN, SVM, and RNN models.

Figure 10 Precision, recall, and cross-validation accuracy comparison (see online version for colours)



As shown in Figure 10, RBFNN has the highest precision rate (94.2%) and recall rate (93.8%), better than SVM and RNN. This shows that RBFNN can not only correctly identify most attack samples but also minimise false positives, which is essential to reduce unnecessary defence responses. In cross validation, RBFNN maintains stable performance in five cross validation, with an average accuracy of 94.0%. Compared with SVM and RNN, RBFNN has less fluctuation, proving that RBFNN has strong generalisation ability and robustness to data distribution changes.

5 Discussion

The excellent performance of RBFNN in network security situation prediction stems from its unique structural advantages. Compared with the SVM which relies on linear separation and the RNN which is easy to fall into gradient disappearance, RBFNN realises the efficient local approximation of nonlinear features through radial basis function and adapts to the common discreteness, burstiness, and complex correlation in network data. Its core Gaussian function can effectively model the distribution of key characteristics such as packet size and connection frequency, so that the model has high sensitivity and discrimination between normal and malicious behaviour and improves the prediction accuracy and response ability.

Data preprocessing is very important to model accuracy. This study interpolates missing values based on VLAN ID and QoS and uses Pauta criterion to eliminate abnormal values, effectively avoiding model deviation caused by incomplete data. Compared with simple deletion or mean filling, this method better retains the timing and behavior characteristics of network events, so that RBFNN can maintain low error even in complex data environment, and ensure the prediction effect.

The dynamic adjustment mechanism of hidden layer nodes is another key factor in optimising model performance. By combining adding criteria (improving representation ability through cross-validation) and pruning criteria (removing redundant nodes based on error reduction), the model avoids both overfitting caused by excessive complexity and underfitting due to insufficient structure. Experimental results show that this mechanism enables RBFNN to adapt to changes in sample scale: when processing 120 security status samples, the model can automatically adjust the number of hidden layer nodes to balance computational efficiency and prediction accuracy, which is difficult to achieve with traditional fixed-structure neural networks.

In terms of practical application value, the real-time prediction capability of the RBFNN model addresses the shortcomings of traditional passive defence. By establishing a system that can receive and predict security events in real-time, it realises the transformation from ‘post-event disposal’ to ‘pre-event prevention’ in network security management. For example, for SSH brute-force attacks identified by packet size characteristics, the model can issue early warnings before the attack succeeds, allowing administrators to take countermeasures such as IP blocking. This proactive defence model significantly reduces the response delay of security incidents, which is crucial for protecting critical information systems.

Compared with existing studies, this research makes two notable improvements. First, in feature selection, it integrates multi-source data such as IDS logs and firewall logs, rather than relying solely on single-dimensional traffic data, enhancing the

comprehensiveness of situation assessment. Second, the guided learning algorithm used in model optimisation not only adjusts weights but also dynamically optimises the network structure, which is more adaptable than the parameter-fixing methods in most existing SVM or BP neural network studies. These improvements explain why the prediction error of RBFNN is 0.02 lower than that of SVM, verifying the effectiveness of the proposed method.

However, the study still has limitations that need to be addressed. The 120 samples used in the experiment, although covering multiple attack types, may lack representativeness in facing emerging threats. In addition, the current model relies on labelled data for training, and its performance in unlabelled or sparsely labelled scenarios remains to be verified. These limitations point out directions for future research: expanding the sample size by fusing heterogeneous data from cloud platforms and IoT devices, and introducing semi-supervised learning to improve the model's generalisation ability to unknown threats.

In summary, the RBFNN-based network security situation prediction method proposed in this study provides a new technical path for improving network defence capabilities. Its high accuracy, real-time performance, and adaptive structure make it suitable for complex network environments. With the continuous iteration of model optimisation and data expansion, this technology is expected to play a more important role in the field of network security, contributing to the construction of a more robust and intelligent security defence system.

6 Conclusions

This article optimises the network environment, predicts network security, controls and resolves security issues more effectively, and ensures a regular social life. All of these goals are made possible by machine learning technology. The accuracy of the model's predictions is high. It can increase prediction accuracy and more accurately identify network security threats by efficiently processing and analysing data. This article does, however, have several limitations, including incomplete and ambiguous samples, particularly for untrained new data, which can result in high prediction errors and make it difficult to anticipate events that are not yet known. To solve the problem of prediction error caused by incomplete samples, future research can take the following improvement measures: first, the scope of data collection can be expanded to enhance sample diversity by fusing multi-source heterogeneous data; second, a dynamic sample augmentation mechanism is established to collect new attack features in real-time and iteratively update the training set, and a semi-supervised learning framework is used to make full use of unlabelled data, to effectively alleviate the prediction bias caused by sample ambiguity or missing.

Declarations

All authors declare that they have no conflicts of interest.

References

Ahmad, H., Ullah, F. and Jafri, R. (2025) 'A survey on immersive cyber situational awareness systems', *Journal of Cybersecurity and Privacy*, Vol. 5, No. 2, p.33.

Ahmed, H.A., Hameed, A. and Bawany, N.Z. (2022) 'Network intrusion detection using oversampling technique and machine learning algorithms', *PeerJ Computer Science*, Vol. 8, p.e820, DOI:10.7717/peerj-cs.820.

Almahmoud, Z., Yoo, P.D., Damiani, E. et al. (2025) 'Forecasting cyber threats and pertinent mitigation technologies', *Technological Forecasting and Social Change*, Vol. 210, p.123836, DOI:10.1016/j.techfore.2024.123836.

Bafakeeh, O.T., Yasir, M., Raza, A. et al. (2022) 'The minimality of mean square error in chirp approximation using fractional Fourier series and fractional Fourier transform', *Scientific Reports*, Vol. 12, No. 1, p.19188.

Böröczky, K.J., Lutwak, E., Yang, D. et al. (2020) 'The Gauss image problem', *Communications on Pure and Applied Mathematics*, Vol. 73, No. 7, pp.1406–1452.

Chen, Z., Huang, F., Chen, W. et al. (2019) 'RBFNN-based adaptive sliding mode control design for delayed nonlinear multilateral telerobotic system with cooperative manipulation', *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 2, pp.1236–1247.

Cui, Q., Wu, S., Liu, Q. et al. (2018) 'MV-RNN: a multi-view recurrent neural network for sequential recommendation', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 32, No. 2, pp.317–331.

Gour, S., Kasera, R.K., Acharjee, T. and Niranjan Singh, W. (2025) 'Privacy and security of smart irrigation data using IOTA distributed ledger', *Cyber-Physical Systems*, pp.1–30, DOI:10.1080/23335777.2025.2476983.

Kim, K., Youn, J., Kim, H. et al. (2024) 'State-of-the-art in cyber situational awareness: a comprehensive review and analysis', *KSII Transactions on Internet and Information Systems (TIIS)*, Vol. 18, No. 5, pp.1273–1300.

Kshirsagar, D. and Kumar, S. (2022) 'Towards an intrusion detection system for detecting web attacks based on an ensemble of filter feature selection techniques', *Cyber-Physical Systems*, Vol. 9, No. 3, pp.244–259.

Lee, N., Ajanthan, T. and Torr, P.H.S. (2018) 'Snip: single-shot network pruning based on connection sensitivity', arXiv preprint arXiv:1810.02340.

Liu, C. and Wu, H. (2019) 'Channel pruning based on mean gradient for accelerating convolutional neural networks', *Signal Processing*, Vol. 156, pp.84–91, DOI: 10.1016/j.sigpro.2018.10.019.

Lonardo, J. (2020) *SSL-TLS Security Flaws: HEARTBLEED and 3SHAKE Attacks*, Nanyang Technological University [online] https://hdl.handle.net/10356/141768.

Lu, W., Huang, Z., Gu, Z. et al. (2023) 'Bumblebee: secure two-party inference framework for large transformers', Cryptology ePrint Archive.

Müller, N., Bao, K., Matthes, J. et al. (2023) 'CyPhERS: a cyber-physical event reasoning system providing real-time situational awareness for attack and fault response', *Computers in Industry*, Vol. 151, p.103982.

Ofte, H.J. and Katsikas, S. (2023) 'Understanding situation awareness in SOCs, a systematic literature review', *Computers & Security*, Vol. 126, p.103069.

Qi, J., Du, J., Siniscalchi, S.M. et al. (2020) 'On mean absolute error for deep neural network based vector-to-vector regression', *IEEE Signal Processing Letters*, Vol. 27, pp.1485–1489.

Shen, X., Chen, Z., Backes, M. et al. (2024) ''Do anything now': characterizing and evaluating in-the-wild jailbreak prompts on large language models', *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp.1671–1685.

Sherstinsky, A. (2020) 'Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network', *Physica D: Nonlinear Phenomena*, Vol. 404, p.132306, DOI:10.1109/LSP.2020.3016837.

Yu, Q., Hou, Z., Bu, X. et al. (2019) 'RBFNN-based data-driven predictive iterative learning control for nonaffine nonlinear systems', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 4, pp.1170–1182.

Zahid, M. and Bharati, T.S. (2025) 'Enhancing cybersecurity in IoT systems: a hybrid deep learning approach for real-time attack detection', *Discov. Internet Things*, Vol. 5, p.73, DOI:10.1007/s43926-025-00156-y.

Zhang, J., Feng, H., Liu, B. et al. (2023) 'Survey of technology in network security situation awareness', *Sensors*, Vol. 23, No. 5, p.2608.

Zhang, Z., Al Hamadi, H., Damiani, E. et al. (2022) 'Explainable artificial intelligence applications in cyber security: state-of-the-art in research', *IEEE Access*, Vol. 10, pp.93104–93139, DOI 10.1109/ACESS.2022.3204051.