Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Deep learning-based network information security threat prediction and defense mechanism design

Shan Lu[1, ✉]

[1] *Public Security Information Technology and Intelligence College, Criminal Investigation Police University of China, Shenyang, Liaoning, 110000, China*

ABSTRACT

With the continuous development of the network environment, the traffic data in the network increasingly presents high-dimensional, huge and complex characteristics, and the network threat is also increasing, the network information security threat prediction and defense mechanism plays an irreplaceable position in network security. Based on the general process of network anomaly detection, combined with deep learning algorithms, the article proposes a network anomaly detection method based on data enhancement to improve the detection accuracy of network anomaly detection model. Self-attention mechanism is embedded in the neural network framework to accomplish the improved SA-GRU network information security threat prediction method. In the performance index comparison experiments of network security posture values predicted using different prediction models, the average absolute error of the training data of the results predicted by this paper's model is 0.00266, and the average absolute error of the test data is 0.00369, and the prediction accuracy of this paper's model prediction is significantly higher than that of other deep learning methods. This verifies the effectiveness of the method proposed in this paper. Finally, based on the experimental results, the network information security defense mechanism is proposed from the three levels of data encryption, the use of secret keys and intrusion detection.

Keywords: deep learning, network information security, SA-GRU, threat prediction, network anomaly detection

1. Introduction

The rapid development of computer networks has not only facilitated the flow of information and the sharing of resources, but also effectively improved work efficiency and quality of life. However, with the rapid development of information and communication technology, computer networks are facing an increasing number of security challenges, from traditional viruses and Trojan horse attacks to

✉ Corresponding author.
E-mail address: Scs3766@163.com (S. Lu).
Received 10 February 2024; Revised 25 June 2024; Accepted 15 October 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-034

sophisticated phishing and ransomware, to highly sophisticated state-level cyber wars and commercial espionage, a great variety of security threats are constantly threatening the information security of individuals, enterprises and even the country [1-4]. Especially with the promotion of new technologies such as the Internet of Things and cloud computing, a large amount of sensitive data flows through the network, making the data security problem more complicated [5-6]. Therefore, improving the protection measures of computer network information security and building a secure network environment has become an important challenge for the society nowadays [7-9].

Deep learning, as a powerful machine learning technology, has achieved remarkable results in various fields. Deep learning is a machine learning technique based on simulating the neural network structure of the human brain, and its core idea is the deep structure of multilayer neural networks [10]. Using deep neural network models to classify, recognize, generate and other tasks for data in different fields can greatly improve the level of intelligence in various fields and bring more convenience to human life [11-13]. In the field of network security, deep learning is widely used in intrusion detection, security vulnerability mining and malicious code detection [14-16]. Using deep learning techniques, more accurate, efficient and intelligent network security defense strategies can be achieved [17]. It uses deep learning technology to analyze and predict the characteristics and patterns of network security threats, and monitors network traffic in real time to detect and stop attacks in time, thus realizing active defense of network security [18-20]. Therefore, network security threat prediction and defense strategies based on deep learning can effectively improve network security and reduce network security risks.

This chapter first introduces the general process of network anomaly detection in terms of dataset acquisition, dataset preprocessing, model training, and performance evaluation. On the basis of introducing the proposed process of data enhancement based network anomaly detection method and its related algorithmic principles. The network posture prediction method based on SA-GRU is designed by combining the SA-GRU method architecture, Self-Attention structure and SA-GRU's method, and the implementation process and pseudo-code implementation of the related method are given. Finally, based on the CIC-IDS-2017 dataset and CIC-DDoS-2019 dataset, the model in this paper is experimentally verified by binary classification and multiple classification, and compared with other models to verify the effectiveness and advancement of the model. The models and algorithms proposed in this paper are experimentally compared with NARX, single-layer GRU, RNN, CNN, two-layer GRU, and Att-LSTM models and algorithms, which proves that the models and algorithms in this paper are more effective in solving the problem of uncertainty in the prediction of cybersecurity posture in the complex network environment compared to other models and algorithms.

2. Method

2.1. Deep learning based network anomaly detection research

2.1.1. Network Anomaly Detection Overview. Network Anomaly Detection (NAD), as a key technology to improve the security of network systems, is different from the previous passive defense strategy, NAD adopts an active defense strategy. Through real-time monitoring of network behavior, obtaining and analyzing the data generated by the network system, digging deep into the hidden correlation information between the network data features, and once the data related to abnormal behavior is detected, it is fed back to the network administrator, and corresponding strategies are adopted to cope with the network anomalous behavior and minimize the harm brought by the anomalous behavior [21]. Network anomaly detection generally has three processes, network anomaly detection process is shown in Figure 1.

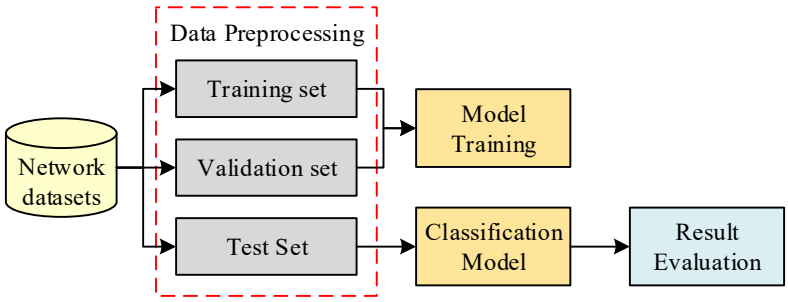

Fig. 1. Network anomaly detection process

2.1.2. Common models for deep learning:

1) Convolutional Neural Networks. Convolutional Neural Networks (CNNs) can automatically extract high-dimensional, nonlinear features of a sample through an end-to-end network structure. Both Convolutional Neural Networks and Fully Connected Neural Networks are made up of neurons, which are fed through the network and finally output the results, which are used to optimize the parameters in the network through a loss function. The difference between convolutional neural network and fully connected network is the difference in the layer structure of the network [22]. Convolutional neural network is mainly composed of input layer, convolutional layer, pooling layer, fully connected layer, classification layer, output layer, depending on the different problems to be dealt with, the number of convolutional layers and activation function vary.

(1) Convolutional layer. Convolutional layer is the core of convolutional neural network, most of the operations are carried out in the convolutional layer. The convolutional layer consists of a number of filters that can be learned, the width and height of the filters are small, and the depth of the filters is consistent with the depth of the input data. The filters slide along the depth and height of the input data to generate an inner product, which forms a new activation map, the output of the convolutional layer.

The following formula is the convolution operation in a convolutional neural network:

$$h_j^{(n)} = \sum_{k=1}^{K} h_k^{(n-1)} \otimes w_{kj}^{(n)} + b_{kj}^{(n)} \tag{1}$$

where here $\otimes$ denotes the 2D convolution and $h_j^{(n)}$ denotes the output of the j th feature map in the n rd hidden layer. Also, $h_k^{(n-1)}$ denotes the k th channel of the $(n-1)$ hidden, $w_{kj}^{(n)}$ denotes the weight of the k th channel in the j th filter of the n th layer, and $b_{kj}^{(n)}$ denotes the associated bias term.

(2) Pooling Layers. Multiple convolutional layers create a large number of feature maps, and in order to reduce the dimensionality of these attributes, an extra layer is usually added to the convolutional layers. This is known as pooling layer. On the other hand, adding a pooling layer also increases the sensory field of the convolutional kernel, making the model more focused with global features, taking out redundant information and keeping important features.

The commonly used pooling methods are maximum pooling, average pooling, etc. Among them, the maximum pooling method is to take the maximum value of so neurons in a specific region (R) to represent the output of the whole region:

$$\gamma_1 = \max_{x_i} (x \in R) \tag{2}$$

In the average pooling method, the value taken is then the average of the region.

(3) Activation function. In neural networks, the activation function is a nonlinear function that is applied to the output of each neuron to increase the nonlinear capability of the model. The activation function can be any nonlinear function. Commonly used activation functions include:

① Sigmoid function

Sigmoid function: sigmoid function is an S-shaped function that compresses the input between 0 and 1. Sigmoid function is commonly used in binary classification tasks. However, due to the exponential operation in the computation, there is a problem of high computation and easy gradient vanishing.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3}$$

② Tanh function

The Tanh function is also a S-form function that compresses the input to between -1 and 1, and outputs 1 when the x value of the input is very large, and 0 when the value of the input is very small.The Tanh function is usually used in multivariate classification tasks. It also suffers from high computational effort and vanishing gradient.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} - e^{-x}} \tag{4}$$

③ReLU function

The RcLU function is a very simple activation function that sets all negative input values to zero and retains only the positive values.The ReLU function is commonly used in Convolutional Neural Networks to speed up the training process and improve accuracy.

$$\mathrm{Re}\,lu(x) = \max(0, x) \tag{5}$$

④ Softmax function

The Softmax function maps multiple inputs between 0 and 1 and they sum to 1. It is commonly used in multivariate classification tasks to convert the output of a neural network into a probability distribution.

$$soft\max(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{6}$$

2) Recurrent Neural Networks:

(1) RNN. Fully connected neural networks and convolutional neural networks can extract features, but it is difficult to extract the connection between features and features, and cannot meet the needs of some tasks that require contextual relationships. In traditional recurrent neural networks, x is the input data, y represents the true value, L represents the loss function, o is the predicted value, h represents the hidden state, W, U, and V represent the full-time matrix. At moment t, the neuron hidden state h_t is jointly determined by the current input value x_t and the neuron hidden state h_{t-1} of the previous moment. In this way, the recurrent neural network realizes the ability to relate the above.

$$h^{(t)} = \phi\left(Uxx^{(t)} + Wh^{(t-1)} + b_h\right) \tag{7}$$

$\phi(x)$ is generally the activation function and b_h is the bias value. $h(t)$ is also used as the output of the network at this moment and also to enter the hidden state operation at the next moment. h The prediction value o is obtained by continuing the operation from the following equation.

$$o^{(t)} = \varphi\left(Vh^{(t)} + b_o\right) \tag{8}$$

where $\varphi(\cdot)$ is the activation function, usually choose softmax function. In order to make the best training results of the model, the loss function is used as the optimization objective function of the model, and the loss function can be selected such as cross entropy.

(2) Long and short-term memory network. Long Short-Term Memory Network (LSTM) can solve the problem of long time dependency to some extent.LSTM consists of a cell state and three gating units (forgetting gate, input gate and output gate) [23]. The forgetting gate filters the transmitted information of the previous moment, choosing to keep the important information and forgetting the useless information. The input gate selectively retains the input information of the current network and proposes invalid information. The output gate is a selective output of the current state. The formula is shown in (9).

$$\begin{cases} i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \\ f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \\ g_t = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \\ C_t = i_t \square\ g_i + f_i \square\ C_{t-1} \\ o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \\ h_t = o_t \square\ \tanh\left(C_t\right) \end{cases} \tag{9}$$

where W_i, W_f, W_o, W_c denote the weight matrices of the input gates, the forgetting gates, the output gates and the cell states, respectively. b_i, b_i, b_o, b_c denote the bias terms of each gating cell as well as the cellular cell, respectively. σ denotes the sigmoid function. $\square$ denotes the multiplication of the corresponding elements of the two vectors involved in the operation. tanh denotes the hyperbolic tangent activation function.

(3) GRU. GRU integrates the forgetting gate and the input gate into a single “update gate”, and no longer outputs C_t, but passes h_t backward as a memory state.GRU has only two gate units, the update gate and the reset gate, which makes the parameters of the network model more concise.

The reset gate combines historical information with new information. The reset gate uses tanh as the activation function.

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right] + b_r\right) \tag{10}$$

where W_r is the weight matrix of the reset gate and b_r is its bias term.

The update gate will synthesize the hidden state h_{t-1} of the previous moment and the current input information.

$$z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right] + b_z\right) \tag{11}$$

$$\tilde{h}_t = \tanh\left(W_h\left[r_t h_{t-1}, x_t\right] + b_h\right) \tag{12}$$

$$h_t = \left(1 - z_t\right) h_{t-1} + z_t \tilde{h}_t \tag{13}$$

where W_z and W_h are the weight matrices and b_z and b_h are their bias terms.

(4) BiGRU. Bi-GRU is a recurrent neural network structure. It consists of gated recurrent units (GRUs) in both directions that process the input sequence in the forward and backward direction respectively, thus enabling better capture of contextual information of the input sequence.Bi-GRU can be used for various tasks such as text categorization, sequence annotation, and machine translation. The basic framework of the recurrent neural network based intrusion detection system is shown in Fig. 2. By utilizing the ability to process time series features that recurrent networks have, the effective correlation information between features is extracted for analysis and prediction, which is used to construct the intrusion detection model.

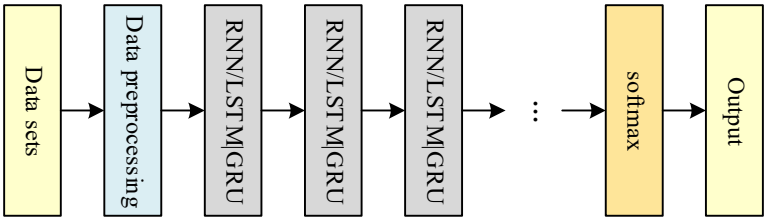

Fig. 2. Schematic diagram of the basic framework of the intrusion detection system

2.1.3. Methodological principles. First, the NSL-KDD dataset is viewed as a matrix X with data features and labels, denoted by m rows and n columns, $X \subseteq R^{m \times n}$, R as the set of real numbers, m as the number of network data streams, i.e., the number of samples, and n as the dimensionality of the network data stream features:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & \cdots & x_{mn} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix} \tag{14}$$

where x_i is the i nd sample of the NSL-KDD dataset.

Secondly, considering that the symbolic features in the dataset cannot be directly involved in the operation and have a certain impact on the model's merits rather than simply discarding them, the three symbolic features, protocol type, service and flag, need to be numericalized. Similarly, the features service and flag are mapped to values from 0 to 69 and 0 to 10, respectively. After numericalization of the network traffic feature matrix X using the mapping method, a new feature matrix X_{new} is obtained and after dividing the dataset, the training set is X_{ntrain} and the test set is X_{ntest}.

Further, the number of minority class samples is increased using the data augmentation algorithm, where p_i is a minority class sample in the training set and p_{new} is the synthesized new sample.

$$p_{new} = p_i' + r_j \times p_{diff}^j, j = 1, 2, \cdots, s \tag{15}$$

where p_i' is the sample to be synthesized in the minority category, p_{diff}^j is the difference between p_i' and its s nearest neighbors, s is an integer between 1 and $k_{neighbors}$, and r_j is a random number between 0 and 1.

After the above process, the training set X'_{ntrain} is obtained after augmentation for a few categories.

Further, different features in the NSL-KDD dataset are considered. The maximization method in the linear proportional transformation method is introduced to normalize the enhanced training set X'_{ntrain} and test set X_{ntest} as shown in Equation (17):

Each column of the training set is a feature, and in turn, the training set X'_{ntrain} is denoted as:

$$X'_{ntrain} = \left[c_1, c_2, c_3, \cdots, c_n\right] \tag{16}$$

$$c'_{i,j} = \frac{c_{i,j}}{c_{\max}} \tag{17}$$

where, c_{ij} is the j rd sample value of the i nd feature before normalization and $c_{\max}$ is the maximum sample value of the i th feature.

The normalized data set can be expressed as:

$$X''_{ntrain} = \left[c_1', c_2', c_3', \cdots, c_n'\right] \tag{18}$$

Similarly, the linear proportional transformation method is introduced to deal with test set X_{ntest}, and the normalized test set is obtained as X'_{ntest}.

Finally, on the basis of data enhancement and data transformation and other operations on the training set and the test set, a combined neural network model combining the series-parallel approach is then established, so as to improve the accuracy of network anomaly detection.

The normalized training set X_{train}, which can be denoted as $x_i = \left[x_{i,1}, x_{i,2}, x_{i,3}, \cdots, x_{i,n}\right]$ by rows, is used as the input of the convolutional neural network, and the training set X_{train} firstly undergoes two convolutional layers with two concatenated convolutional kernels of W_i ($0 \le 1 \le l_f$, l_f represent the number of filters) to compute a new feature v_i^l according to Eq. (19), and $x_{i,j,j+d-1}$ consists of $x_{i,j}, x_{i,j+1}, \cdots, x_{i,j+d-1}$ with the same length as the convolution kernel, and each convolutional kernel pair can be obtained after convolutional operation to obtain a feature map V^l as shown in Eq. (19). (20) is shown.

$$v_i^l = f_{conv}\left(W_l \cdot x_{i,j,j+d-1} + b_l\right) \tag{19}$$

$$V^l = \left[v_1^l, v_2^l, \cdots, v_{n-d+1}^l\right] \tag{20}$$

where $f_{conv}(\cdot)$ is the ReLU function and b_t represents the bias of this convolution kernel.

Thus, the new features generated by a filter in the i rd sample can be expressed as:

$$v_i = \left[v_i^1, v_i^2, \cdots, v_i^{l_f}\right]^T \tag{21}$$

The feature map representation extracted after l_f filter is shown in equation (22):

$$V = \left[v_1, v_2, \cdots, v_{n-d+1}\right] \tag{22}$$

The new sequences obtained after two convolutional layers are denoted as V_{conv1} and V_{conv2} respectively and are concatenated to obtain sequence V_{conv}. The pooling layer is a pooling operation on sequence V_{conv}, and the vector p^l is obtained by performing pooling operation on the feature map V^l according to Eqn. (23), and further the pooled feature map is denoted as $P_{\max\ pooling} = \left[P_1, P_2, \cdots, P_{\left[(n-d+1)/2\right]}\right]$, where $P_i = \left[P_i^1, P_i^2, \cdots, P_i^{l_f}\right]$ is the vector of l_f feature maps that have been spliced together after pooling.

$$p^l = \max\ pooling\left(V^l\right) \tag{23}$$

where $maxpooling(\cdot)$ is the maximum pooling function for a one-dimensional input to the maximum pooling layer.

Then a convolutional layer with a step size of 2 is connected in series to achieve data dimensionality reduction, and then a parallel convolutional layer is connected in series to perform pooling and dimensionality reduction operations in the same way, and so the spatial feature extraction of the network traffic sequence is completed. Further, the pooled feature map sequence $P_{\max pooling}$ is input to the LSTM network, and on time step t, the LSTM performs temporal feature extraction according to Eq. pair P. The pooled feature sequence $P_{\max pooling}$ is processed by the LSTM for network traffic data, which are vectors T with both spatial and temporal characteristics, and n_{units} is the number of neurons $(0 \le u \le n_{units})$ set by the LSTM layer:

$$t^{\alpha} = \left[t^{\alpha}{}_1, t^{\alpha}{}_2, \cdots, t^{\alpha}{}_{[(n-d+1)/2]} \right] \tag{24}$$

$$t_i = \left[t_i^1, t_i^2, \cdots, t_i^M \right]^T \tag{25}$$

$$T = \left[t_1, t_2, \cdots, t_{[(n-d+1)/2]} \right] \tag{26}$$

where t^{α} is the feature obtained by the u nd neuron after computation: t_i is the new feature of all neurons after computation in the i th sample, and finally the spatio-temporal feature vector T is obtained by stacking the newly generated features of all neurons.

Then Batch Normalization layer is added to accelerate the convergence to prevent overfitting, and finally the probability of predicting category i is $\alpha_i (0 \le \alpha_i \le 1)$ by the output of the three-layer fully connected neural network, which is calculated as follows.

$$y = f_{dense}\left(W_{dense} T + b_{dense}\right) \tag{27}$$

$$\alpha_i = \frac{e^{v_i}}{\sum_1^k e^{y_i}}, i = 1, 2, \cdots, k \tag{28}$$

where $f_{dense}(\cdot)$ is the Sigmoid function, W_{dense} is the fully connected layer weights, b_{dense} is the bias, k denotes the number of sample categories in the dataset which is also the number of species predicted by the model, y is the output vector of the fully connected layer, and y_i is the i th dimension of the fully connected output features.

2.1.4. Method implementation. The steps to realize the proposed method are as follows:

Step 1: The NSL-KDD dataset is set as the network data feature matrix X and the symbolic features in it are transformed into numerical features using mapping method. The transformed training set and test set are obtained as X_{ntrain} and X_{ntest}, respectively.

Step 2: For the numerical training set X_{ntrain}, the number of samples of the minority category U2R is much less than that of the majority category Normal, so the principle is used to mark all the minority category samples with the corresponding states, and a certain number of minority category samples are generated according to Eq. to obtain the enhanced training set X'_{train}.

Step 3: According to the principle shown in Eq. Normalization operation is performed on the enhanced training set X'_{train} and test set X_{ntest} to obtain the training set X''_{train} and test set X'_{ntest} processed by the great warp method, respectively.

Step 4: The normalized training set X''_{train} and test set X'_{ntest} are two-dimensional data for the input of the neural network, and the reshape function is introduced to transform each feature into a data with one channel, and the transformed training set and test set are X_{train} and X_{test}, respectively.

Step 5: Considering that the network data has both temporal and spatial characteristics, the spatial and temporal features of the network data are extracted using a combination of CNN and LSTM, in which, for the spatial feature extraction part of the CNN, a parallel connection is used within the convolutional layers, and a series connection is used between the layers.

Step 6: Input the data-transformed training set X_{train} into the model constructed in step 5, optimize the model step by step by adjusting the hyperparameters and the network structure, and test the detection performance of the model on the test set X_{test}.

The proposed combined neural network anomaly detection method based on data enhancement, the flowchart of the implementation of the network anomaly detection method based on data enhancement is shown in Fig. 3.

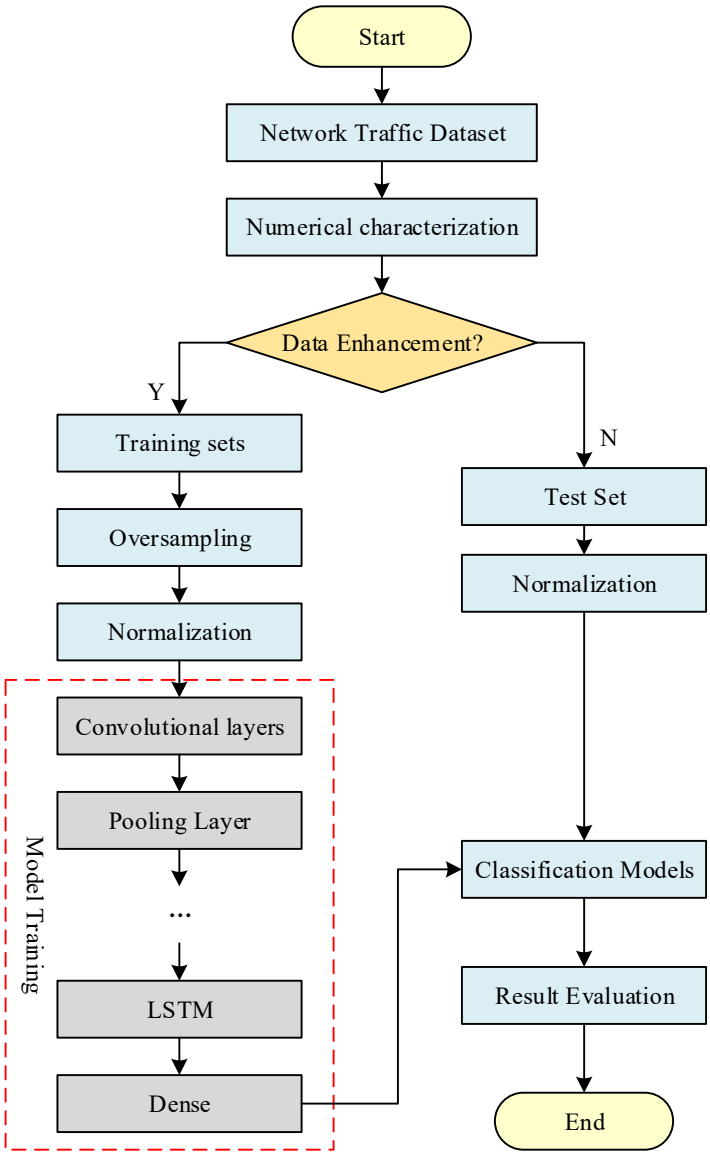

Fig. 3. Flowchart of network anomaly detection method based on data augmentation

2.2. Research on Network Information Security Threat Prediction Methods

2.2.1. Design of SA-GRU based posture prediction methodology:

1) Method Architecture Design. In recent years, cybersecurity attacks have become more and more diverse and random, which leads to the fact that cyber posture data are usually characterized by temporality and irregularity, bringing many new problems for cybersecurity posture prediction. Specifically, the temporal and irregular nature of posture data affects the accuracy and validity of network security posture prediction results. A network security posture prediction method based on SA-GRU is proposed. The method mainly consists of the SA-GRU method to characterize the posture data to learn and then realize the prediction. The proposed method in this chapter can better manage the relationship between the forgetting of past information and the degree of information retention at the current moment than the traditional GRU network.

2) Self-Attention structure. Self-Attention is mainly used to learn the interconnections that exist between input data. Unlike traditional Attention, Self-Attention is not an Attention mechanism between input statements and output statements, but an Attention mechanism that occurs between input statements or between output statements [24]. That is, the associations between inputs (or outputs) and inputs (or outputs) are automatically captured by the Attention mechanism with itself. Here, Attention mechanism denotes for the machine to learn to have the same focus of attention as human beings, intuitively Attention mechanism is similar to human's attention allocation process, that is, assigning different attentional weights to different contents during the information processing process.

The data a^i outputs the corresponding b^i results after Self-Attention learning, where the data inputs a^i and outputs b^i correspond to each other. Taking b^1 as an example, b^1 can be expressed as the information data after a^1 the data learns the feature information associated with other $a^2,\cdots,a^n$ data, i.e., b^1 it can filter out the useful information that has a strong correlation with it in this learning and then retain it, removing the useless information.

Here we take a^1 input Self-Attention calculation to get b^1 as an example to illustrate the calculation process, and the rest of $a^2,\cdots,a^n$ calculation method is the same as a^1. The specific calculation process of Self-Attention is as follows:

First, the initialized weight matrix is used to initialize the three vectors $q(query)$, $k(key)$ and $v(value)$ as follows:

$$q^i = W^q a^i \tag{29}$$

$$k^i = W^k a^i \tag{30}$$

$$v^i = W^v a^i \tag{31}$$

where, W^q, W^k, and W^v represent the three weight matrices of q, k, and v respectively, and then, q and each k are subjected to similarity computation to obtain weight $a_{1,i}$, and then weight $\hat{a}_{1,i}$ is obtained by softmax computation with the following formula:

$$a_{1,i} = q^1 \cdot \left(k^T\right)^i \tag{32}$$

$$a_{1,i} = \frac{a_{1,i}}{\sqrt{d}} \tag{33}$$

$$\hat{a}_{1,i} = soft\max\left(a_{1,i}\right) \tag{34}$$

where Eq. d is the dimension of q and k, where the calculation of $a_{1,i}$ needs to be divided by $\sqrt{d}$ in order to ensure stabilization of the gradient descent in the subsequent backpropagation. Finally the weights $\hat{a}_{1,i}$ are then computed with each v to obtain b^1.

$$b^1 = \sum_i \hat{a}_{1,i} v^i \tag{35}$$

3) Method design of SA-GRU. Using Self-Attention to effectively learn the characteristics of interconnection between data, we actively learn the association between the hidden state in the past moment and the current candidate set, so as to construct a new association matrix to highlight the important part of the candidate set that can express the current candidate information, remove the redundant part of the hidden state in the past that has nothing to do with the information in the candidate set, and ultimately supplement the information of the original candidate set, so as to realize the updating of the current hidden state information in the original GRU, and improve the defects of the update method of the current hidden state in the original GRU. Finally, the original candidate set information is supplemented to realize the updating of the current hidden state information, which improves the defects of the updating of the current hidden state in the original GRU, and proposes a network security posture prediction method based on SA-GRU to improve the posture prediction accuracy.

where R_t represents the reset gate in the GRU, X_t represents the input of the current moment, h_{t-1} represents the hidden state of the past moment, h_t represents the hidden state of the current moment, $\tilde{h}_t$ represents the candidate set in the GRU, h'_{t-1} represents the hidden information of the past moment after Self-Attention learning, W_R represents the weight of the reset gate, $W_{\tilde{h}}$ represents the weight of the candidate set, and W_O represents the weight of the final output, SA- The specific implementation process of GRU is as follows:

(1) Initialize the reset gate. The state of the reset gate is determined by the hidden state h_{t-1} from the previous node and the input X_t from the current node:

$$R_t = \sigma\left(W_R \cdot \left[h_{t-1}, X_t\right]\right) \tag{36}$$

(2) Construct the candidate set. The specific state $\tilde{h}_t$ on the current candidate set can be represented as:

$$\tilde{h}_t = \tanh\left(W_{\tilde{h}} \cdot \left[R_t \times h_{t-1}, x_t\right]\right) \tag{37}$$

(3) Construct Self-Attention input data. Splice h_{t-1} and $\tilde{h}_t$ using the cat function to prepare for SelfAttention learning.

$$h_{cat} = cat\left(\tilde{h}_t, h_{t-1}\right) \tag{38}$$

(4) Self-Attention learning. Use Self-Attention learning to learn the interrelationships between $\tilde{h}_t$ and h_{t-1}, and construct a new association matrix h'_{t-1} that contains the portion of information associated with h_{t-1} and $\tilde{h}_t$, and the portion of key features in $\tilde{h}_t$ that represent the candidate set.

$$h'_{t-1} = Self - Attention\left(h_{cat}\right) \tag{39}$$

The specific calculation process of Self-Attention here refers to the calculation process of Self-Attention, and it is worth noting that since only the supplementary information h'_{t-1} of candidate set

$\hat{h}_t$ needs to be constructed by Self-Attention in the method designed in this chapter, the value taken after Self-Attention learning is only taken as h'_{t-1}.

(5) Calculate the current hidden state. This step corresponds to the computation of h_t in GRU. In order to avoid the problem that the update gate in GRU cannot effectively control the forgetting of the past information and the degree of information retention in the current moment, this chapter directly sums the candidate set $\hat{h}_t$ and h'_{t-1} to realize the update of the current hidden state h_t. At this time, h_{t-1} contains the key feature part of candidate set $\hat{h}_t$ and the part of past hidden state information associated with the candidate set in h_{t-1}. Therefore, Equation (40) can realize highlighting the key information in the candidate set and removing the redundant information part in the past hidden state. The specific formula is as follows:

$$h_t = h'_{t-1} + \hat{h}_t \tag{40}$$

(6) Prediction result output. The final output of the forward propagation is as follows:

$$y_t = W_o \cdot h_t \tag{41}$$

2.2.2. SA-GRU based posture prediction method implementation. The flowchart of the SA-GRU based network security posture prediction method implementation is shown in Figure 4. The flow of SA-GRU-based network security posture prediction method is described as follows:

1) The posture data obtained through network posture assessment is divided into prediction training set and prediction test set in a certain ratio.

2) Input the prediction training set into the SA-GRU neural network to train the posture prediction model.

3) Determine whether the training of the model is completed or not, if it is not completed then continue training, if it is completed then proceed to the next step.

4) Input the prediction test set into the trained SA-GRU network-based security posture prediction model for prediction and output the corresponding predicted values.

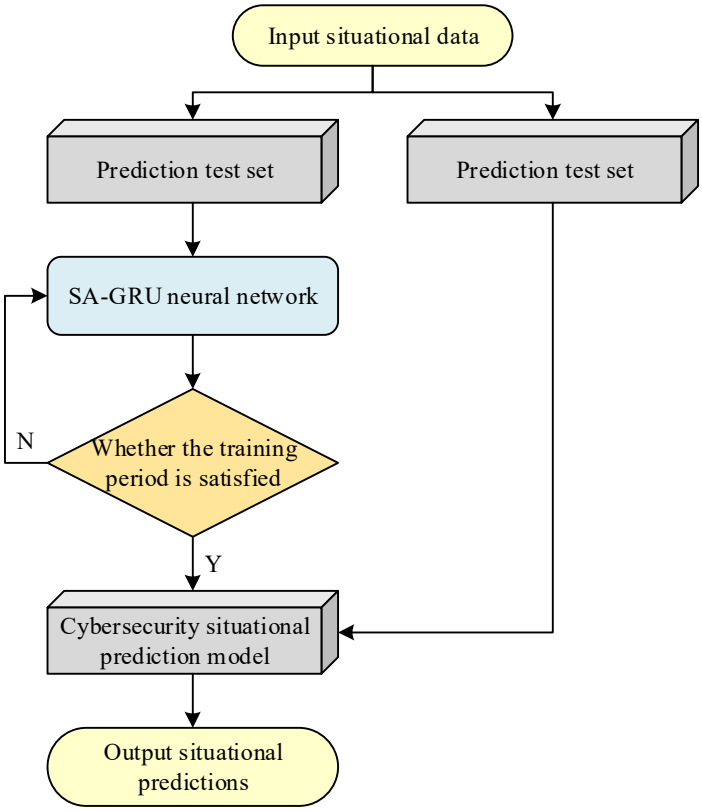

Fig. 4. The flow chart of the network safe state prediction based on SA-GRU

3. Results and discussion

3.1. Experiments related to model intrusion detection

3.1.1. Dichotomized experimental detection performance. In this section of the experimental analysis, this section focuses on performing the binary classification task on the CIC-IDS-2017 and CIC-DDoS-2019 datasets to explore the distinction between normal and attack data in network traffic. Considering that the amount of normal and attack traffic data in the binary classification task is comparable and there is no need to rely excessively on data enhancement techniques, this experiment decides not to use the improved ACGAN method for data enhancement.

Instead, this section chooses to use the TIBS (Transformer-Inception-BiGRU-SA) feature extraction model to directly detect and classify the preprocessed dataset. In order to evaluate the performance of the TIBS model and the effectiveness of its components in depth, a series of ablation experiments are designed in this section to reveal the advantages and uniqueness of TIBS in binary classification tasks by comparing the TIBS model with the IBS, TIB, TIS, and TBS models. These ablation experiments not only validate the importance of each component in the model, but also demonstrate the superior performance of the model in handling network intrusion detection tasks.

In addition, experimental comparisons of different learning rates are also conducted, aiming to find out the optimal learning rate setting to further enhance the detection efficiency and accuracy of the TIBS model. Through careful experimental analysis, this study demonstrates the potential application of the TIBS model in the field of network intrusion detection, proving its effectiveness and accuracy when dealing with balanced datasets. (The left figure is based on the CIC-IDS-2017 dataset and the right figure is based on the CIC-DDoS-2019 dataset.) The model training loss curve is shown in Figure 5. From the analysis of the training loss curves, the model proposed in this study performs much better in terms of both the stability of model training and the convergence speed. These results not only prove

the efficiency of the model, but also highlight its reliability in handling complex network intrusion detection tasks. Compared with other ablation models, the model in this paper requires fewer iterations and converges faster. This feature not only reduces training time and improves efficiency, but also means that the model in this paper is able to tune and optimize its parameters faster to achieve the desired level of performance.

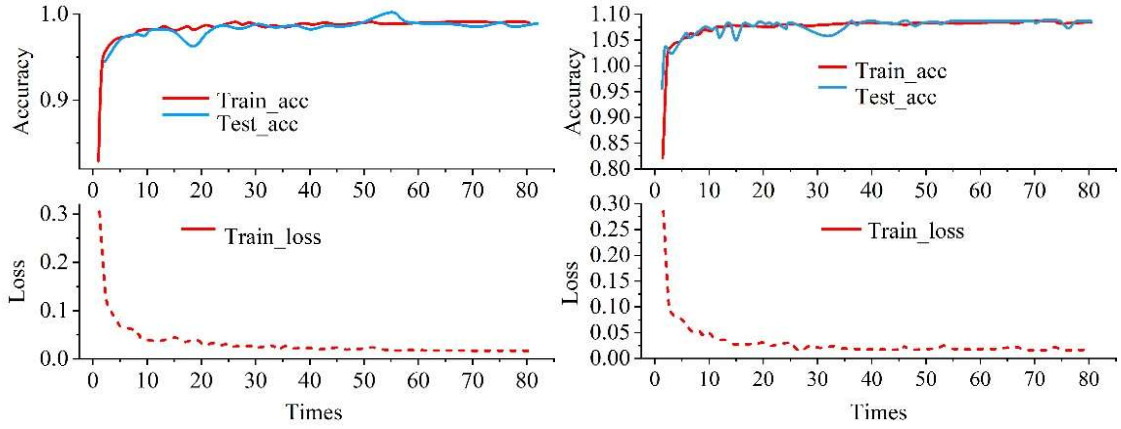

Fig. 5. Model training loss curve

Comparison of the results of the binary ablation experiments based on the CICIDS2017 and CICDDoS2019 datasets (%) As shown in Tables 1 and 2, the improved TIBS model, there is a substantial improvement in the performance indexes, in the dataset CIC-IDS-2017 the indexes can be reached to more than 99.7%, in the CIC-DDoS-2019 dataset indexes can reach more than 99.6%.

Table 1. Compared with the results of the cicids2017 data set of dichotomy

Model	ACC	P	R	F1-Score
IBS	89.72	89.99	90.11	89.91
TBS	91.28	91.25	91.26	91.78
TIS	92.12	91.69	92.49	92.06
TIB	90.72	91.43	91.07	91.08
TIBS	99.64	99.64	99.44	99.44

Table 2. Compared with the results of the CICDDoS2019 data set of dichotomy

Model	ACC	P	R	F1-Score
IBS	91.56	91.76	91.97	91.6
TBS	91.93	92.28	92.57	92.15
TIS	91.33	91.19	91.25	90.9
TIB	84.54	84.82	84.92	84.42
TIBS	99.52	99.63	99.94	99.73

The confusion matrices based on the CICIDS2017 and CICDDoS2019 datasets are shown in Figures 6 and 7. , as can be seen from the data in the figures, the detection rate for attack data is extremely high, providing unique and deep insights within the field of cybersecurity, especially in the evaluation of intrusion detection systems. The confusion matrices based on the CIC-IDS-2017 dataset reveal the accuracy and false alarm rate of the model in identifying various types of network intrusions, while Fig. 7 shows the performance of the model on the CIC-DDoS-2019 dataset. With these confusion matrices, researchers can visually assess the model's ability to recognize different types of attacks, thus providing a comprehensive analysis of the model's accuracy and reliability.

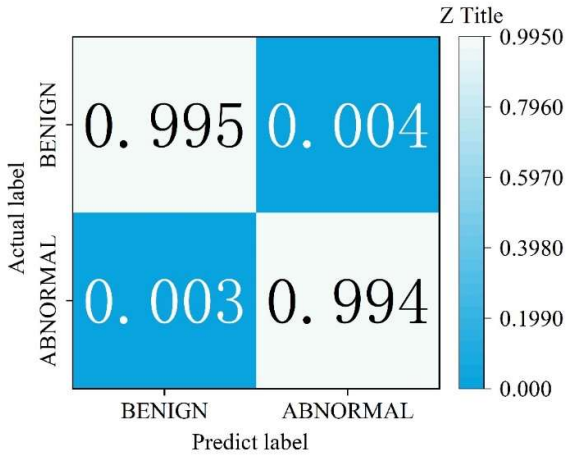

Fig. 6. Confusion matrix based on cicids2017 data set

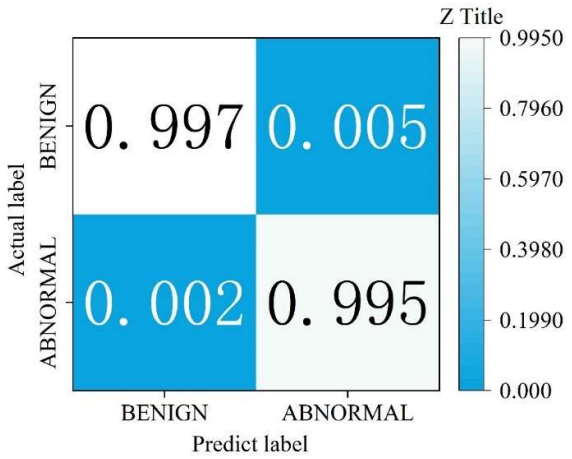

Fig. 7. The confusion matrix based on the cicddos2019 data set

The CICIDS2017 dataset learning rates of 0.01 and 0.001 are shown in Figures 8 and 9.The CICDDoS2019 dataset learning rates of 0.01 and 0.001 are shown in Figures 10 and 11. These curves provide intuition about the impact of learning rate settings on model training effectiveness, revealing how the size of the learning rate affects model convergence speed and stability. The curve changes demonstrate how the model performance changes when the learning rate is gradually reduced, helping researchers to find the best learning rate setting to optimize the model training process and achieve faster convergence speed and higher model accuracy. These graphs and comparative experiments not only provide valuable data and insights for research in the field of intrusion detection, but also emphasize the importance of careful consideration of different parameter settings during model design and training to ensure that the research can move forward efficiently, and at the same time improve the validity and reliability of the intrusion detection models in practical applications.

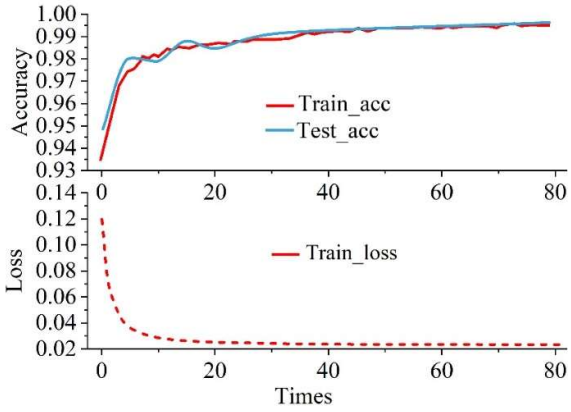

Fig. 8. Cicids2017 data set learning rate is 0.01

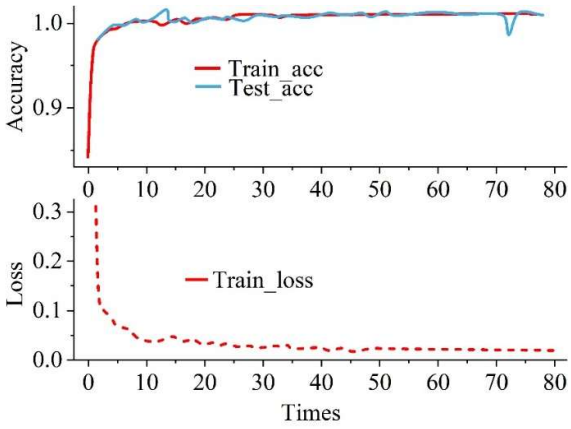

Fig. 9. Cicids2017 data set learning rate is 0.001

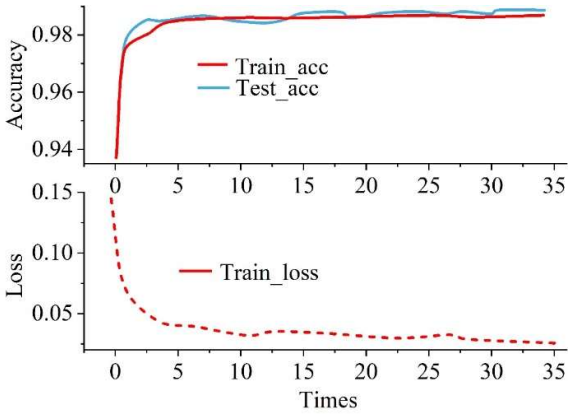

Fig. 10. The cicddos2019 data set learning rate is 0.01

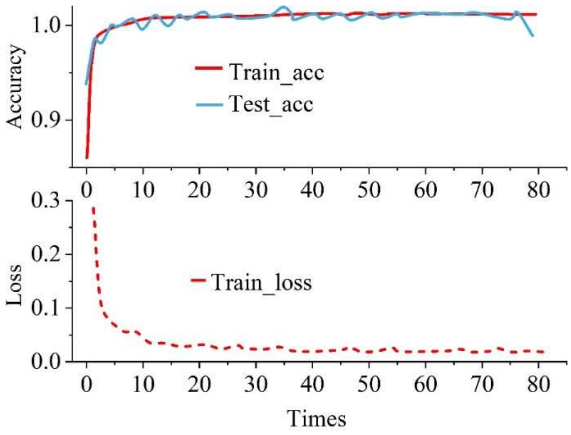

Fig. 11. The learning rate for cicddos2019 is 0.001

As can be seen from the above graphs, the training of the two datasets is more stable and faster when the learning rate is 0.001, and all the indexes are higher than other learning rates. Therefore, the learning rate of TIBS model in this study is set to 0.001. The comparison of performance results (%) with different learning rates is shown in Table 3.

Table 3. Comparison of performance results of different learning rates

Data set	Learning rate	ACC	P	R	F1-Score
CIC-IDS-2017	0.01	99.79	99.79	99.79	99.79
	0.001	99.54	99.54	99.54	99.54
CIC-DDoS-2019	0.01	99.39	99.39	99.39	99.39
	0.001	99.61	99.61	99.61	99.61

3.1.2. Multi-classification experimental detection performance. In this section of the experimental analysis, this section focuses on performing a ten-class task on the dataset CIC-IDS-2017 and a seven-classification task on the CIC-DDoS-2019 dataset, exploring the distinction between normal and various types of attack data in network traffic. This section trains the TIBS model based on the training set enhanced with the improved ACGAN model introduced in Chapter 3, and performs the validation analysis of the experiments in this section using the test set without data enhancement.

A series of ablation experiments are designed in this section to reveal the advantages and uniqueness of TIBS in multi-classification tasks by comparing it with models such as IBS, TIB, TIS, and TBS. These ablation experiments not only validate the importance of each component in the model, but also demonstrate the superior performance of the TIBS model in handling network intrusion detection tasks.

In addition, experimental comparisons of different learning rates were conducted, aiming to find out the optimal learning rate setting to further enhance the detection efficiency and accuracy of the TIBS model. Through careful experimental analysis, this study demonstrates the potential application of the TIBS model in the field of network intrusion detection, proving its effectiveness and accuracy when dealing with extremely unbalanced datasets, (the left figure is based on the CIC-IDS-2017 dataset, and the right figure is based on the CIC-DDoS-2019 dataset). The model training loss curve is shown in Figure 12. From the analysis of the training loss curves, the TIBS model proposed in this study demonstrates significant advantages over the other four ablation models examined. Specifically, the TIBS model shows better performance in terms of stability of model training and convergence speed. These results not only prove the efficiency of the TIBS model, but also highlight its reliability in dealing with complex network intrusion detection tasks.The stability of the training process exhibited by the TIBS model implies that the model is robust to the variability and noise in the training data. This stability is achieved through a well-designed model structure and optimization algorithm, which ensures a smooth decrease of the loss function throughout the training process, thus avoiding large fluctuations that may occur during the training process. Compared to other ablation models, the TIBS model converges faster and the model training is stable, a property that not only reduces training time and improves efficiency, but also means that the model is able to adjust and optimize its parameters faster to achieve the desired level of performance.

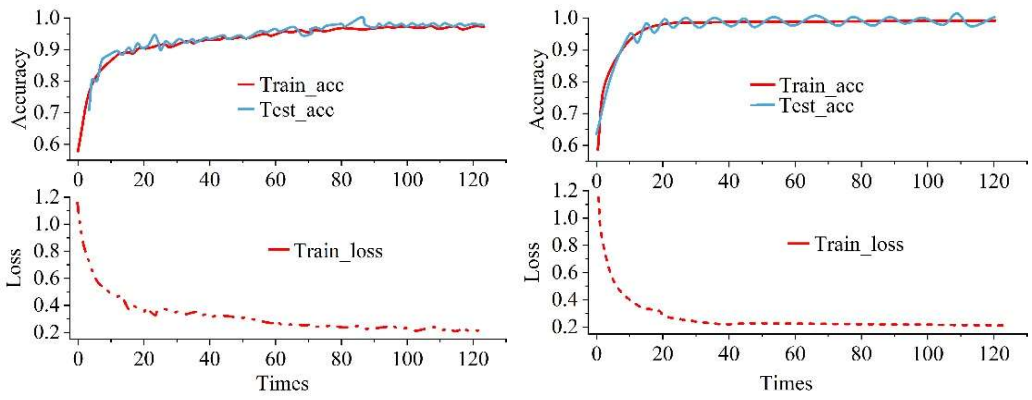

Fig. 12. Model training loss curve

The comparison of the results (%) of the multiclassification ablation experiments based on the CICIDS2017 and CICDDoS2019 datasets are shown in Tables 4 and 5. As can be seen from the table, the improved model, there is a substantial improvement in various performance indicators, the accuracy can reach 94.9% in the dataset CIC-IDS-2017, and 96.94% in the CIC-DDoS-2019 dataset, which also incorporates the performance comparison with the SOTA model, in the CIC-IDS-2017 dataset, the current best-performing model is the CNN-BiLSTM model, and in the CIC-DDoS-2019 dataset, the current best performing model is the DDoSNet model.

Table 4. Compared with the results of the cicids2017 data set polypartial ablation experiment

Model	ACC	P	R	F1-Score
DT	87.41	81.21	80.15	--
RNN	83.99	81.45	82.51	82.85
BiLSTM	--	80.34	85.48	82.21
RF-DAGMM	75.03	71.49	42.03	52.09
Transformer-CNN	91.04	--	--	--
H E Fed(CNN)	92.79	--	--	--
BiGAN	81.78	76.45	76.9	76.11
SGAN	89.27	87.6	71.99	--
CNN-BiLSTM	--	95.7	85.72	91.19
Ours	94.9	92.31	77.97	83.62

Table 5. Compared with the results of the CICDDoS2019 data set polypartial ablation experiment

Model	ACC	P	R	F1-Score
Convolutuinal AE[	85.15	87.89	85.22	85.39
ATFT	92.14	92.09	92.82	--
CNN-BiLSTM	94.18	94.67	91.51	93.37
Can	--	91.14	79.51	78.88
DDoSNet	99.37	99.37	99.37	99.37
Ours	96.94	95.89	86.47	89.85

The CICIDS2017 dataset learning rates of 0.01 and 0.001 are shown in Figures 13 and 14.The CICDDoS2019 dataset learning rates of 0.01 and 0.001 are shown in Figures 15 and 16.

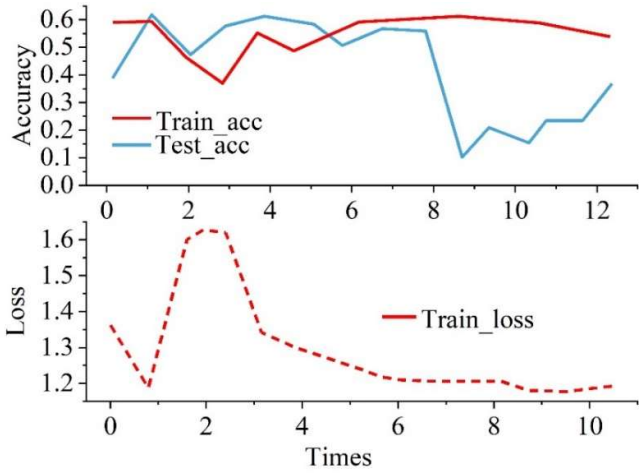

Fig. 13. Cicids2017 data set learning rate is 0.01

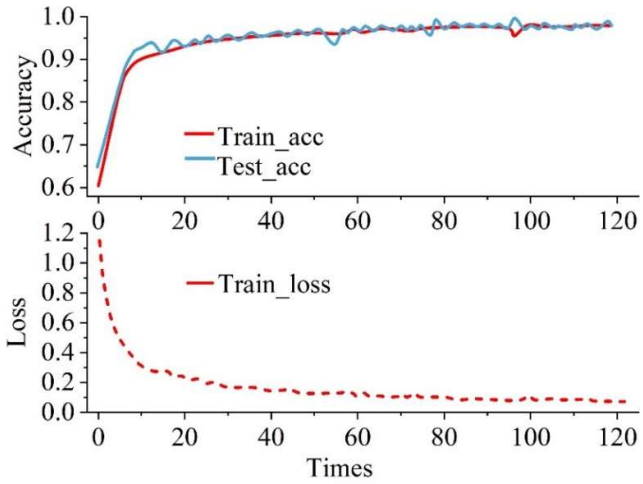

Fig. 14. Cicids2017 data set learning rate is 0.001

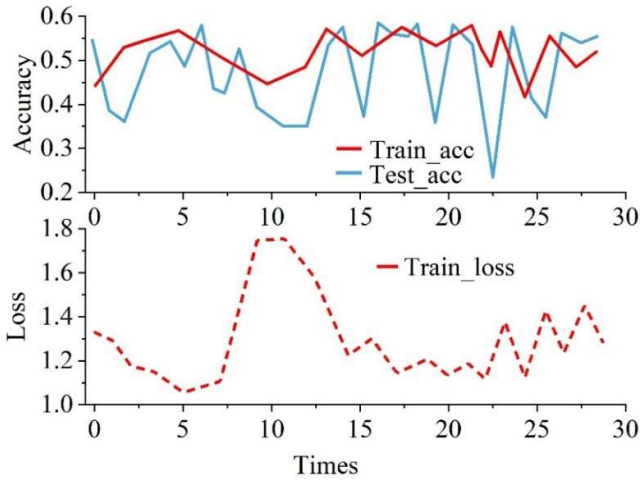

Fig. 15. The cicddos2019 data set learning rate is 0.01

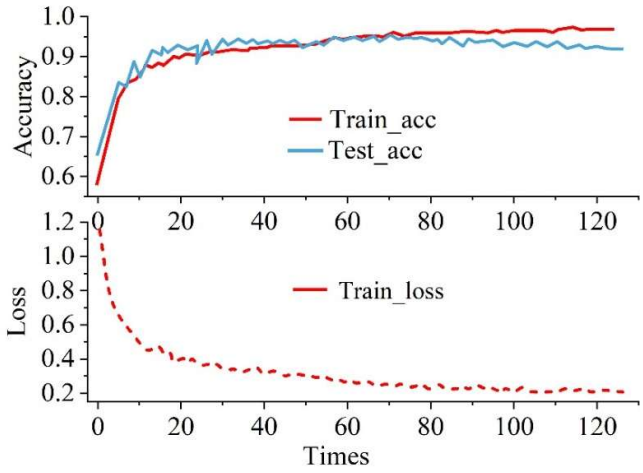

Fig. 16. The learning rate for cicddos2019 is 0.001

As can be seen from the above graphs, the training of the two datasets is more stable and faster when the learning rate is 0.001, and all the indexes are higher than other learning rates. Therefore, the learning rate of TIBS model in this study is set to 0.001. The comparison of performance results (%) with different learning rates is shown in Table 6.

Table 6. Comparison of performance results of different learning rates

Data set	Learning rate	ACC	P	R	F1-Score
CIC-IDS-2017	0.01	61.28	28.69	24.64	23.09
	0.001	95.2	92.78	78.35	82.83
CIC-DDoS-2019	0.01	59.82	25.25	22.27	20.5
	0.001	97.04	95.54	86.66	90.15

To summarize the above experiments: for binary classification experiments, TIBS can effectively separate normal data traffic and attack data traffic, and effectively improve the performance of intrusion detection. For multi-classification experiments, the data distribution is not balanced in the dataset, the data enhancement of ACGAN model can be improved to effectively improve the detection rate of the model, which is more helpful for the identification of a few attack data. And the model proposed in this paper has a large improvement compared with other improved models released in recent years, which shows the advanced and innovative nature of the model.

3.2. Experiments and discussions

3.2.1. Experimental environment. Hardware environment of this experiment: the server is Dell PowerEdge R730, CPU is Inter(R) Core(TM) i5-8500, RAM is 8G, operating system is CentOS 7, database is mySQL 5, client PC is ThinkPad X260, CPU is Intel Core i5 6200U, RAM is 4G, hard disk is 500G, and the operating system is Windows 10. The training and testing of the model was done in python 3.7 environment using Keras deep learning framework. The experimental environment parameters are shown in Table 7.

Table 7. Experimental environment parameters

Environment	Parameter
Server	Dell PowerEdge R730
Client Side	ThinkPad X260
Operating System	CentOS 7
Database	mySQL 5
Processor	Inter(R) Core(TM) i5-8500
Memory	64 GB DDR3 1 333 MHz
Model Depth Learning Training And Testing	python3.7

3.2.2. Experimental data. The experimental data in this section is selected from the Internet network security posture database released by an Internet emergency response center as the basis for the experiment. The scope of network security monitoring includes computers, websites, information security vulnerability sharing platforms, network devices and applications in the territory. The data model includes relevant records such as host data infected with network viruses, data of tampered websites, data of security vulnerabilities, and data of processed network security events. The data dimensions of this experiment include harmful programs, website security, security vulnerabilities, and security events, etc., and the attributes of data features measured by each data dimension are different, and the experimental data features are shown in Table 8. The data dimensions of this experiment include harmful programs, website security, security vulnerabilities, security events, etc., and the data feature attributes measured by each data dimension are not the same. Among them, the data characteristics measured by harmful programs include the number and increment of hosts infected with network viruses, the number and increment of hosts controlled by Trojan horses or zombie programs, the number and increment of hosts infected with flyer worms in the country, the number and increment of domain names involved in the site of the release, the number and increment of foreign domain names involved in the site of the release, the number and increment of domain names with the top-level domain of the site of the release of the release of the release of the release of the release of . The number and increment of IP addresses involved in the deflowered site, the number and increment of foreign IP addresses involved in the deflowered site, and 15 other features. The data characteristics of website security measurement include the number and increment of tampered non-government websites, the number and increment of tampered government websites, the number and increment of non-government websites with implanted backdoors, the number and increment of government websites with implanted backdoors, the number and increment of domain names involved in the counterfeit pages of websites, and 12 other characteristics. The data characteristics of security vulnerability measurement include the number and increment of vulnerabilities in security products, the number and increment of vulnerabilities in operating systems, the number and increment of vulnerabilities in Web applications, the number and increment of vulnerabilities in network devices, the number and increment of vulnerabilities in applications, the number and increment of vulnerabilities in databases, and other 13 characteristics. The data features of security event measurement include 35 features such as the number and increment of cybersecurity event occurrences such as web page imitation, denial-of-service attack, domain name anomaly, website backdoor, malicious code, malicious program, web page tampering, vulnerability, malicious probe scanning, spamming, mobile Internet malware, web page hanging, unauthorized access, mobile APP imitation, investigation and forensics, information leakage, and other cybersecurity event occurrences and increment.

The experimental samples are the consecutive weekly observations from October 30, 2020 to October 18, 2024, after smoothing the original posture dataset and obtaining the data of consecutive time periods after noise processing, the training set selected in this paper is the data of the first 176 weeks, the test set is the data of the last 32 weeks, and the prediction duration is one week.

Table 8. Experimental data characteristics

Data dimension	Data characteristics
Harmful procedure	15
Website security	12
Safety loophole	13
Safety event	35

3.2.3. Parameter setting. The experimental data is taken as hesitancy threshold of 0.2 and the hesitancy, intuitionistic fuzzy affiliation and non-affiliation are calculated for the sample set of big data security features.The parameters of the LM-PSO optimization algorithm are chosen as the population size of 5, the number of evolutions of 20, the dimensionality of each particle of 3, ε =10^{-6}, β =10, and μ_0 =0.001.The parameter settings are shown in Table 9.

Table 9. Parameter setting

Parameter name	Parameter value
GRU units	65
Fully connect units	30
Batch size	30
Dropout	0.8
Learning rate	0.001
Time step	5

3.2.4. Experiments and discussions. In this section, the cybersecurity big data feature dataset and the corresponding cybersecurity posture value sample dataset are used as inputs, and the cybersecurity posture value data predicted by this paper's method in the latter 32 weeks are compared with the results of the prediction using the NARX, single-layer GRU, RNN, CNN, two-layer GRU, and Att-LSTM models, and the overall accuracy and test time of different prediction models are shown in Table 10. From the comparison of the experimental results, it can be seen that the prediction accuracy of the method in this paper is 99.28%, which is significantly higher than other prediction models and methods. The prediction effect of the two-layer stacked GRU network is also better than that of the single-layer network, and the prediction accuracy of the GRU network incorporating two-layer timing features is 2.29% higher than that of the single-layer GRU network. Compared with Att-LSTM, CNN, and RNN studied in the past two years, the prediction accuracy of the model in this paper is improved by 1.23%, 1.87%, and 2.97%, respectively, and by 7.25% compared with the traditional NARX prediction models and methods. In addition, the experimental results also show that the model in this paper also needs more training time, but the prediction time of the model tested after it is trained is also faster than other deep learning networks such as Att-LSTM, CNN, and RNN.

Table 10. Accuracy and test time of network security situation prediction model

Method	Accuracy rate(%)	Model training time(S)	Test time(S)
NARX	92.52%	3.35	0.61
GRU(Single layer)	94.79%	3.5	0.56
RNN	96.31%	7.74	1.44
CNN	97.41%	9	1.8
GRU(Double layer)	97.08%	6.19	0.88
Att-LSTM	98.05%	10.36	1.53
Ours	99.28%	12.99	1.09

Comparison of performance indicators of network security posture values predicted using different prediction models is shown in Table 11. As can be seen from the comparison of the experimental results: in this experiment, the results of this paper's model prediction of the average absolute error of the training data is 0.00266, the average absolute error of the test data is 0.00369, and the absolute error is controlled within 0.003, the root mean square error of the training data is 0.01151, and the root mean square error of the test data is 0.00551, and it can be seen that the prediction accuracy of this paper's model prediction is is significantly higher than that of other Att-LSTM, CNN, RNN deep learning methods, and the prediction accuracy of the GRU network with the introduction of the attention mechanism is significantly better than that of the two-layer GRU network without the introduction of the attention mechanism. The average absolute error of the training data, the average absolute error of the test data, the root mean square error of the training data and the root mean square error of the test data of the two-layer GRU network without the introduction of the attention mechanism are 0.00481, 0.00253, 0.01665 and 0.01494, respectively, while the average absolute error of the training data, the average absolute error of the test data and the root mean square error of the training data of the one-layer GRU network are 0.01233, respectively. error are 0.01231, 0.02549 and 0.02725, respectively, and the root mean square error of test data is 0.04508, which shows that the prediction accuracy of the two-layer stacked GRU network is better than that of the single-layer GRU network. Compared with other prediction methods, the traditional NARX prediction method is difficult to recognize the long-term dependence between time series, and its prediction results have the lowest accuracy.

Table 11. Comparison of performance indexes of network security situation values predicted

Method	MAE		RMSE	
	Training(85%)	Testing(15%)	Training(85%)	Testing(15%)
NARX	0.02041	0.04778	0.03869	0.09219
GRU(Single layer)	0.01231	0.02549	0.02725	0.04508
RNN	0.00715	0.01256	0.0249	0.04161
CNN	0.00525	0.00272	0.01342	0.01588
GRU(Double layer)	0.00481	0.00253	0.01665	0.01494
Att-LSTM	0.00509	0.00243	0.01248	0.0057
Ours	0.00266	0.00369	0.01151	0.00551

The comparison of the results of the predicted values of cyber security posture for different prediction models is shown in Figure 17. The figure shows the comparison between the results of the prediction using NARX, single-layer GRU, RNN, CNN, two-layer GRU, Att-LSTM models and the predicted values of the knots predicted by the models in this paper. It can be seen that the cybersecurity posture prediction model based on the basic text model can accurately predict the

cybersecurity posture, the predicted value has the highest fit with the actual value, and the predicted trend and the actual trend highly overlap.

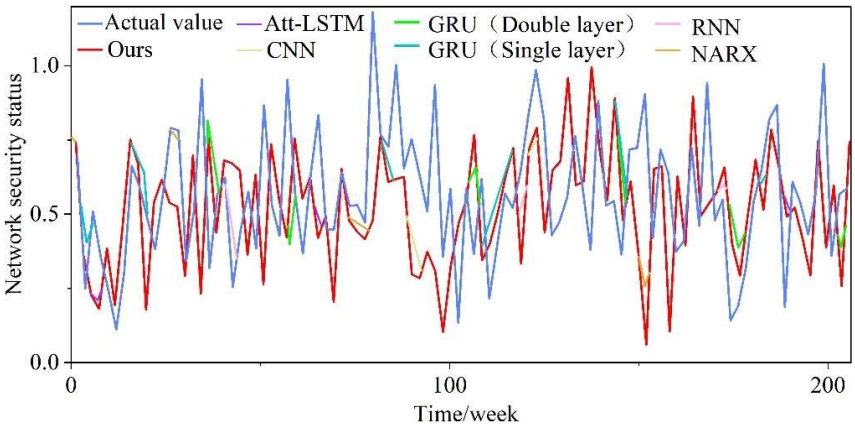

Fig. 17. Comparison of network security situation prediction results

3.3. Network information security defense mechanism design

3.3.1. Data encryption. Although the development of wireless networks and its unique capacity and flexibility for people's work and life to provide great convenience, but people are still highly concerned about information security issues. In the wireless transmission network connection, sensor node work, subsequent data transmission of three aspects there will be various types of problems, so sound and perfect information security defense mechanism is to ensure that the user uses the wireless sensor network of the privacy of the key way. The so-called data encryption is information encryption technology, that is, encrypted processing of information transmission of all kinds of information, to ensure the reliability of information transmission, accuracy and secrecy. However, it should be pointed out that because the sensor itself covers low energy, its carrying capacity is relatively weak, and it is necessary to choose a scientific and reasonable encryption algorithm. Under the premise of fully considering the actual situation, optimizing the encryption algorithm can reduce the pressure of the sensor node to a certain extent, so the encryption algorithm is usually selected to implement encryption processing of data. In addition, data encryption technology often appears in the campus network, the reason for this is mostly related to the complexity of the campus wireless network path, sensor nodes, unknown environment, etc., undoubtedly is the best area for hackers and viruses to invade, so in the campus wireless sensor network, data encryption security measures can be taken to maximize the security of the campus network.

3.3.2. Utilizing keys. A secret key is also known as a secret key, and since it is called a “secret”, it cannot be known by most people in an area that utilizes a wireless network. A key is an important way of unlocking data and transmitting files, and it is also known as a password. Currently in various fields of work and people's real life, due to the importance of the transmitted data, data users at both ends of the transmission will use key technology to ensure the accuracy and privacy of the data. Among them, the key technology is to protect the network information security, according to the different levels of importance of the data, the key technology is also different. The use of the same key to decrypt data files can be called symmetric key, this type of key for the data of both ends of the user's clear existence, simple, convenient and fast private. In addition to the secret key type, there are also public keys with private keys, and users with public keys cannot decrypt data files in private keys. The above key technology has strong confidentiality and security properties, maximizing the phenomenon of interception, misappropriation or theft in file transmission. However, this kind of key technology

requires very high algorithms, especially in the process of running a high occupancy rate of resources, if the transmitted file is not very secret, usually in real life will not use this key technology.

3.3.3. Intrusion detection. Usually, when information security incidents occur due to force majeure factors, the problem can often be solved through human maintenance and management. However, some of the information security problems are related to data misappropriation or human-induced information leakage. For the above situation, the need to do wireless sensor network intrusion detection work, in order to prevent data encroachment and information leakage. General intrusion detection in the inspection path can be divided into the following two categories, one of the distributed intrusion detection, the most notable feature of the inspection method that is relatively independent, that is, the inspection method detector in addition to set up in a host or the mother network, but by setting up detectors in the parallel sub-network and a number of hosts to intertwine the formation of the presence of the array. Each detector has its own jurisdiction, in its own jurisdiction can be accurate and rapid implementation of intrusion detection, not managed by other detectors. The above intrusion detection can effectively improve the rate of detection in the region, each jurisdiction through information interaction to achieve interoperability. Everything has its advantages and disadvantages, the intrusion detection technology drawbacks due to the jurisdiction of independent, interactive information due to human damage and can not play a due role, each jurisdiction has to spend more energy to detect the same suspected invasion or similar danger, invariably reduces the efficiency of the investigation. In addition to the above inspection methods, hierarchical inspection is also an invasion of inspection methods, each level has its own important task, in the form of task interaction and the upper and lower levels to get in touch, so this invasion of inspection methods can also be referred to as tree-type inspection. The inspection method, the bottom detector is responsible for the collection of data, and then pass the data information to the upper level or higher levels, to maximize the simplification of the inspection form, improve inspection efficiency.

4. Conclusion

Network information security threat prediction and defense mechanism as a key technology to improve network security capability has received extensive attention from researchers. Through experimental research, this paper draws the following conclusions:

In the binary classification ablation experiments, it is found that the model in this paper has a substantial improvement in various performance indicators, and the indicators reach more than 99.7% and 99.6% on the datasets CIC-IDS-2017 and CIC-DDoS-2019, respectively. After experimental testing of the overall accuracy and test time of different prediction models, the method in this paper is significantly higher than other prediction models and methods, with a prediction accuracy of 99.28%. Compared with Att-LSTM, CNN, and RNN, the prediction accuracy of this paper's model is improved by 1.23%, 1.87%, and 2.97%, respectively, thus verifying the effectiveness of this paper's model.

In summary, the design of network information security threat prediction and defense mechanism proposed in this paper improves the real-time and accuracy of network security threat prediction, and provides new methods and means for network security prediction in big data environment.

About the Author

Shan Lu, Female, professional and technical first-class police inspector, born in August 1973. Specialized in criminal intelligence. She has participated in and presided over a number of provincial and ministerial level courses. She has traveled to Thailand, Australia, Hong Kong and other countries and regions to participate in criminal intelligence programs, and has repeatedly provided training on criminal intelligence analysis for combat units and foreign police.

References

[1] Husák, M., Komárková, J., Bou-Harb, E., & Čeleda, P. (2018). Survey of attack projection, prediction, and forecasting in cyber security. *IEEE Communications Surveys & Tutorials*, 21(1), 640-660.

[2] Hu, H., Zhang, H., Liu, Y., & Wang, Y. (2017). Quantitative method for network security situation based on attack prediction. *Security and Communication Networks*, 2017(1), 3407642.

[3] Maddireddy, B. R., & Maddireddy, B. R. (2020). Proactive Cyber Defense: Utilizing AI for Early Threat Detection and Risk Assessment. *International Journal of Advanced Engineering Technologies and Innovations*, 1(2), 64-83.

[4] Benzaïd, C., & Taleb, T. (2020). AI for beyond 5G networks: A cyber-security defense or offense enabler?. *IEEE network*, 34(6), 140-147.

[5] Wheelus, C., & Zhu, X. (2020). IoT network security: Threats, risks, and a data-driven defense framework. *IoT*, 1(2), 259-285.

[6] Li, J., Wu, Y., Li, Y., Zhang, Z., Fouad, H., & Altameem, T. (2023). A Network Security Prediction Method Based on Attack Defense Tree. *Journal of Nanoelectronics and Optoelectronics*, 18(3), 357-366.

[7] Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., & Zhang, J. (2023). Cyber threat intelligence mining for proactive cybersecurity defense: a survey and new perspectives. *IEEE Communications Surveys & Tutorials*, 25(3), 1748-1774.

[8] Duary, S., Choudhury, P., Mishra, S., Sharma, V., Rao, D. D., & Aderemi, A. P. (2024, February). Cybersecurity threats detection in intelligent networks using predictive analytics approaches. *In 2024 4th International Conference on Innovative Practices in Technology and Management* (ICIPTM) (pp. 1-5). IEEE.

[9] Xie, L., Hang, F., Guo, W., Lv, Y., Ou, W., & Shibly, F. H. A. (2021). Network security defence system based on artificial intelligence and big data technology. *International journal of high performance systems architecture*, 10(3-4), 140-151.

[10] Macas, M., Wu, C., & Fuertes, W. (2022). A survey on deep learning for cybersecurity: Progress, challenges, and opportunities. *Computer Networks*, 212, 109032.

[11] Ullah, F., Naeem, H., Jabbar, S., Khalid, S., Latif, M. A., Al-Turjman, F., & Mostarda, L. (2019). Cyber security threats detection in internet of things using deep learning approach. *IEEE access*, 7, 124379-124389.

[12] Tariq, M. I., Memon, N. A., Ahmed, S., Tayyaba, S., Mushtaq, M. T., Mian, N. A., ... & Ashraf, M. W. (2020). A review of deep learning security and privacy defensive techniques. *Mobile Information Systems*, 2020(1), 6535834.

[13] Sarker, I. H. (2021). Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective. *SN Computer Science*, 2(3), 154.

[14] Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). Machine learning and deep learning approaches for cybersecurity: A review. *IEEE Access*, 10, 19572-19585.

[15] Dixit, P., & Silakari, S. (2021). Deep learning algorithms for cybersecurity applications: A technological and status review. *Computer Science Review*, 39, 100317.

[16] Ahsan, M., Nygard, K. E., Gomes, R., Chowdhury, M. M., Rifat, N., & Connolly, J. F. (2022). Cybersecurity threats and their mitigation approaches using Machine Learning—A Review. *Journal of Cybersecurity and Privacy*, 2(3), 527-555.

[17] Sokolov, S. A., Iliev, T. B., & Stoyanov, I. S. (2019, May). Analysis of cybersecurity threats in cloud applications using deep learning techniques. *In 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics* (MIPRO) (pp. 441-446). IEEE.

[18] Berman, D. S., Buczak, A. L., Chavis, J. S., & Corbett, C. L. (2019). A survey of deep learning methods for cyber security. *Information*, 10(4), 122.

[19] Salloum, S. A., Alshurideh, M., Elnagar, A., & Shaalan, K. (2020, March). Machine learning and deep learning techniques for cybersecurity: a review. *In The International Conference on Artificial Intelligence and Computer Vision* (pp. 50-57). Cham: Springer International Publishing.

[20] Nassar, A., & Kamal, M. (2021). Machine Learning and Big Data analytics for Cybersecurity Threat Detection: A Holistic review of techniques and case studies. *Journal of Artificial Intelligence and Machine Learning in Management*, 5(1), 51-63.

[21] Seethalakshmi Perumal,P. Kola Sujatha,Krishnaa S. & Muralitharan Krishnan. (2025). Clusters in chaos: A deep unsupervised learning paradigm for network anomaly detection. Journal of Network and Computer Applications104083-104083.

[22] Haijun Geng,Qi Ma,Haotian Chi,Zhi Zhang,Jing Yang & Xia Yin. (2025). DUdetector: A dual-granularity unsupervised model for network anomaly detection. Computer Networks110937-110937.

[23] Yuan Feng,Hongying Zhao,Jianwei Zhang,Zengyu Cai,Liang Zhu & Ran Zhang. (2024). Prediction of Network Security Situation Based on Attention Mechanism and Convolutional Neural Network–Gated Recurrent Unit. Applied Sciences(15),6652-6652.

[24] Sun Junfeng,Li Chenghai & Song Yafei. (2024). A network security situation prediction approach based on MAML and BiGRU. Journal of Intelligent & Fuzzy Systems(3-4),307-319.