Network Security Based on GCN and Multi-Layer Perception

Wei Yu*, Huitong Liu, Yu Song, Jiaming Wang
Guangzhou Bureau, EHV Power Transmission Company of China, Southern Power Grid, Guangzhou, 510663, China

***Abstract*—With the continuous progress of network technology, network security has become a critical issue at present. There are already many network security intrusion detection models, but these detection models still have problems such as low detection accuracy and long interception time of intrusion information. To address these drawbacks, this study utilizes graph convolutional network to optimize multi-layer perceptron. An optimization algorithm based on multi-layer perceptron is innovatively proposed to construct an intrusion detection model. Comparative experiments are conducted on the improved algorithm. The accuracy of the algorithm was 0.98, the F1 value was 0.97, and the detection time was 1.1s. The overall performance was much better than comparison algorithms. Subsequently, the intrusion detection model was applied to network security detection. The detection time was 0.1s, the accuracy was 0.98, and the overall performance outperformed other comparison algorithms. The results demonstrate that the intrusion detection method on the basis of optimized multi-layer perceptron can enhance the detection ability of illegal intrusion information. This study optimizes the performance of detecting illegal network intrusion information, providing a theoretical basis for further development of network security. However, the types of intrusion information in this study are limited and there is still uncertainty. In the future, data augmentation techniques can be used to oversample minority class samples, synthesize new minority class samples, expand sample size, increase detection information, and improve the overall detection performance of the model.**

***Keywords*—Network security; graph convolutional network; multi-layer perceptron; intrusion detection model**

I. Introduction

In the current era of rapid digital development, network security is becoming increasingly prominent, which is an important challenge that countries, enterprises, and individuals must face [1]. Affected by the popularity of information technology and the Internet, network attacks are constantly evolving, and the traditional security measures have been difficult to deal with. Therefore, exploring new methods for network security protection is of great significance [2]. Many scholars have conducted research on network intrusion detection models. For example, Fu et al. proposed an intrusion detection model based on attention mechanism to enhance the performance of traditional network firewalls and data encryption methods. Through experimental verification, the model achieved a detection accuracy of 90.73% [3]. In addition, Hnamte et al. designed a network intrusion detection model based on deep neural networks for network attacks. Then, the model was applied to detect in practical situations. The results showed that the model could detect most of the intrusion information in the network [4]. In recent years, Graph Convolutional Network (GCN) has shown strong feature extraction and relationship learning capabilities in multiple fields. Especially when dealing with non-Euclidean structured data, it has significant advantages [5]. Therefore, multiple scholars have applied it to network security protection. Diao et al. developed a spatiotemporal multi-scale GCN security model to improve the security of network data in vehicle prediction. After using this network security protection model, the security of network data in vehicle prediction was significantly improved [6]. To optimize the intrusion detection performance of labeled IoT networks, Deng et al. developed a GCN on the basis of flow topology. Comparative experiments on this model demonstrated that the intrusion detection accuracy of the GCN based on flow topology for labeled IoT networks was 92.31%, significantly better than other traditional methods [7]. Afterwards, Al-Ibraheemi et al. built an intrusion detection method on the basis of GCN and deep reinforcement learning algorithms to response the insufficient performance of intrusion detection models in software defined networks. The accuracy of this intrusion detection model was enhanced by 15.32% than the traditional intrusion detection model [8].

Meanwhile, Multi-layer Perceptron (MLP), as a classic deep learning model, also performs well in dealing with linearly inseparable problems [9]. Therefore, many scholars have also applied it to network security protection. Specifically, Shewale et al. designed an intrusion detection approach on the basis of MLP and Long Short-Term Memory Network (LSTM) to improve the network security. Comparative experiments showed that the intrusion detection accuracy was 91.83%, significantly better than traditional models [10]. In addition, to address the difficulty of detecting distributed denial of service attacks, Najar et al. designed a hybrid algorithm based on MLP and random forest. Comparative experiments were conducted on a distributed denial of service attack dataset. It was found that the detection accuracy was 93.85% [11].

The above research indicates that in the field of network security, although some research have attempted to apply advanced technologies such as GCN and MLP, there are still some drawbacks. At present, the research mainly focuses on using machine learning frameworks for network intrusion detection and abnormal behavior recognition, but these methods ignore complex relationships between network data, resulting in limited detection accuracy and efficiency. In addition, existing research lacks sufficient flexibility and adaptability in dealing with constantly changing network threats. Therefore, this study designs a network security protection method on the basis of GCN and MLP. This method aims to combine the powerful relationship learning ability of

GCN with the nonlinear processing ability of MLP to process complex network data. At the same time, the GCN algorithm is used to optimize the initial parameters in MLP, improve its flexibility and generalization ability, reduce the impact of complex data on detection results in previous intrusion detection models, and more effectively identify and defend against network attacks. The innovation of the research lies in the organic combination of GCN and MLP, forming a new network security protection framework. This framework can not only handle complex network relational data, but also adaptively learn and respond to constantly changing network threats. It is expected to provide a new and more effective technological means for network security, contributing to building a more secure and reliable network environment. The contribution of this study is to timely detect abnormal information in the network through the GCN-MLP intrusion detection model, timely identify potential security threats, and reduce the damage caused by network attacks. This model promptly prevents malicious attackers from invading, protects the secure operation of networks or systems, and ensures the integrity and confidentiality of data information in the network. This model ensures that users or processes use system resources according to prescribed permissions, preventing resources from being illegally occupied.

The article is divided into five sections for discussion. Section II mainly covers network security related content and research on MLP and GCN algorithms. Section III construct an network intrusion detection model based on GCN and MLP algorithms. Section IV analyzes the effectiveness of the proposed intrusion detection model. Section V summarizes the entire text.

II. Methods and Materials

A. Multi-Layer Perceptron Optimization Integrating Graph Convolutional Network

At present, people are paying more attention to network security issues, and there are also more network information intrusion detection models. However, these models still have problems such as false positives and missed detection [12]. MLP is a deep learning algorithm based on feedforward neural networks, which is composed of multiple neural structures. This algorithm has strong representation and generalization capabilities, which can process various complex data, reducing the false detection rate of dangerous intrusion detection [13]. Fig. 1 displays the basic structure of the MLP.

From Fig. 1, the perceptron contains input and output layers. The perceptron allocates weights and assigns values to the input vector, then sums up the calculated data, and iteratively updates the weights until the error is reduced to the allowable range. The obtained values are then outputted [14]. MLP introduces a Hidden Layer (HL) based on single-layer neural network, making the neural network have multiple layers. MLP can adjust the number and dimensions of hidden layers, input layers, and output layers as necessary. Each node in the HL is a perceptron, and each perceptron contains some parameters. These nodes in the HL are all fully connected, that is, the previous node output is connected together as the next layer node input. The output result of the HL is shown in Eq. (1).

$$H = XW_h + b_h \tag{1}$$

In Eq. (1), H represents the output result of the HL. X signifies the given sample. W_h represents the weight of the HL. b_h signifies the deviation coefficient of the HL. If it is a single HL, the output of HL is shown in Eq. (2).

$$O = HW_0 + b_0 \tag{2}$$

In Eq. (2), W_0 signifies the weight of the output layer. b_0 represents the deviation coefficient of the output layer. Eq. (1) and (2) are combined to obtain the input of the output layer, as displayed in Eq. (3).

$$O = (XW_h + b_h)W_0 + b_0 = XW_hW_0 + b_hW_0 + b_0 \tag{3}$$

In equation (3), the weight coefficient of the output layer is changed to W_hW_0. The deviation coefficient is changed to $b_hW_0 + b_0$. The ReLu activation function is introduced to perform nonlinear function transformation on hidden variables, making them the input of the next fully connected layer. The ReLu activation function is displayed in Eq. (4).

$$\mathrm{Re}\,Lu(x) = \max(x, 0) \tag{4}$$

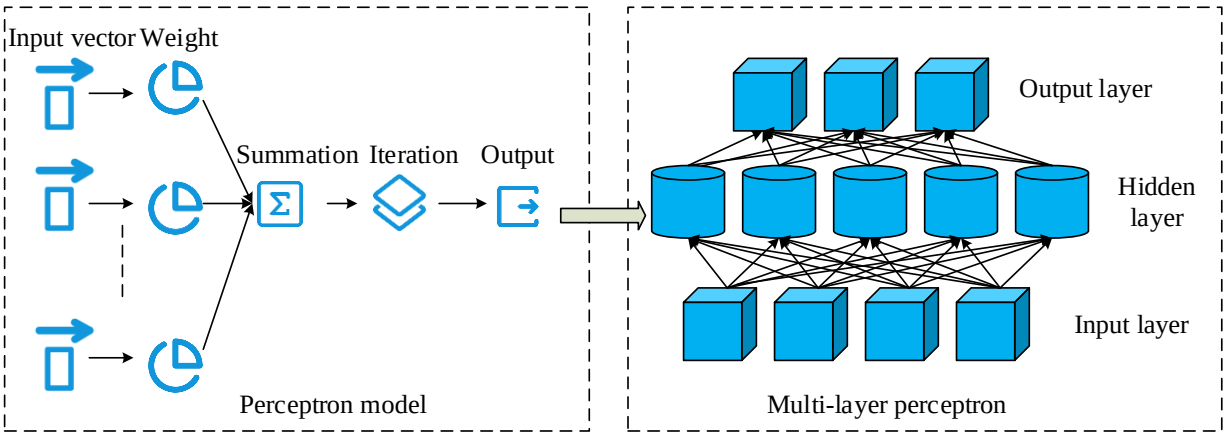


Fig. 1. Basic structure of multi-layer perceptron.

In Eq. (4), x signifies the input sample. The output expression of the MLP combined with the activation function is displayed in Eq. (5).

$$\begin{cases} H = R(XW_h + b_h) \\ O = HW_0 + b_0 \end{cases} \tag{5}$$

In Eq. (5), R represents the activation function ReLu. Afterwards, the error information is computed in Eq. (6).

$$E_i = \sum_{i=1}^{n} \frac{1}{2} (\tilde{y}_i - y_i)^2 \tag{6}$$

In Eq. (6), E_i represents the prediction error of the i-th output unit. $\tilde{y}_i$ signifies the predicted value of the i-th output unit. y_i is the i-th output unit. n signifies the number of neurons in the output layer. The influence of weights on the overall error is shown in Eq. (7).

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial y1} \cdot \frac{\partial y1}{\partial s_{y1}} \cdot \frac{\partial s_{y1}}{\partial w_j} \tag{7}$$

In Eq. (7), s_{y1} signifies the weighted sum of input y_i. w_j signifies the weight of the j-th HL. The weight value is updated, as shown in Eq. (8).

$$w_j^+ = w_j - \eta \frac{\partial E}{\partial w_j} \tag{8}$$

In Eq. (8), η represents the learning rate. The above is the calculation method of MLP, which updates weights to iterate continuously. Finally, the error is reduced to the minimum allowable range. However, the training time is long, the number of calculated parameters is too large, and over-fitting is prone to occur, which can affect the detection accuracy and efficiency. The GCN algorithm has data normalization, small parameter size, and strong extraction ability [15]. Therefore, the GCN is used to optimize the MLP to improve its accuracy and efficiency. The GCN is displayed in Fig. 2.

As shown in Fig. 2, the GCN algorithm contains an input layer, multiple hidden convolutional layers, an activation layer, and an output layer [16]. In the input layer, data is clustered, and its feature information can be obtained from the neighboring nodes of that node during clustering. Then, the clustered data is passed into the HL, which is the core layer of the algorithm. In the HL, data graph convolution operations are performed. The features of each node in the clustered data are transformed through convolutional propagation to extract and retain their own feature information, removing irrelevant information. Finally, the data is normalized using the Softmax activation function. The propagation rule for each convolutional layer is shown in Eq. (9) [17].

$$M^{(l+1)} = \sigma(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} B^{(l)} C^{(l)}) \tag{9}$$

In Eq. (9), A signifies the sum of the adjacency matrix and the closed-loop self-connection in the undirected graph. D signifies the degree matrix of A. $B^{(l)}$ is the activation unit matrix of l-th layer. $C^{(l)}$ signifies the parameter matrix of l-th layer. The nodes in the l-th layer complete the feature transformation operation, and the expression for this process is displayed in Eq. (10).

$$X^{(l+1)} = \sigma(NX^{(l)} K^{(l)} + m^{(l)}) \tag{10}$$

In Eq. (10), $X^{(l)}$ signifies the node feature of the l-th layer in the GCN. $K^{(l)}$ signifies the weight defined in layer l. σ is a nonlinear transformation. The adjacency matrix is normalized through a degree matrix, and the final expression is shown in Eq. (11).

$$x_i^{(l+1)} = \sigma(\sum_{j=N_i} D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X^{(l)} C^{(l)} + b^{(l)}) \tag{11}$$

In Eq. (11), D signifies the degree matrix of A. All numbers on the diagonal of the adjacency matrix are changed to 1 through Eq. (11). The forward propagation is shown in Eq. (12).

$$Z = soft\max(A\,\mathrm{Re}\,Lu(AXC^{(0)})W^{(l)}) \tag{12}$$

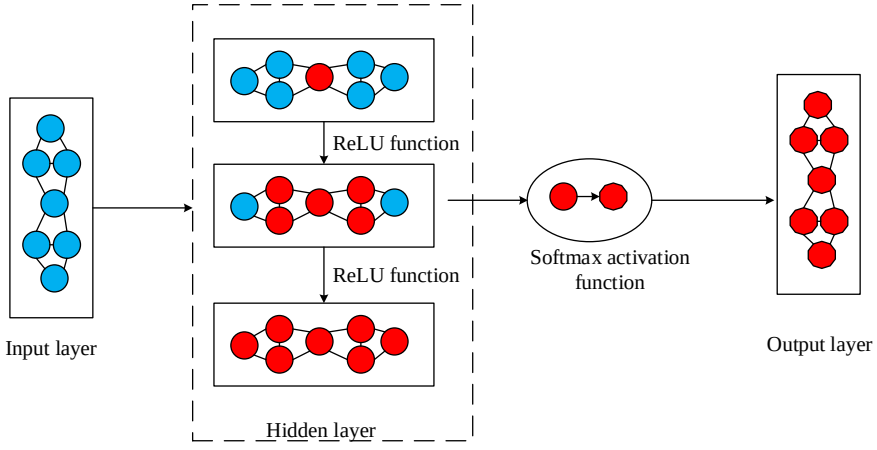


Fig. 2. GCN algorithm and basic structure diagram.

Finally, the loss function of all points is calculated, as shown in Eq. (13).

$$L = -\sum_{l=y_L} \sum_{f=1} Yl_f \ln_{lf} \tag{13}$$

In Eq. (13), f represents the soft activation function. The extraction and preprocessing of data feature information are completed through the above process. Then, the information is transmitted into MLP for data analysis. The basic flowchart of MLP optimized by GCN is shown in Fig. 3.

From Fig. 3, the optimized MLP has a one-step data preprocessing process compared with the previous one. Firstly, the data is input into the GCN module for clustering analysis, and then the convolution operation is carried out to extract the data features. Afterwords, the data is normalized through the activation function to make the data the same form. Then, the data is taken as the input value of the MLP module. In the MLP module, the weight of the received data is allocated, and then it is calculated. Through continuous iteration, the data error is reduced to the allowable range. Then, the data is output. GCN is applied to preprocess the data, unify the data type and reduce the data volume, so as to enhance the operation speed and accuracy of MLP module.

B. Construction of Intrusion Detection Model Based on GCN-MLP

This study uses an intrusion detection model on the basis of GCN-MLP algorithm to detect information intrusion behavior in network security. It is hoped that this model can solve the low detection efficiency, false positives, and missed detection in current network intrusion detection models. The network security detection model is displayed in Fig. 4.

In Fig. 4, the network security detection model contains a detection layer, a transmission layer, a monitoring layer, and an application layer. In the detection layer of the model, network intrusion information is captured and transmitted to the algorithm detection model through sensors. The intrusion information is judged in the algorithm detection model. The transmission layer transmits the judged network intrusion information to the network through the server. In the monitoring layer, the intrusion information is monitored based on the judged intrusion information. Then, the user is searched through the database server and browser, and the intrusion information is transmitted to the user. This study uses an intrusion detection model based on GCN-MLP to investigate the module in the network security detection model. The basic structure diagram of the GCN-MLP intrusion detection model is shown in Fig. 5.

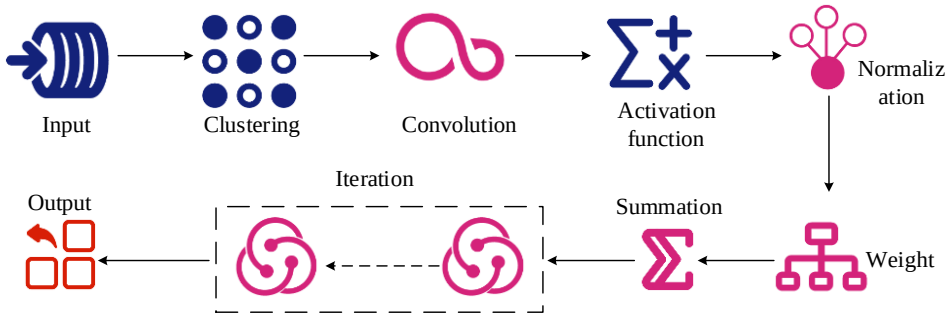


Fig. 3. Basic flowchart of GCN-MLP algorithm.

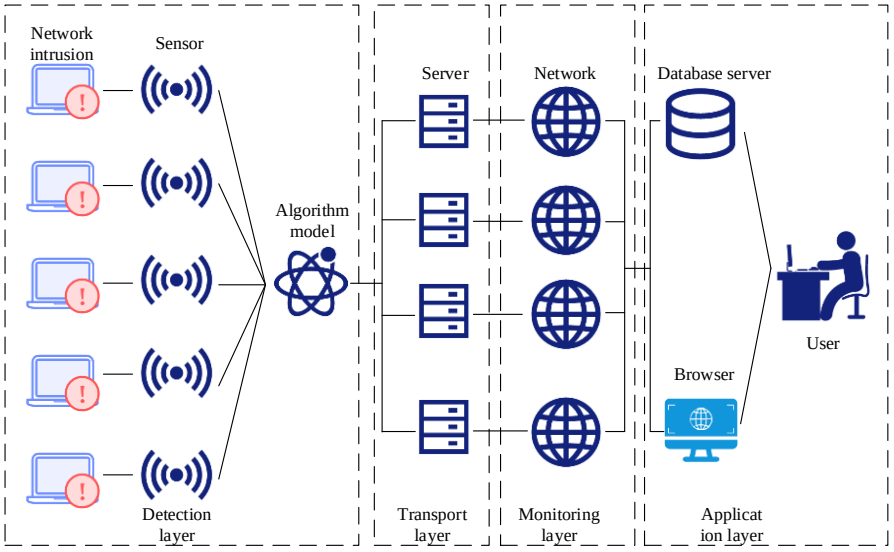


Fig. 4. Basic structure diagram of network security detection model.

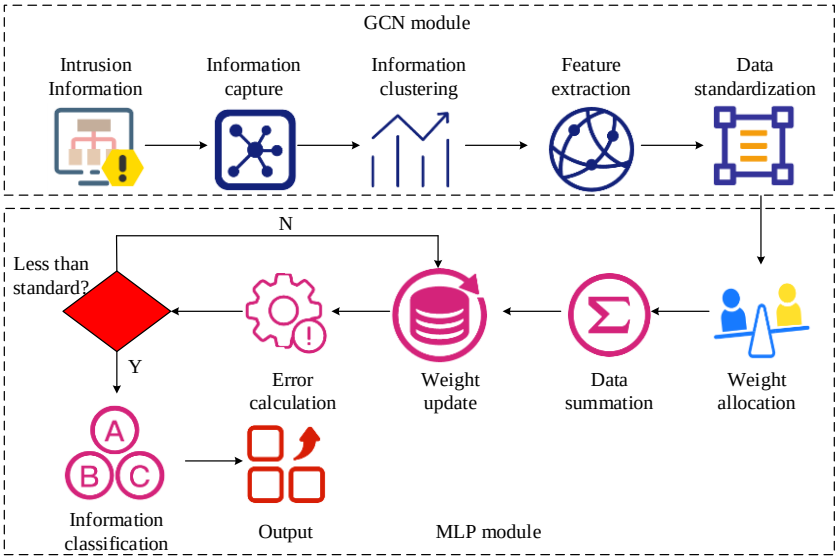


Fig. 5. GCN-MLP intrusion detection model.

As shown in Fig. 5, the model is divided into GCN module and MLP module. The GCN module captures the network intrusion information, clusters and integrates the captured information into data, extracts the features of the integrated data, and then standardizes the extracted feature information data to unify the data type. Then, the preprocessed data is sent as input information to the MLP. In this module, the incoming data is assigned weights, the weighted data is summed, the weights are updated, and the error value of the data is calculated. The error is compared with the minimum allowable error. If it is less than the allowable error, the intrusion information is classified based on the output data size to confirm the type of intrusion information. If the calculated error exceeds the allowable error, the weight is updated and the error is recalculated until the error is less than the allowable error value. The output calculation method of this model is shown in Eq. (14).

$$\hat{y} = h^T \begin{bmatrix} \Phi_{GCN} \\ \Phi_{MLP} \end{bmatrix} \tag{14}$$

In Eq. (14), h^T represents the weight matrix. The square loss is used as the loss function of the output model, as displayed in Eq. (15).

$$L = \sum_{(u,i)\in S} (\hat{y}_{ui} - y_{ui})^2 + \lambda \| \Theta \|^2 \tag{15}$$

In Eq. (15), (u,i) represents any number in the GCN dataset and MLP dataset, respectively. S represents the training dataset. $\hat{y}_{ui}$ represents the predicted score. y_{ui} is the true score. Θ represents the weight parameter. λ represents the regularization parameter. To demonstrate the model effectiveness, the root mean square error is used as the evaluation index, as displayed in Eq. (16).

$$R = \sqrt{\frac{\sum_{(u,i)\in S} (\hat{y}_{ui} - y_{ui})^2}{N}} \tag{16}$$

In Eq. (16), N signifies the total data contained. The network intrusion detection model can timely detect various security risks in the network and effectively prevent network intrusion, thereby protecting network security.

III. Results

A. Performance Analysis of GCN-MLP

To prove the superiority of GCN-MLP, the GCN-MLP is compared with Fusion Algorithm combined Convolutional Neural Network algorithm with Convolutional Attention Module (CNN-CBAM), Fusion Algorithm based on Time Convolutional Network and Bidirectional LSTM (TCN-BiLSTM), as well as Fusion Algorithm combined Principal Component Analysis algorithm with K-means clustering (PCA-K-means). Table I displays the configuration.

TABLE I. Experimental Configuration Table

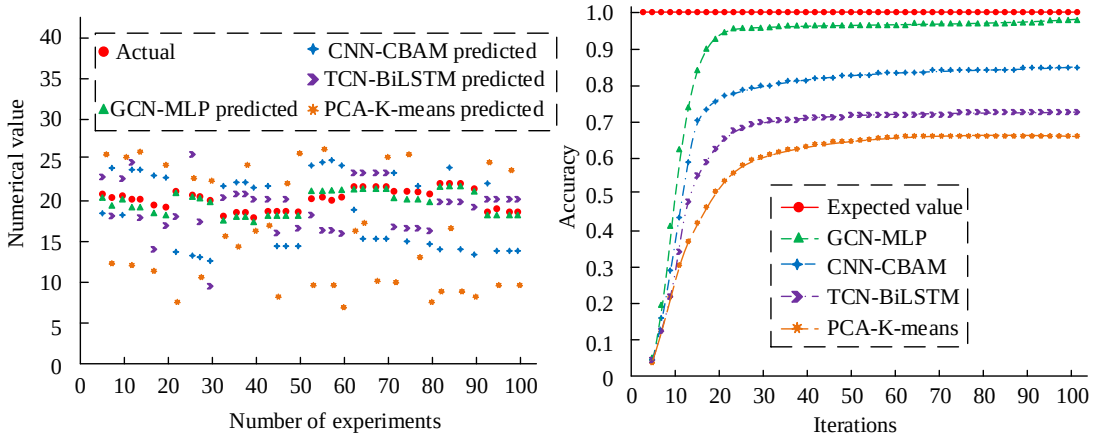
Environment	Index	Type
Hardware environment	OS system	Winds 10
	Hardpan	500G
	CPU	I7 3.4Hz
	Internal memory	4GB
Software environment	Pyrhon	Pyrhon 3.x
	Matlab	Matlab7.0

According to Table I, the environmental configuration conditions during the experiment are obtained. During the experiment, the node features, HL features, and output layer features of the GCN are 50, and the number of layers in GCN is 5. The penalty coefficient is 0.001, and the learning rate is 0.005 in the MLP. The learning rate is 0.1, the capacity is 100, the weight attenuation is 0.005, and the training frequency is 50 in the CNN. The convolution kernel in CBAM is 9, the

convolution kernel size is 3*3, the weight threshold is 0.5, and the maximum pooling layer is set to 3*3. The number of neurons in BiLSTM is 100, and the batch size is 10. The n_components in the PCA algorithm is set to none, the copy value is True, and the white value is False. The K-value in the K-means is 50, and the maximum iteration is 500. Comparative experiments are carried out on the KDD CUP 99 dataset based on the parameter settings mentioned above. The superiority of the proposed algorithm was verified by comparing the accuracy, loss function value, F1 value, detection time, and ROC curve of four algorithms. The comparison between the predicted and the actual results, as well as the accuracy results, are shown in Fig. 6.

According to Fig. 6 (a), the GCN-MLP algorithm had the closest predicted result and the smallest difference. The difference of CNN-CBAM algorithm and TCN-BiLSTM algorithm was greater than that of GCN-MLP algorithm. The PCA-K-means algorithm had the greatest difference. In Fig. 6 (b), the accuracy of the four algorithms increased when the iteration was between 0 and 20. However, when the iteration exceeded 20, the accuracy stabilized. The accuracy of the GCN-MLP algorithm stabilized at 0.98 after more than 20 iterations. The accuracy of the CNN-CBAM algorithm, TCN-BiLSTM algorithm, and PCA-K-means algorithm were 0.81, 0.69, and 0.61, respectively. Afterwards, comparative experiments are conducted on the F1 values and loss function values, as displayed in Fig. 7.

From Fig. 7, the F1 values and loss function values varied with the increase of iterations. From the Figure, the F1 value of the GCN-MLP algorithm reached its maximum value at 5 iterations, with a maximum F1 value of 0.97. However, the CNN-CBAM algorithm, TCN-BiLSTM algorithm, and PAC-K-means algorithm only reached their maximum F1 value at 10 iterations. The maximum F1 of these three algorithms was 0.92, 0.87, and 0.78. The loss function values decreased with the increase of iterations. In Fig. 7, the loss function value of the GCN-MLP algorithm stabilized at 0.03, which was much lower than the CNN-CBAM at 0.09, TCN-BiLSTM at 0.12, and PCA-K-means at 0.15. In Fig. 7, when the number of iterations was greater than 20, the loss function fluctuation range of the TCN-BiLSTM algorithm and PAC-K-means algorithm was larger, with the PCA-K-means algorithm having the largest fluctuation range and the smallest stability. Further analysis is conducted on the detection time and ROC curves, as displayed in Fig. 8.



(a)Comparison between predicted and actual values (b)Changes in model accuracy

Fig. 6. Algorithm prediction results and accuracy.

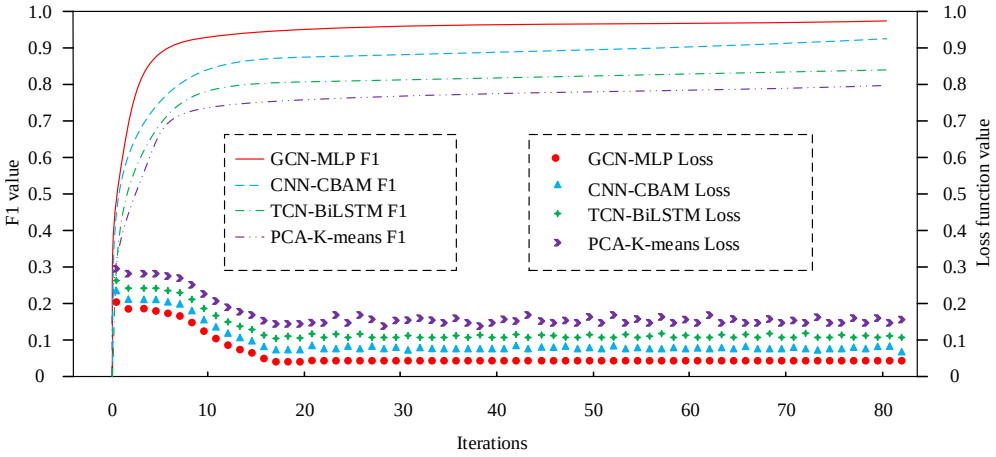


Fig. 7. Comparison of F1 value and loss function value.

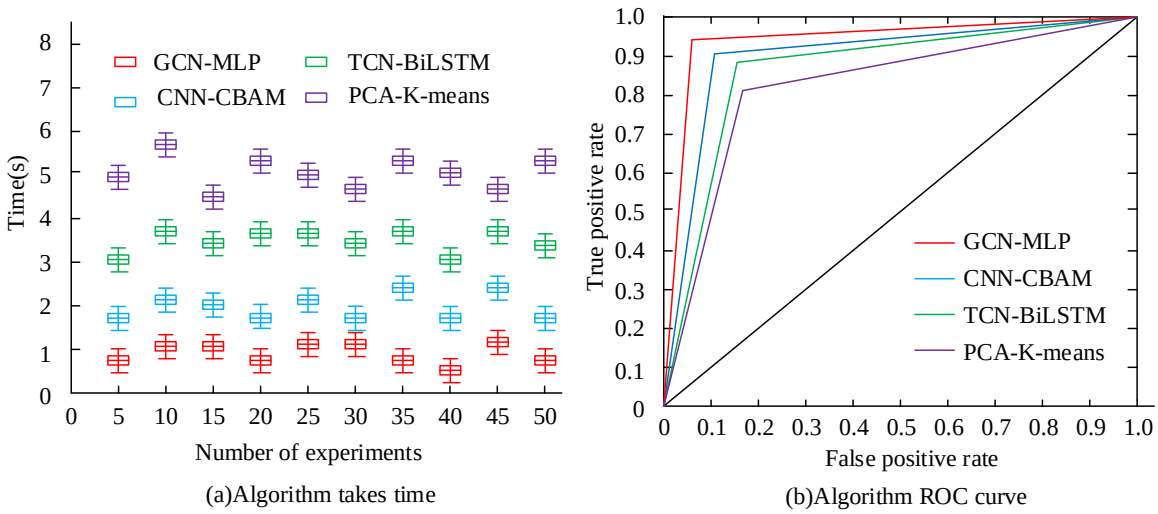


Fig. 8. Detection time and ROC curve of the algorithm.

According to Fig. 8 (a), the average detection time of the GCN-MLP was the shortest, at 1.1s. The average detection time of the CNN-CBAM was 1.9s. The average time for the TCN-BiLSTM was 3.2s. The PCA-K-means algorithm had the longest average time, which was 5.3s. The accuracy, false detection rate, and missed detection rate can be observed from the curve in Fig. 8 (b). The ROC close to the upper left corner demonstrates that the prediction accuracy is higher. From Fig. 8 (b), the ROC of GCN-MLP algorithm was closest to the upper left corner, followed by CNN-CBAM algorithm, and PCA-K-means algorithm was farthest. Therefore, among the four algorithms, GCN-MLP algorithm had the highest prediction accuracy, and PCA-K-means algorithm had the lowest prediction accuracy. GCN-MLP has the highest accuracy, fastest detection speed, and strongest stability. The overall performance is significantly better than other algorithms.

B. *Application Effect of GCN-MLP Model in Network Security Detection*

After verifying the superiority of the GCN-MLP algorithm, experimental analysis is conducted on the detection model based on the algorithm. The proposed model (Model 1) is compared with intrusion detection model integrating improved auto-encoder and residual network (Model 2), intrusion detection model integrating contrastive learning and feature selection (Model 3), and residual network detection model combined with fusion attention mechanism (Model 4). The accuracy, precision, recall, F1, underreporting rate, and detection rate of the four models are analyzed. The comparison results are shown in Fig. 9.

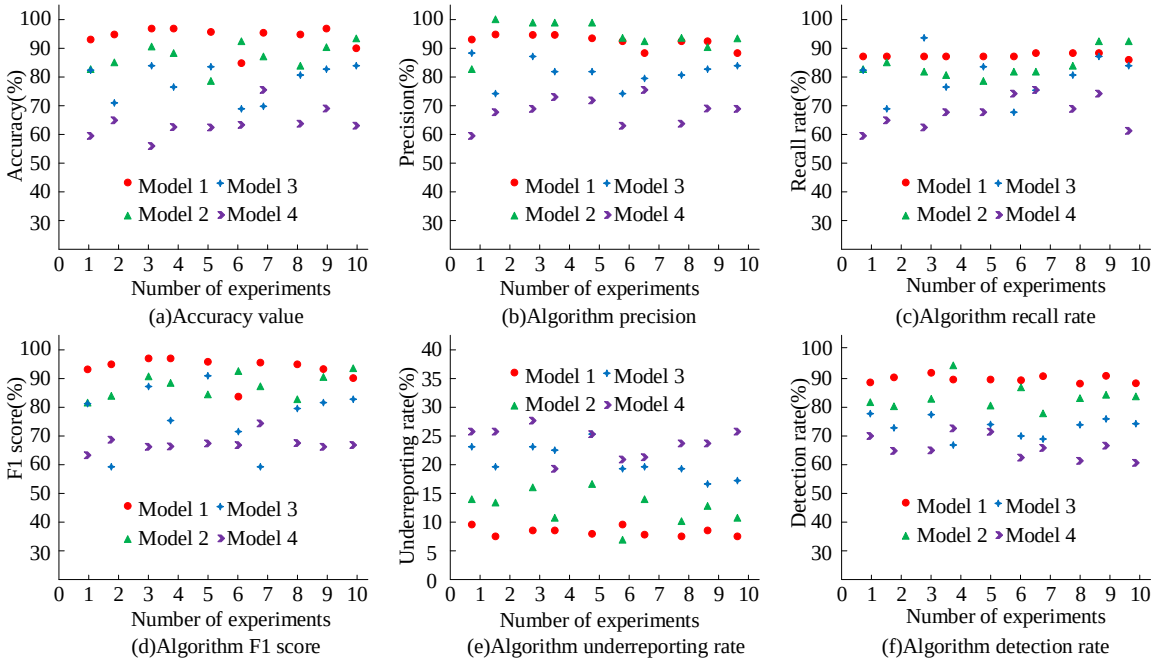


Fig. 9. Comparison of algorithm indicators.

Fig. 9 displays the comparison results of various indicators. From Fig. 9 (a), the detection accuracy of Model 1 was the highest among the four models at 98%, while the detection accuracy of Model 2, Model 3, and Model 4 were 89%, 80%, and 68%, respectively. From Fig. 9 (b), Model 2 had the highest detection precision of 97%, while Model 4 had the lowest detection precision of 78%. In Fig. 9 (c), the recall rate gradually decreased from Model 1 to Model 4. From Fig. 9 (d), after multiple experiments, the F1 value of Model 1, Model 2, Model 3, and Model 4 was 97%, 90%, 87%, and 68%, respectively. From Fig. 9 (e) and 9 (f), Model 1 had the lowest underreporting rate, but the highest data detection rate, with a underreporting rate of 6% and a detection rate of 92%, respectively. Through experiments, it is known that Model 1 has slightly lower detection accuracy than Model 2, and all other indicators are better than comparison models. The overall performance is the best among the four models. In summary, the detection model based on GCN-MLP algorithm has the best overall performance. The GCN-MLP detection model is applied to actual network security detection. The accuracy of intercepting intrusion information and the interception time of various illegal intrusions in network security detection are compared. 20 experimental results are taken, and the average accuracy and interception time of every 5 experimental results are calculated and represented by a coordinate graph. The accuracy and detection time results are shown in Fig. 10.

According to Fig. 10 (a), the average accuracy of Model 1 in network security detection was 0.98, the accuracy of Model 2 in network security detection was 0.89, and the accuracy of Model 3 was 0.71. The accuracy of Model 4 was the lowest among the four models, which was 0.57. Fig. 10 (b) shows the time it takes for four models to detect and judge intrusion information. From Fig. 10 (b), the average time for Model 1 to detect intrusion information was 0.1s, which was much lower than the 0.9s of Model 2, 2.7s of Model 3, and 4.2s of Model 4. Further experiments are conducted on the accuracy of four models in determining various types of network intrusion information, as displayed in Fig. 11.

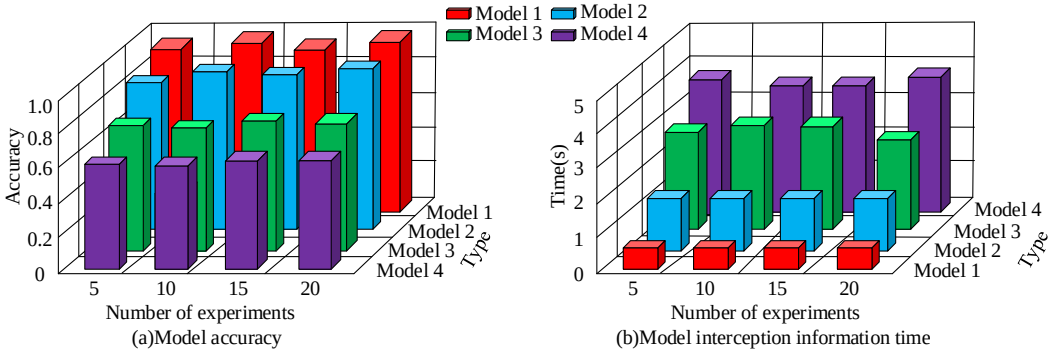


Fig. 10. Accuracy and interception time of network security inspection model.

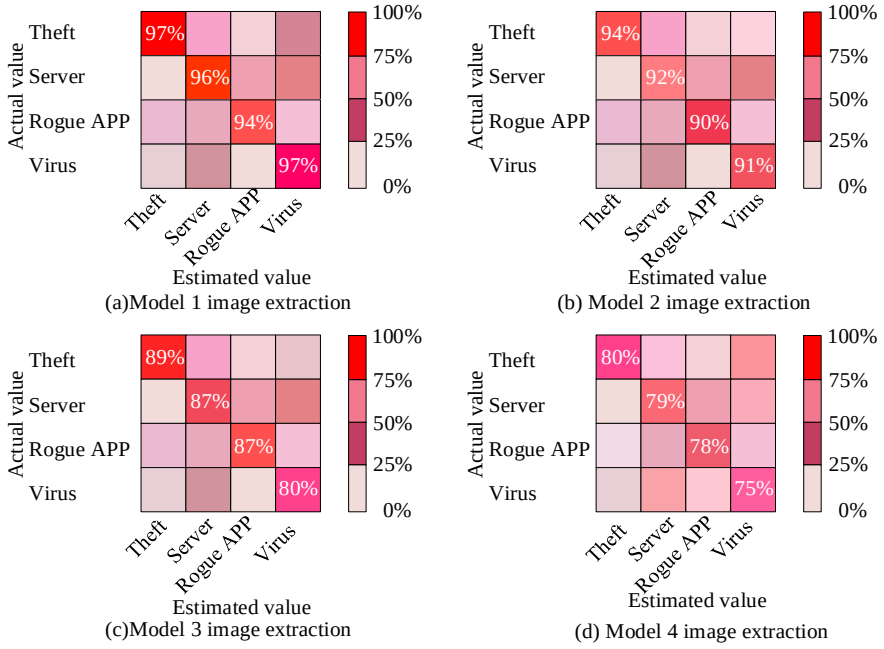


Fig. 11. Model ability to judge intrusion information.

Fig. 11 shows the accuracy results of four models in judging intrusion information encountered in network security detection. The elements on the main diagonal signify the proportion of correctly predicted intrusion information types. The elements in the lower left triangle signify the proportion of missed intrusion information types. The elements in the upper right triangle represent the proportion of false detected intrusion information types. According to Fig. 11 (a), Model 1 had a prediction accuracy of 97% for the theft intrusion information in the intrusion information, a detection accuracy of 96% for server intrusion information, a detection accuracy of 94% for malware information, and a detection accuracy of 97% for virus intrusion. The detection accuracy of Model 2 for the four types of intrusion information was 94%, 92%, 90%, and 91%, respectively. The detection accuracy of Model 3 and Model 4 for the four types of intrusion information was much lower than that of Model 1 and Model 2. From the above experimental results, the GCN-MLP has the best performance among the four detection models. This model is used in network security intrusion systems, which has the highest accuracy in detecting intrusion information.

IV. Discussion

The study verified the significant advantages of the network security detection model based on GCN-MLP in accuracy, speed, and stability through experiments. Compared with the other three algorithms, GCN-MLP not only achieved a stable high accuracy of 0.98 after 20 iterations, but also had an F1 value of 0.97. The loss function value remained stable at a lower level of 0.03, which fully demonstrated the efficiency of the algorithm. This is fitted with the conclusion drawn by Yao et al. on the GCN-MLP algorithm [18]. In addition, from the experimental results, the GCN-MLP algorithm performed equally well in detection time, averaging only 1.1s, which was much faster than the other three algorithms. In the field of network security, fast detection time means that potential threats can be responded to more quickly, effectively reducing risks, which is linked to the results drawn by He et al [19]. Further research found that when comparing the GCN-MLP detection model with three other advanced detection models, the GCN-MLP model maintained a leading position in multiple key indicators such as accuracy, recall, F1 value, and detection rate. Especially, the underreporting rate was only 6%, far lower than other models, which was extremely important in the field of network security because underreporting may lead to serious security risks.

The GCN-MLP model had a detection accuracy of over 94% for theft intrusion, server intrusion, malware information, and virus intrusion, demonstrating extremely high reliability and comprehensiveness. Compared with the algorithms and models designed by Yu et al. and Yang et al., the GCN-MLP algorithm also exhibited excellent performance. Because the deep learning model designed by Yu et al. and Yang et al. had an accuracy of only 80%-90% in network security detection, the GCN-MLP model further enhanced this standard [20-21]. Meanwhile, the stability of the GCN-MLP was also commendable. During the experiment, the fluctuation of the loss function value was relatively small. It means that in practical applications, the model can provide more reliable and consistent results. This result is significantly better than the stability of the network security protection model designed by Wang et al [22]. In summary, the network security detection model based on GCN-MLP shows significant advantages in multiple aspects, which not only proves the effectiveness of this method, but also provides strong support for its application in practical network security protection.

V. Conclusion

In response to the low accuracy, serious false positives, and missed detection rate in current information intrusion detection models, this study proposed the CN-MLP algorithm integrating GCN algorithm and MLP algorithm. Then, an information intrusion detection model was constructed based on the fused algorithm. Comparative experiments were conducted on the GCN-MLP algorithm, CNN-CBAM algorithm, TCN-BiLSTM algorithm, and PCA-K-means algorithm. The overall performance of the GCN-MLP algorithm outperformed other comparison algorithms. Subsequently, the method was compared with intrusion detection model integrating improved auto-encoder and residual network, intrusion detection model integrating contrastive learning and feature selection, and residual network detection model combined with fusion attention mechanism. The designed intrusion detection method had a much higher detection accuracy for network intrusion information than the other comparison models. In summary, the detection model on the basis of GCN-MLP has the best overall performance in network security intrusion information detection, which can effectively improve network security. However, the types of intrusion information discussed in this study are limited, and there is still uncertainty. In the future, data augmentation techniques can be used to oversample minority class samples, synthesize new minority class samples, expand the sample size, and increase detection information. Meanwhile, generative adversarial networks can be used to generate similar intrusion detection information, increase sample size, and improve the overall detection performance of the model.

References

[1] Khan M, Ghafoor L. Adversarial Machine Learning in the Context of Network Security: Challenges and Solutions. Journal of Computational Intelligence and Robotics, 2024, 4(1): 51-63.

[2] Bandewad G, Datta K P, Gawali B W, Pawar, S. N. Review on Discrimination of Hazardous Gases by Smart Sensing Technology. Artificial Intelligence and Applications. 2023, 1(2): 86-97.

[3] Fu Y, Du Y, Cao Z, Li Q, Xiang W. A deep learning model for network intrusion detection with imbalanced data. Electronics, 2022, 11(6): 898-901.

[4] Hnamte V, Nhung-Nguyen H, Hussain J, Hwa-Kim Y. A novel two-stage deep learning model for network intrusion detection: LSTM-AE. Ieee Access, 2023, 11(5): 37131-37148.

[5] Dai J, Zhu W, Luo X. A targeted universal attack on graph convolutional network by using fake nodes. Neural Processing Letters, 2022, 54(4): 3321-3337.

[6] Diao C, Zhang D, Liang W, Li K C, Hong Y, Gaudiot J L. A novel spatial-temporal multi-scale alignment graph neural network security model for vehicles prediction. IEEE Transactions on Intelligent Transportation Systems, 2022, 24(1): 904-914.

[7] Deng X, Zhu J, Pei X, Zhang L, Ling Z, Xue K. Flow topology-based graph convolutional network for intrusion detection in label-limited IoT networks. IEEE Transactions on Network and Service Management, 2022, 20(1): 684-696.

[8] Al-Ibraheemi F A, Hazzaa F, Jabbar M S, Tawfeq J F, Sekhar R, Shah P, Parihar S. Intrusion Detection in Software-Defined Networks: Leveraging Deep Reinforcement Learning with Graph Convolutional Networks for Resilient Infrastructure. Full Length Article, 2024, 15(1): 78-87.

[9] Setitra M A, Fan M, Agbley B L Y, Bensalem Z E A. Optimized MLP-CNN Model to Enhance Detecting DDoS Attacks in SDN Environment. Network, 2023, 3(4): 538-562.

[10] Shewale Y, Kumar S, Banait S. Machine Learning Based Intrusion Detection in IoT Network Using MLP and LSTM. International Journal of Intelligent Systems and Applications in Engineering, 2023, 11(7): 210-223.

[11] Najar A A, Manohar Naik S. DDoS attack detection using MLP and Random Forest Algorithms. International Journal of Information Technology, 2022, 14(5): 2317-2327.

[12] Diao C, Zhang D, Liang W, et al. A novel spatial-temporal multi-scale alignment graph neural network security model for vehicles prediction. IEEE Transactions on Intelligent Transportation Systems, 2022, 24(1): 904-914.

[13] Alsirhani A, Alshahrani M M, Abukwaik A, Taloba A I, Abd El-Aziz R M, Salem M. A novel approach to predicting the stability of the smart grid utilizing MLP-ELM technique. Alexandria Engineering Journal, 2023, 74(5): 495-508.

[14] Wang W, Wen F, Zheng H, Ying R, Liu P. Conv-MLP: A convolution and MLP mixed model for multimodal face anti-spoofing. IEEE Transactions on Information Forensics and Security, 2022, 17(4): 2284-2297.

[15] Pankova M, Kwilinski A, Dalevska N, Khobta V. Modelling the Level of the Enterprise'Resource Security Using Artificial Neural Networks. Virtual Economics, 2023, 6(1): 71-91.

[16] Zheng H, Li X, Li Y, Yan Z, Li T. GCN-GAN: integrating graph convolutional network and generative adversarial network for traffic flow prediction. IEEE Access, 2022, 10(5): 94051-94062.

[17] Huang D, Liu H, Bi T, Yang Q. GCN-LSTM spatiotemporal-network-based method for post-disturbance frequency prediction of power systems. Global Energy Interconnection, 2022, 5(1): 96-107.

[18] Yao Z, Yu J, Zhang J, He W. Graph and dynamics interpretation in robotic reinforcement learning task. Information Sciences, 2022, 611(4): 317-334.

[19] He J, Abueidda D, Koric S, Jasiuk I. On the use of graph neural networks and shape-function-based gradient computation in the deep energy method. International Journal for Numerical Methods in Engineering, 2023, 124(4): 864-879.

[20] Yu J, Ye X, Li H. A high precision intrusion detection system for network security communication based on multi-scale convolutional neural network. Future Generation Computer Systems, 2022, 129(6): 399-406.

[21] Yang H, Zhang Z, Xie L, Zhang L. Network security situation assessment with network attack behavior classification. International Journal of Intelligent Systems, 2022, 37(10): 6909-6927.

[22] Wang Z, Xie X, Chen L, Song S, Wang Z. Intrusion detection and network information security based on deep learning algorithm in urban rail transit management system. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(2): 2135-2143.