Design of a Lightweight Network Intrusion Detection System Based on Artificial Intelligence Technology

Li He

School of Data and Information, Changjiang Polytechnic, Wuhan, 430074, China
E-mail: xianxianheli@163.com

Received 21 March 2024; Accepted 19 June 2024

Abstract

Network security issues have become crucial with the boost of Internet of Things technology. To detect lightweight network intrusion, this research improves the population initialization mode of given the genetic algorithm given the Pearson correlation coefficient and constructs a feature selection model. In view of the one-dimensional convolutional neural network model, it introduces the gated cyclic unit neural network model. It uses pruning operations to realize the lightweight of the model and build an intrusion detection model. The results showed that the accuracy, detection rate, and time average of the improved genetic algorithm were 79.55%, 90.32%, and 189.4 s, which were 14.87%, 30.35%, and 33.05% higher than the traditional genetic algorithm model, respectively. The intrusion detection model has achieved an accuracy of 95.0%, and the loss function value is 0.15. Compared with other deep learning models, it is more robust and performs better in intrusion detection. The average accuracy of the model testing after lightweight is 88.6%, the average detection rate is 98.12%, and the average testing time is 82 s, which improves the model's performance compared to

Journal of Cyber Security and Mobility, Vol. 13_5, 1129–1148.
doi: 10.13052/jcsm2245-1439.13514

before lightweight. This study could markedly enhance the accuracy and detection rate of lightweight network intrusion detection, with higher detection efficiency and better performance, and possesses an essential influence in improving network security.

Keywords: Lightweight network, intrusion detection, Pearson correlation coefficient, genetic algorithm, one dimensional convolutional neural network.

1 Introduction

With the boost of the Internet, network security issues have become more essential. Intrusion detection (ID) systems have emerged as one of the critical technologies for protecting network security. Recently, deep learning technology has reached excellent success in various aspects and is gradually being applied in the field of ID [1]. Compared with traditional ID systems, deep learning-based ID systems can automatically learn and extract advanced features, thereby better capturing the complex patterns and changes of network intrusion behavior. However, because network ID is faced with large-scale and high-dimensional (HD) network traffic data, ID systems given deep learning often need huge network models and complex computing resources to ensure the accuracy and reliability of detection [2]. Resource-constrained scenarios, such as Internet of Things devices and embedded systems, traditional deep learning models cannot meet practical needs. Traditional deep learning models cannot meet practical needs in some resource-constrained scenarios, such as Internet of Things devices and embedded systems. Therefore, this study aims to design and study a lightweight network ID system given deep learning to meet the requirements of network security in resource constrained scenarios. This study employs a novel approach to feature selection, integrating the Pearson correlation coefficient with a genetic algorithm. This integration optimizes the feature selection process in network ID through genetic operations, including natural selection, crossover, and mutation. This combination provides a new optimization path for feature selection. The construction of lightweight network ID models is impeded by the problem of gradient disappearance in one-dimensional (OD) convolutional neural networks (CNN). The introduction of a gated loop unit addresses this issue, reducing the number of parameters in the model through pruning technology. This design effectively reduces model complexity and computational requirements while maintaining model performance.

This study is divided into four parts. The first part is domestic and foreign scholars' research on lightweight networks and ID models. In the second part, the Feature selection model is constructed by improving the genetic algorithm (GA) in view of the Pearson correlation coefficient (PCC). Given the OD CNN model, it introduces the gated cyclic unit neural network model to build an ID model. The third part tests and analyzes the model, while the fourth part summarizes the article and proposes shortcomings.

2 Related Work

A lightweight network is a kind of network that improves the volume and speed of an ordinary network. Some scholars have studied the privacy protection of lightweight networks. R S Miyanaji et al. [3] found that most existing networks require a large amount of memory and computation for continuous authentication. To improve this issue, this study presented a protocol between nodes and servers in the Internet of Things. The proposed method offered privacy protection through anonymous nodes, Forward secrecy without asynchronous encryption, and key agreement. Compared with the review protocol, the computation time of nodes and servers in authentication was reduced by 16.8% and 8.7%, with a communication cost of 1902 bits.

F Algarni [4] found that object programming languages create security vulnerabilities that exploited for launching attacks. To protect Internet of Things systems from intrusions, a security method for protecting the operation of Internet of Things systems from memory heap infiltration and address modification attacks is proposed. The results showed that this method can effectively resist hacker intrusion by encrypting object garbage collection at runtime to prevent targeted attacks.

P Krishnakumar [5] believed that while the Internet of Things is widely used, the relevant problems are increasingly emerging due to the need for more human intervention. To ensure data security in the Internet of Things, it used an extreme method called lightweight encryption, mainly focusing on encryption, hashing, and authentication technologies. The results indicated that this method effectively protects privacy in lightweight networks. V Dahiphale et al. [6] believed that since most Internet of Things devices operate on 8-bit controllers with limited storage and computing power, lightweight passwords need to be utilized on both the sending and receiving ends to achieve encryption and decryption. In light of the aforementioned considerations, a novel architectural framework for the hardware implementation of ANU cryptography is put forth, accompanied by an analysis of the associated outcomes. The experimental results proved that ANU password is the best choice for achieving security in systems with minimal resources.

The deep learning neural network model was extensively utilized in network intrusion, which can markedly enhance the detection rate of ID. Some scholars have related research results given this. To enhance the precision of identification, Z X Ran [7] proposed a two-step network identification model, drawing inspiration from the GoogLeNet inception model and the deep convolutional neural network (CNN) model. This model employs the GoogLeNet inception model to address the binary classification of network packets and to extract the original data characteristics and traffic characteristics of network packets. Furthermore, it utilizes the depth CNN to recognize features. The results showed that the model's accuracy reached 99.63%. To build an effective traffic classification method, A H Azizan et al. [8] evaluated the decision tree (DT), random forest (RF), and support vector machine (SVM) classifiers, and proposed a network ID model in view of these machine learning techniques. The outcomes showcased that the average accuracy (AA) of the SVM classifier is 98.18%, the AA of the DT is 96.50%, and the AA of the RF is 96.76%. To establish a real-time detection and dynamic defense security system, X Luo [9] constructed an adaptive network ID and defense system model in view of an automatic fuzzy rule generation strategy. The experiment showcased this method has adaptability and scalability. X Li et al. [10] found that the Krill group algorithm is an efficient Swarm intelligence algorithm that possesses excellent function. A special improved Krill swarm algorithm was presented in response to the issue of low ID efficiency and high false alarm rate resulting from the increase in HD data. The results showed that the model markedly maintains high accuracy.

In conclusion, although deep learning neural networks are frequently employed in network identification, they are less frequently utilized in lightweight network identification. Moreover, existing deep learning networks are prone to the problem of parameter redundancy. Therefore, this research is in view of OD CNN, and improved by GA and PCC. In addition, it uses pruning method to reduce the number of parameters, which has important reference value in the field of lightweight network ID.

3 Construction of Lightweight Network Intrusion Detection System based on Feature Selection and Artificial Intelligence Technology

To achieve accurate identification in lightweight networks, this chapter is divided into two sections, which together constitute a model. The first section employs a traditional GA in conjunction with PCC in order to enhance the initialization of the population and construct a feature selection model. The second part, in view of the OD CNN model, introduces the gated Recurrent neural network model and uses pruning operations to build a lightweight ID system.

3.1 Construction of Genetic Algorithm Feature Selection Model on the Ground of Pearson Correlation Coefficient

PCC is a commonly used statistic utilized to measure the strength and direction of the linear relation in two variables. It represents the degree of linear correlation between two variables, with values ranging from -1 to 1 [11]. Covariance is a metric used to measure the correlation between two variables, and the PCC is the covariance of standardized random variable values. Therefore, the PCC calculation formula is shown in Equation (1).

$$\begin{aligned} Pearson(X,Y) &= Cov(X,Y)/\sigma_X\sigma_Y \\ &= \sum_{i=1}^{N} \frac{(x_i-\bar{x})(y_i-\bar{y})}{N\sigma_X\sigma_Y} \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\bar{x}}{\sigma_X}\right)\left(\frac{y_i-\bar{y}}{\sigma_Y}\right) \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{x_i-\bar{x}}{(\|X-\bar{X}\|_2)}\frac{y_i-\bar{y}}{(\|Y-\bar{Y}\|_2)} = Cov(S_X,S_Y) \end{aligned} \tag{1}$$

In Equation (1), X and Y represent the initial random variables. x_i and y_i represent standardized random variables. $\bar{x}$ and $\bar{y}$ represent the mean of variables. The relationship between covariance and variable variance is shown in Equation (2).

$$|Cov(S_X,S_Y)| \leq D(S_X)D(S_Y) = 1 \tag{2}$$

In Equation (2), $Cov(S_X,S_Y)$ represents the covariance of the initial random variable; $D(S_X)D(S_Y)$ represents the variance of the initial random variable. It can be proven that the PCC has a value range of [-1,1]. When PCC $= 1$, it indicates that the two variables are totally positively correlated. When the PCC is -1, it suggests that the two variables are completely negatively correlated. When PCC $= 0$, it indicates that there is no linear relation between two variables, and the two variables are independent and

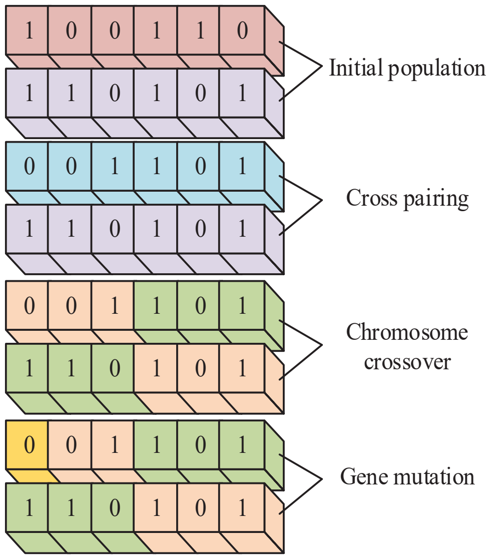


Figure 1 Schematic diagram of various operations of genetic algorithm.

not affected by each other [12]. The closer the absolute value of the PCC is to 1, the stronger the linear relationship between the two variables. GA is an optimization algorithm that simulates biological evolution. It searches for the optimal solution or problems close to the optimal solution by simulating Natural selection, genetic mutation, crossover, and other operations. The schematic diagram of each operation is illustrated in Figure 1.

In traditional GA, the selection of the initial population is usually randomly generated. Still, this randomly generated initial population may have problems, such as not fully covering the search space, resulting in a slow Rate of algorithm convergence or falling into a locally optimal solution. In GA, each feasible solution is encoded into an OD vector, and the PCC can be extended to measure the correlation between two random vectors [13]. Therefore, it adopts PCC to improve the population initialization of GA and strengthen the optimization. Initially, a certain number of chromosomes are randomly generated as the initial population. It randomly generates a chromosome and compares it with the PCC values of each chromosome in the initial population [14]. The study sorted the PCCs obtained from the calculation, selecting individuals with lower correlation with other individuals as a part of the initial population given the size of the correlation coefficients. It will choose individuals as part of the initial population, and others can continue to join the initial population to ensure population diversity. If the initial population does not meet the requirements, the selection step can be repeated until the initial population meets the set requirements. This study uses the random tournament selection method to select the initial population of a GA in view of PCC. This method simulates a competition by randomly selecting individuals for comparison and selecting individuals who perform better in the competition to form the initial population [15]. Firstly, it sets the scale of a competition and randomly selects individuals from the initial population as a group of competition groups. This study evaluates individuals within the competition group and determines their strengths and weaknesses in view of a fitness function. The fitness function is calculated as shown in Equation (3).

$$\mathit{fitness}(x) = a \times \mathit{calc\_Acc}(X) + (1-a) \times \left(\sum_{i}^{m} X_i\right)^{-1} \tag{3}$$

In Equation (3), X represents the chromosome of any feasible solution. X_i represents the selection of chromosome for the i-th feature in the dataset. $\mathit{calc\_Acc}(X)$ denotes the accuracy calculation function of the classifier. a represents a weighted real number used to balance accuracy and number of features. In view of the evaluation results, it selects the best performing individual within the competition group as a part of the initial population until it meets the population requirements. This study uses a single point crossover operator as a method to improve the crossover operation of GA. This method generates new individuals by randomly selecting a crossover point, cutting the chromosomes of two individuals at that crossover point, and exchanging gene fragments with each other [16]. This method employs a random selection process to identify a specific location on the chromosome as the intersection point. At this point, two parent individuals are divided into two gene fragments. Gene fragments from the two parent individuals are then exchanged, resulting in the generation of two new individuals. These new individuals are subsequently incorporated into the next generation's population as offspring. The improved GA employs the basic bit mutation operator as the mutation operation method. This operator randomly selects one or more bits for mutation, inverts or randomly transforms the gene bits in the chromosome into other effective genes. This method randomly selects one or more gene loci in chromosomes, mutates the selected gene loci, and adds the mutated individuals as offspring to the next generation population. The flowchart of the GA in view of PCC is shown in Figure 2.

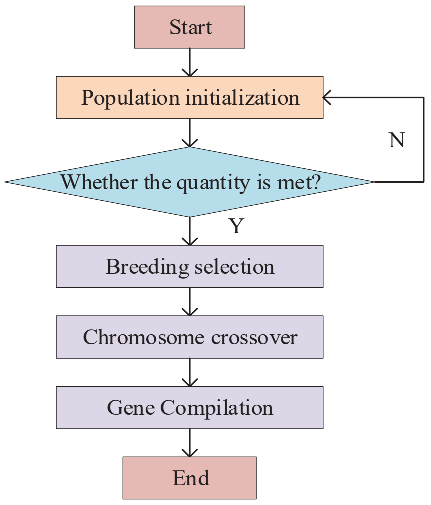


Figure 2 Model construction flow chart.

3.2 Construction of Network Intrusion Detection System on the Ground of Deep Learning Neural Network Model

CNN deep learning model, which is composed of convolution layer (CL), pooling layer (PL), full connection layer, etc., uses convolution operation as the core to achieve feature extraction and classification [17]. The CL is the main module of CNN, which performs convolution operations on the input image by using a set of learnable filters. These filters locally connect each small region of the input image in a sliding window manner to extract local features. The relevant calculation is showcased in Equation (4).

$$y_j^l = f\left(\sum_{i \in M} y_i^{l-1} \times w_{ij}^l + b_i'\right) \tag{4}$$

In formula (4), f serves as the Activation function. M serves as the set of input feature vectors. y_i^{l-1} represents the i-th vector of the output of the $l-1$-th CL. w_{ij}^l serves as the weight of the convolutional kernel (CK). b_i' represents the offset value. y_j^l serves as the output of the layer l CNN after being convolved by the i CK. The bias operation is shown in Equation (5).

$$w = k_{out} \times kz \times k_{in} + bias \tag{5}$$

In Equation (5), k_{in} serves as the quantity of input channels in the CL. kz denotes the CK length of the CL. k_{out} means the quantity of output channels.

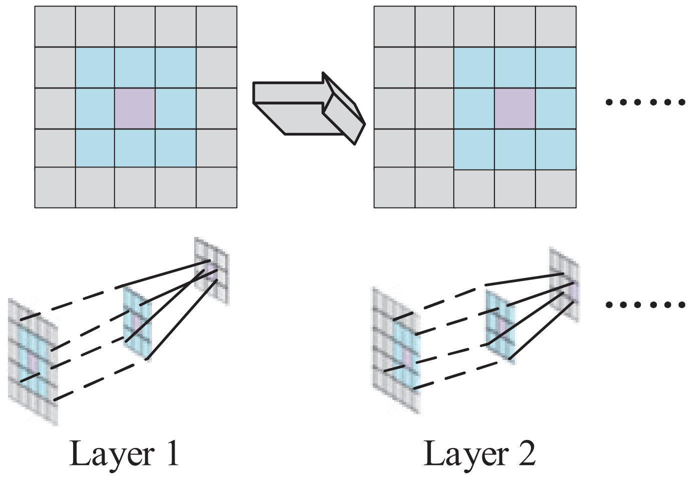


Figure 3 Sliding window execution mode.

$bias$ serves as the quantity of offset terms. Convolutional operations can capture features such as edges, textures, and shapes in images. The sliding window execution mode is shown in Figure 3.

Compared with the traditional CNN, the OD CNN model has lower environmental requirements and higher efficiency. Moreover, most devices in the Internet of Things environment have weak computing power and limited memory space, and the sampled data is mostly OD features. Therefore, the OD CNN model has more advantages when dealing with lightweight networks. OD CNN is a deep learning model applied to serial data or time series data. It extracts local features in the input sequence by applying OD convolution operations and performs classification or regression tasks through pooling operations and full connection layers [18]. The output feature map of the OD convolution operation will have the same sequence length as the input sequence, but may reduce or expand the dimension of the feature. Fill operations and stride parameters can be used to control the size of the feature map. The filling operation can add additional elements on both sides of the input sequence to maintain the same length of the output feature map as the input sequence. The stride parameter determines the distance the sliding window moves on the input sequence, and the filling operation calculation formula is presented in Equation (6).

$$\dim_{out} = 1 + \frac{\dim_{in} + 2size_{padding} - size_{kernel}}{s} \tag{6}$$

In Equation (6), $\dim_{out}$ represents the dimension of the output, $\dim_{in}$ denotes the input dimension. $size_{padding}$ means the filling size, $size_{kernel}$ represents the size of the CK. s serves as the convolution step size of the convolution kernel. The PL is used for downsampling the output of the CL, usually using either maximum pooling or average pooling to reduce the dimensionality of the feature map and diminish the quantity of model parameters. The calculation formula for PL is showcased in Equation (7).

$$z_j^l = \beta(down(y_j^l) + b) \tag{7}$$

In Equation (7), β, b denote scalar parameters. y_j^l means the data after convolution of the l layer CNN. $down(y_j^l)$ represents the down sampling Choice function. Pooling operation helps to extract spatial invariance of images, while also reducing computational complexity and memory requirements. In OD CNN, there is a flattening layer. The relevant calculation is demonstrated in Equation (8).

$$FLOPS_{fc} = size_b \times (\dim_{in} + 1) \times \dim_{out} \tag{8}$$

In Equation (8), $size_b$ represents the batch quantity. This layer flattens the multidimensional feature map into an OD vector and inputs it into a fully connected system. The fully connected layer is utilized for mapping the output of CL and PL to specific categories, performing weight calculation and nonlinear transformation under the sigmoid function, and obtaining the final output result. The calculation formula for the fully connected layer is presented in Equation (9).

$$y_j^l = \sum w_j^l * x_i^{l-1} + b_j^i \tag{9}$$

In Equation (9), w_j^l denotes the connection weight. b_j^i represents the offset value; x_i^{l-1} means the characteristic value. y_j^l represents the output result. To solve the problem of gradient explosion or disappearance in OD CNN, it introduced gated recurrent unit neural network (GRU) for building an OD CNN in view of 1D CNN rated recurrent unit neural network (1D CNN-GRU) [19]. The relevant diagram of the GRU framework structure is demonstrated in Figure 4.

GRU is improved in view of the long short-term memory network (LSTM) model. LSTM contains forgetting gates, input and output gates [20]. The expression of forgetting gates is expressed in Equation (10).

$$f_i = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \tag{10}$$

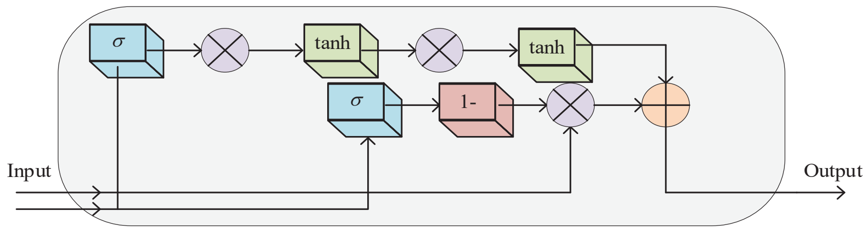


Figure 4 Schematic diagram of GRU frame structure.

In Equation (10), h_{t-1} serves as the output of the previous unit, x_t denotes as the output of the current unit, σ represents the sigmoid function. W_f means weight, b_f represents deviation. The content of updates and substitutions is determined by the sigmoid function and tanh function in the input gate, and the state of the LSTM unit is updated by combining the results of the two functions, as shown in Equation (11).

$$\begin{cases} it = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + bc) \end{cases} \tag{11}$$

In Equation (11), $\tilde{C}_t$ represents the content replaced by the tanh function. The final update status formula is shown in Equation (12).

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{12}$$

In Equation (12), C_{t-1} denotes the unit status before the update. C_t represents the updated status. In the output gate, the sigmoid function and tanh function determine the content of the output and obtain the final output of the output gate, as shown in Equation (13).

$$\begin{cases} ot = \sigma(W_o[h_{t-1}, x_t] + b_o) \\ ht = ot \times \tanh(C_t) \end{cases} \tag{13}$$

In Equation (13), ot represents the content that determines the output. ht denotes the final output. In GRU, only the update and reset gates are included, which diminish the number of parameters compared to LSTM and can better preserve the information of the data in the time dimension. The updated door calculation formula is shown in Equation (14).

$$Z_t = \sigma(W_z X_t + W_z H_{z-1} + b_z) \tag{14}$$

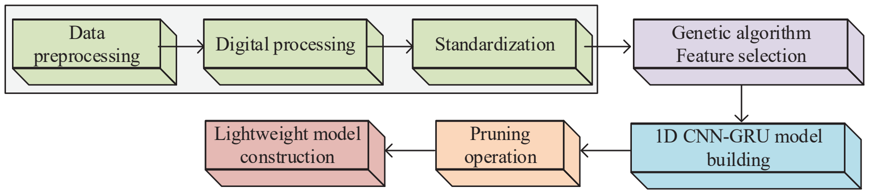

Data preprocessing
Digital processing
Standardization
Genetic algorithm Feature selection
1D CNN-GRU model building
Pruning operation
Lightweight model construction

Figure 5 Flow chart of 1D CNN-GRU model construction.

In Equation (14), W_z denotes the weight parameter. b_z represents the deviation parameter. The calculation formula for resetting the door is shown in Equation (15).

$$r_t = \sigma(W_r X_t + W_r H_{t-1} + b_r) \tag{15}$$

In Equation (14), W_r represents the weight parameter. b_r denotes the deviation parameter. Pruning operations are used to select and reject the weights in the model to achieve lightweight detection. This is visible in Equation (16).

$$s_t = f(t) = s_f + (s_i - s_f)\left(1 - \frac{t - t_0}{n\Delta t}\right)^3, \ t \in \{t_0, t_0 + \Delta t, \ldots t_0 + n\Delta t\} \tag{16}$$

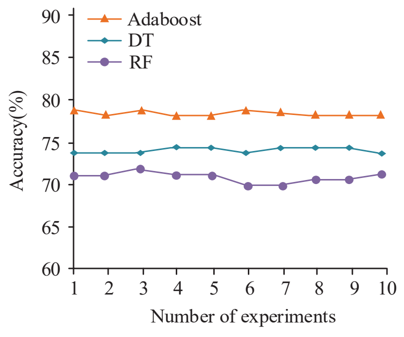
In Equation (16), t_0 represents the initial step, s_i denotes the initial pruning rate. s_f represents the sparse value of the target. The flowchart for constructing the 1D CNN-GRU model is shown in Figure 5.

4 Performance Test and Analysis of Improved Genetic Algorithm Feature Selection Model and Intrusion Detection Model

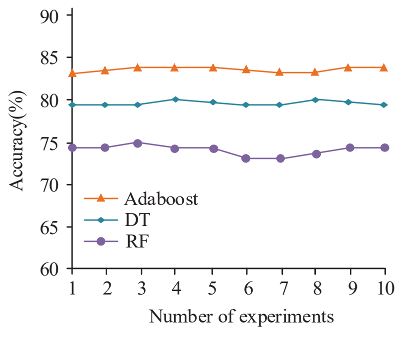
To evaluate the performance of the ID model, this section is divided into two for testing. The first tests the GA in view of Pearson coefficient improvement. The second tests the 1D CNN-GRU ID model in view of an improved GA.

4.1 Performance Test of Improved Genetic Algorithm Feature Selection Model

To determine the optimal classifier and weight real number a, the detection performance of the algorithm is compared under different classifiers. In the performance test of feature selection model, the DT, RF and Adaboost model



(a) Accuracy of different classifiers with a weight of 0.5



(b) Accuracy of different classifiers with a weight of 0.7

Figure 6 Accuracy of different classifiers under different weights.

are studied for comparison test, and the values of a are 0.5 and 0.7. The experimental environment is a Windows 10 OS, with an Intel Core i7 CPU and a 32GB GPU. It uses Python software for encoding and selects the UNSW-NB15 dataset for the database. The UNSW-NB15 dataset is provided by the Network Research Laboratory at the University of New South Wales, Australia. It contains more than 150,000 network traffic samples, including normal traffic and various types of network attack traffic, such as DoS attacks, remote to local attacks, local to remote attacks and 26 different types of network attack. The use of the UNSW-NB15 dataset can ensure the universality of experimental results. A total of 10 experiments are conducted to average the accuracy of different classifiers under different weights, as shown in Figure 6.

Figure 6 shows that when a is 0.5, the AA values of DT, RF, and Adaboost are 74.5%, 70.8%, and 78.6%, respectively. When a is 0.7, the AA values of DT, RF, and Adaboost are 79.7%, 74.2%, and 83.4%, respectively, indicating that the Adaboost model possesses the most excellent accuracy. Three classifiers are trained to further determine the selection of classifiers, and the results are shown in Figure 7.

Figure 7 shows that DT and RF both fell into two local optima during training, and the Adaboost model converges after 20 iterations, with the best optimization ability. Therefore, the Adaboost classifier is used for testing. For testing the feature selection model of the GA with improved PCC, according to the results in Figure 6, the threshold of Pearson coefficient in the experiment is set to 0.4, the population size is set to 50, and the quantity of iterations is set to 200. The classifier is selected as Adaboost, with a weight

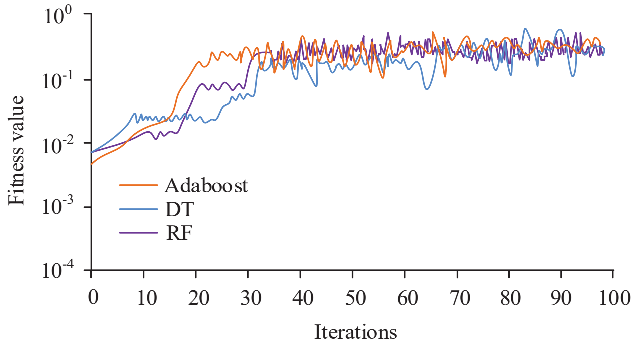


Figure 7 Training results of different classifiers.

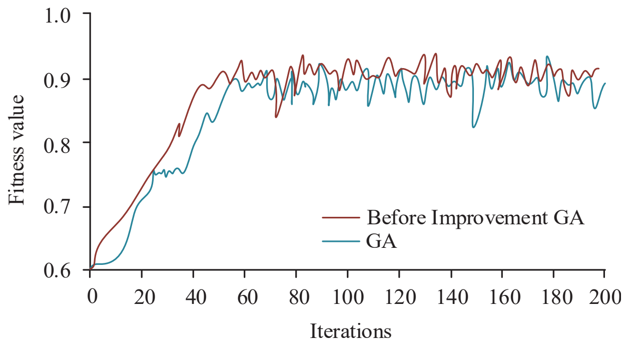


Figure 8 Algorithm iteration curve before and after improvement.

set to 0.7, a crossover probability set to 1.0, and a mutation probability set to 0.1. The algorithm iteration curve before and after improvement is shown in Figure 8.

Figure 8 shows that the pre improved GA model briefly fell into a local optimal solution, while the improved GA model did not. Moreover, the figure also shows that the improved GA model converges earlier than the pre improved GA model, achieving the best fitness value. Therefore, the PCC enhances the diversity of the population and enhances the optimization. To further demonstrate the performance advantages and disadvantages of the two models, the accuracy and time of ID feature extraction of the two algorithms are compared, and a total of 5 tests are conducted. The outcomes are showcased in Table 1.

Table 1 shows that the AA of the GA model is 69.25%, while the AA of the improved GA model is 79.55%, an improvement of 14.87%. The average

Table 1 Test indicator results of two models

Test Indicators	Repetitions	GA (%)	Pre Improvement GA (%)	Change in Amplitude (%)
Accuracy	1	70.15	80.15	14.26%
	2	69.55	78.59	13.00%
	3	68.27	79.68	16.71%
	4	69.17	79.43	14.83%
	5	69.09	79.88	15.62%
Average value	–	69.25	79.55	14.87%
Detection rate	1	70.15	89.44	27.50%
	2	69.81	89.83	28.68%
	3	69.13	90.51	30.93%
	4	68.59	90.75	32.31%
	5	68.77	91.08	32.44%
Average value	–	69.29	90.32	30.35%
Time (s)	1	290	192	−33.79%
	2	288	190	−34.03%
	3	294	188	−36.05%
	4	297	194	−34.68%
	5	289	183	−36.68%
Average value	–	291.6	189.4	−35.05%

detection rate of the GA model is 69.29%, and the improved GA model has an average detection rate of 90.32%, an increase of 30.35%. The average testing time of the GA model is 291.6 s, while the improved GA model has an average testing time of 189.4 s, reducing time consumption by 33.05%. The improved GA model using PCC has significantly improved computational efficiency and accuracy. This proves that the improved GA Feature selection mechanism in view of PCC has better performance.

4.2 Performance Testing of Intrusion Detection Models

The performance of ID model affects the actual efficiency and effect. The 1D CNN-GRU model is coded using Python in this study, and the model is tested under the Windows operating system. Other parameters are set as shown in Table 2.

During the performance test of ID model, the fully connected neural network (FCNN) and recurrent neural network (FCNN) models are selected for this study. RNN, LSTM, and GRU are compared with the 1D CNN-GRU ID model designed in the study, using the data set UNSW-NB15. The accuracy results are presented in Figure 9.

Table 2 Experimental parameter setting table

Experimental Environment and Parameters	Environment and Parameter Settings
Operating system	Windows 10
CPU	Intel(R)Core (TM)-7-10700
RAM	16G
Keras	2.3.1
Tensor Flow	2.0
Learning rate	0.01
Epoch	100
Dropout	0.01
Optimizer	Adam
Activation function	Sigmoid

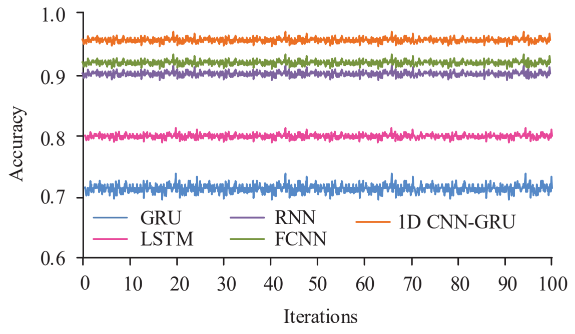


Figure 9 Accuracy test results of different models.

Figure 9 shows that all five models converge within 100 iterations, and the 1D CNN-GRU model in view of the improved GA achieved the optimal accuracy value of 95.0%. FCNN, RNN, LSTM, and GRU achieved accuracy values of 92.2%, 90.2%, 79.9%, and 71.3%, respectively. Therefore, the 1D CNN-GRU model in view of improved GA has the best performance in ID and can detect network intrusions with high accuracy. It compares the Loss function of the five models, as shown in Figure 10.

Figure 10 shows that the value of Loss function of 1D CNN-GRU model in view of improved GA is the smallest, 0.15, and the value of Loss function of FCNN, RNN, LSTM, and GRU models is 0.17, 0.19, 0.42, and 0.47 respectively. Therefore, the 1D CNN-GRU model in view of improved GA has better robustness and performance in ID. For testing the detection effect after lightweight, the pre lightweight model and the post lightweight model are compared for detection, and accuracy, detection rate, and time are used as detection indicators. The results are shown in Figure 11.

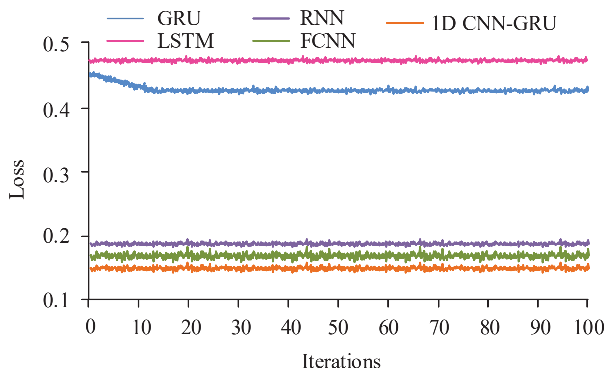


Figure 10 Loss values for five models.

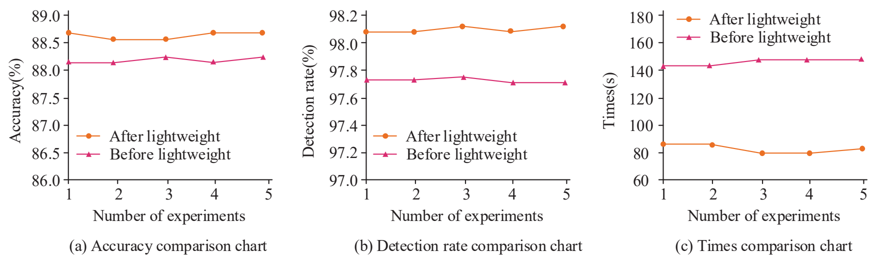


Figure 11 Test results before and after lightweight.

Figure 11(a) shows that the AA of model testing after lightweight is 88.6%, while the accuracy before lightweight is 88.1%. In Figure 11(b), the average detection rate of the lightweight model test is 98.12%, while the pre lightweight model test is 97.75%. In Figure 11(c), the average test time of the model after lightweight is 82 s, while the time before lightweight is 142 s. It demonstrates that the lightweight model reduces the number of parameters and significantly reduces the detection time; While reducing the computational burden, it also improves the detection accuracy of the model.

5 Conclusion

This study improves population initialization in GA in view of PCC in order to accurately detect attackers in the Internet of Things. In 1D CNN, the GRU model is introduced and a 1D CNN GRU model is constructed in view of an improved GA. The results showcased that the AA of the GA model reached 69.25%, and the AA of the improved GA model reached 79.55%, an improvement of 14.87%. The average detection rate of the GA model was 69.29%, and the improved GA model achieved an average detection rate of 90.32%, an increase of 30.35%. The average testing time of the GA model was 291.6s, while the improved GA model had an average testing time of 189.4s, reducing time consumption by 33.05%. The improved GA model using PCC had significantly improved computational efficiency and accuracy. The 1D CNN-GRU model in view of improved GA achieved an accuracy value of 95.0%, while FCNN, RNN, LSTM, and GRU achieved accuracy values of 92.2%, 90.2%, 79.9%, and 71.3%, respectively. The Loss function value of 1D CNN-GRU model in view of improved GA was 0.15, and the Loss function value of FCNN, RNN, LSTM, and GRU models was 0.17, 0.19, 0.42, and 0.47 respectively. Therefore, the 1D CNN-GRU model in view of improved GA had better robustness and performance in ID. The AA of the lightweight model testing was 88.6%, the average detection rate was 98.12%, and the average testing time is 82s. The accuracy before lightweight was 88.1%, the detection rate before lightweight was 97.75%, and the time before lightweight was 142s. Therefore, the lightweight 1D CNN-GRU model diminishes the quantity of parameters and significantly enhances detection efficiency. There are shortcomings in this study, as the amount of data in the dataset is insufficient. In future research, more datasets should be selected for testing.

References

[1] Ji J, Zhu X, Ma H. Apple Fruit Recognition Based on a Deep Learning Algorithm Using an Improved Lightweight Network. Applied Engineering in Agriculture, 2021, 37(1):123–134.

[2] Chen S, Liu Y, Lin C. Lightweight Verifiable Group Authentication Scheme for the Internet of Things. Acta Electronica Sinica, 2022, 50(04):990–1001.

[3] Miyanaji R S, Jabbehdari S, Modiri N. Continuous lightweight authentication according group priority and key agreement for Internet of Things. Transactions on Emerging Telecommunications Technologies, 2022, 7(33):4479–4504.

[4] Algarni F. A lightweight cryptography (LWC) framework to secure memory heap in Internet of Things. Alexandria Engineering Journal, 2021, 60(1):1489–1497.

[5] Krishnakumar P. Lightweight Cryptography and its Algorithms in Internet of Things:An Overview. International Journal of Innovative

Research in Science Engineering and Technology, 2021, 10(5): 4900–4904.

[6] Dahiphale V, Bansod G, Zambare A. Design and implementation of various datapath architectures for the ANU lightweight cipher on an FPGA. Frontiers of Information Technology & Electronic Engineering, 2020, 21(4):615–628.

[7] Ran Z X. A Step-Based Deep Learning Approach for Network Intrusion Detection. Computer Modeling in Engineering and Science, 2021, 128(3):1231–1245.

[8] Azizan A H, Mostafa S A, Mustapha A. A Machine Learning Approach for Improving the Performance of Network Intrusion Detection Systems. Annals of Emerging Technologies in Computing, 2021, 5(5):201–208.

[9] Luo X. Model design artificial intelligence and research of adaptive network intrusion detection and defense system using fuzzy logic. Journal of Intelligent and Fuzzy Systems, 2021, 40(3):1–9.

[10] Li X, Yi P, Wei W. LNNLS-KH: A Feature Selection Method for Network Intrusion Detection. Security and Communication Networks, 2021, 2021(3):1–22.

[11] Zeng S, Huang Z X, Jiang H. From Waste to Wealth: A Lightweight and Flexible Leather Solid Waste/Polyvinyl Alcohol/Silver Paper for Highly Efficient Electromagnetic Interference Shielding. ACS Applied Materials & Interfaces, 2020, 12(46):52038–52049.

[12] Li C, Zhou P. Improved Faster RCNN Object Detection. World Scientific Research Journal, 2020, 6(3):74–81.

[13] Zhang T L, Guo J. Oil spill detection method for SAR images based on the improved Faster R-CNN model. Marine Sciences, 2021, 45(5): 103–112.

[14] Haj-Manouchehri A, Mohammadi H M. Polyp detection using CNNs in colonoscopy video. IET Computer Vision, 2020, 14(5):241–247.

[15] Rauf U, Mohsen F, Wei Z. A taxonomic classification of insider threats: Existing techniques, future directions & recommendations. Journal of Cyber Security and Mobility, 2023, 12(2): 221–252.

[16] Dharma F, Shabrina S, Noviana A. Prediction of Indonesian inflation rate using regression model based on genetic algorithms. Jurnal Online Informatika, 2020, 5(1): 45–52.

[17] Sohail A. Genetic algorithms in the fields of artificial intelligence and data sciences. Annals of Data Science, 2023, 10(4): 1007–1018.

[18] Albahrani E A, Alshekly T K, Lafta S H. A review on audio encryption algorithms using chaos maps-based techniques. Journal of Cyber Security and Mobility, 2022, 11(1): 53–82.

[19] Falch M, Olesen H, Skouby K E, Tadayoni R, Williams I. Cybersecurity Strategies for SMEs in the Nordic Baltic Region. Journal of Cyber Security and Mobility, 2023, 11(6): 727–754.

[20] Chen Z. Research on internet security situation awareness prediction technology based on improved RBF neural network algorithm. Journal of Computational and Cognitive Engineering, 2022, 1(3): 103–108.

Biography

Li He obtained her Master's degree in Computer Technology (2011) from Wuhan University. Presently, she is working as a Vocational Teacher in the Department of Data and Information, Changjiang Polytechnic, Wuhan. She has published articles in more than 20 academic papers in important domestic journals. She has written two provincial planning textbooks and presided over and participated in eight provincial scientific research projects. She guides students to participate in competitions and win many awards. Her areas of interest include image processing, pattern recognition and information security.