CIT. Journal of Computing and Information Technology, Vol. 31, No. 2, June 2023, 91–106
doi: 10.20532/cit.2023.1005737

A Network Intrusion Detection Model Based on GA-Improved NSA

Long Li

Department of information engineering, Xuchang Electrical Vocational College, Xuchang, Henan, China

With the popularization of the Internet, network security issues have also emerged. In response to network security issues, there are certain shortcomings in current network intrusion detection technologies. To improve and optimize this technology, a network intrusion detection model based on genetic algorithm and improved negative selection algorithm is designed. The generation of detectors in the selection algorithm is replaced by genetic algorithm, and the non-self spatial distribution of detectors is optimized. This paper proposes a network intrusion detection model using a genetic algorithm-improved negative selection algorithm (GA-INSA) and an improved LeNet-5 CNN. The GA enhances detector generation and distribution in NSA while SMOTE handles class imbalance for CNN. Experiments show GA-INSA has over 9% higher accuracy than NSA, SVM and GA-BP across different data sizes. The improved LeNet-5 demonstrates superior accuracy and recall rates by over 20% over the baseline LeNet-5. However, more comprehensive evaluation on public datasets, design details on architectures, and discussion around limitations are warranted. The data shows that the designed network intrusion detection model has better performance. The model can provide technical support for network intrusion detection in reality and can enrich the content of network intrusion detection technology.

ACM CCS (2012) Classification: Computing methodologies → Machine learning → Machine learning algorithms

Keywords: Negative selection algorithm, Genetic algorithm, Network intrusion, Detection, LeNet-5

1. Introduction

In modern society, the Internet is ubiquitous and exists in all aspects of people's lives. However, behind the convenience of the internet, there are many network security issues hidden, which also pose a significant threat to human lives and social stability [1–2]. To address the complete network problem, network intrusion detection (NID) technology has emerged and has been developed for over 40 years now. However, the traditional NID technology also has its own shortcomings. For example, if the system performance is not strong enough, it is difficult to keep up with the update speed of network intrusions, and it is difficult to classify attack types with low data volume [3]. Many scholars are also conducting research and attempting to solve these problems. Researchers such as N. Tran *et al.* analyzed different techniques for handling imbalanced data, aiming to find the best technique for it [4]. With the development of technology, deep learning and artificial immune systems' algorithms have gradually been applied to NID and become hotspots in this field [5]. The main problem in finding solutions for network intrusion is improving the accuracy of detection models and minimizing false alarms. However, NID technology based on deep learning and artificial immune systems' algorithms also has certain shortcomings. For example, with poor flexibility, it is easy to miss alarms and detect when facing large amounts of data, and the detection rate is lower when the distribution of attack types is uneven and the data volume is small. In addition, different algorithms also have the disadvantages of low detection rate and high false alarm rate (FAR) when detecting network intrusions. The artificial immune systems' negative selection algorithm is prone to defects such as detector coverage and high redundancy when detecting network intrusions. This study proposes a NID model based on a genetic algorithm (GA) to improve the negative

selection algorithm (NSA). The paper aims to replace detector generation with GA and optimize the non-self generating spatial distribution of the detector. The study also designs a NID model based on improved LeNet-5 and uses it as an effective supplement to the GA-improved NSA model. The purpose is to improve the efficiency of network intrusion detection, solve the detector problem of artificial immune systems' NSA, and avoid uneven distribution of attack types. There are two innovative points. The first is to use GA to improve NSA. The second is to improve LeNet-5 by using Synthetic Minority Oversampling Technique (SMOTE). This study aims to improve NID by introducing new technologies. The study consists of four parts. The first part is an overview of literature related to NID. The second part is the specific design process of the proposed method. The third part is the result analysis and performance verification of the designed model. The fourth part is the conclusion.

2. Related Works

The development and popularization of the Internet have brought many conveniences to people's lives, but also brought many risks, such as information leakage and illegal intrusion. To avoid losses caused by illegal intrusion, numerous scholars have deeply studied NID. X. Kan *et al.* proposed an intrusion detection method based on adaptive particle swarm optimization (PSO) convolutional neural network (CNN) to improve the accuracy of NID. The structural parameters of CNN were optimized using the PSO algorithm. In addition, the study also introduced a new evaluation method that involves probability prediction and labeling. To verify the effectiveness of the method, multiple evaluation indicators and multiple experiments were used in this study. The proposed NID method was effective and had certain advantages [6]. Experts such as H. Qiu *et al.* proposed a new adversarial attack to protect deep learning intrusion detection models. This study replicated the black box model through data extraction and used saliency maps to illustrate the impact of packet attributes on detection results, quickly generating adversarial examples. It showed that when malicious data packets were modified by less than 0.049% bytes, this method could achieve a success rate of 94.27% [7]. To reduce the computational complexity of NID systems, scholars such as M. N. Injadat *et al.* proposed a multi-level optimization based maximum likelihood NID system framework. This study determined the size of training samples and validates the effectiveness of the proposed framework through multiple datasets. The number of training samples and feature sets of this framework had been significantly reduced, and the detection accuracy could reach 98% [8]. To better protect network security, Y. He *et al.* designed an intrusion detection algorithm that combines deep neural networks with CNN, and they tested the activation function and parameters of the algorithm. When using the corrected linear unit activation function, the proposed algorithm had the best recognition performance [9].

To solve the problem of high computational complexity in NID models in wireless sensors, scholars such as R. H. Dong *et al.* had designed an intrusion detection model that integrated the information gain rate and bagging algorithm. This model selected the features of node traffic data in sensors and designed an integrated classifier to train the optimized decision tree. Compared with existing baseline methods, the proposed method had a higher detection accuracy [10]. M. Wei *et al.* constructed a secure network framework to address the network attacks faced by wireless sensors and detected network traffic data through normal profiles. Meanwhile, the study also constructed a testing platform, which can be applied to most wireless sensor networks with low false positive rate and high accuracy [11]. X. Zhou *et al.* designed a hierarchical adversarial attack generation method to respond to unknown types of attacks. In addition, it also constructed an intelligent mechanism based on saliency graph technology and proposed a hierarchical node selection algorithm based on restart random walk. Finally, a comparison between the proposed method and the existing four baseline methods showed that this method had better performance [12]. H. Jagruthi *et al.* designed a method that combines features and bidirectional recursive CNN to improve the NID efficiency. This method involved CNN and bi-

directional long-term and short-term memory, with average accuracy of 97.98% and 91.45% in binary and multi-class classification, respectively, showing good superiority [13].

In summary, there are currently many studies on NID, involving a variety of algorithms. However, these studies also have certain shortcomings, such as insufficient flexibility in intrusion detection, insufficient system performance, and uneven distribution of attack types. Based on these issues, this study innovatively improves the current popular NSA and LeNet-5 structure. A NID model based on GA-improved NSA (GA-INSA) and an improved LeNet-5 NID model have been designed to enhance and optimize NID technology, *i.e.* GA-INSA NID model and iLeNet-5 NID model.

3. Research Model

In response to the improvement of NID technology, this study designed a GA-INSA NID model. The GA was used to replace the generation of the detector and the non-self spatial distribution of the detector was optimized. In addition, in response to the shortcomings of the GA-INSA NID model when dealing with datasets with multiple feature attributes, the study also designed an iLeNet-5 NID model [14]. By using SMOTE to process imbalanced datasets, it was used as an effective supplement to the GA-INSA model.

3.1. Design of NID Model Based on Improved NSA

To improve the NID technology, this study has made improvements to the currently popular NSA and LeNet-5 structures. Firstly, the study aims to address the issues of existing NSAs on detectors by improving the NSA. There is an issue of uneven distribution of attack types, and the improved NSA has shortcomings in handling abundant feature attribute datasets. A NID model based on synthetic minority class oversampling technique is designed to address these issues, and improvements are made to the LeNet-5. It is also used as an effective supplement to the GA-INSA detection model. The NSA is one of the classic artificial immune systems' algorithms and has been extensively applied in NID [15]. The key role of NSA in intrusion detection is to generate detectors, while also ensuring that it does not mistakenly recognize its own data [16]. The advantage of NSA is that it can generate infinite non-self detectors through a small amount of non-self data. Figure 1 outlines the flowchart of the traditional NSA.

In Figure 1, the process of the NSA is segmented into two modules, *i.e.* the detector generation and the anomaly detection modules. The first step of the NSA is to define the self set, the second step is to randomly generate a detector, and the third step is to match the detector with the self set. If the matching result is yes, then the detector is deleted. Otherwise, the detector in the mature detector set is displaced. The fourth step is to match the mature detector with the data to be tested. If the matching result is yes, it is determined that the detector is an abnormal sample, otherwise it is considered a normal sample [17]. However, NSA also has certain problems, such as the uneven distribution of detectors in non-self space, known as the "black hole" problem [18]. Based on this issue, this study uses GA to improve NSA. The specific improvement measure is to replace the generation of detectors with GA and optimize the non-self spatial distribution of detectors. Afterwards, NSA is used to perform self tolerance of the detector and reduce the dimensionality of data features. When generating a detector, the total genetic algebra will be set first, and then the detector will be generated. The generation of detectors mainly consists of six steps, namely encoding, forming the initial population, determining the fitness function, selection, crossover, and mutation. The generation of the detector will not proceed to the sixth step until the initial set maximum genetic total number is reached. The mutation operations will be used to improve the local search performance of the algorithm. In addition, in the selection process, the study uses the roulette wheel selection method to select individuals. The process of a GA-INSA NID model is shown in Figure 2.

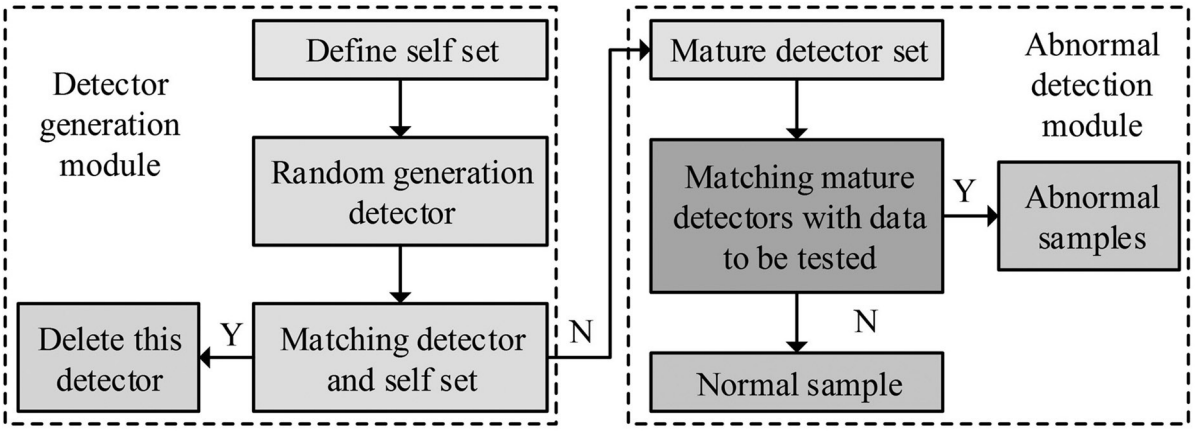


Figure 1. The process of traditional NSA.

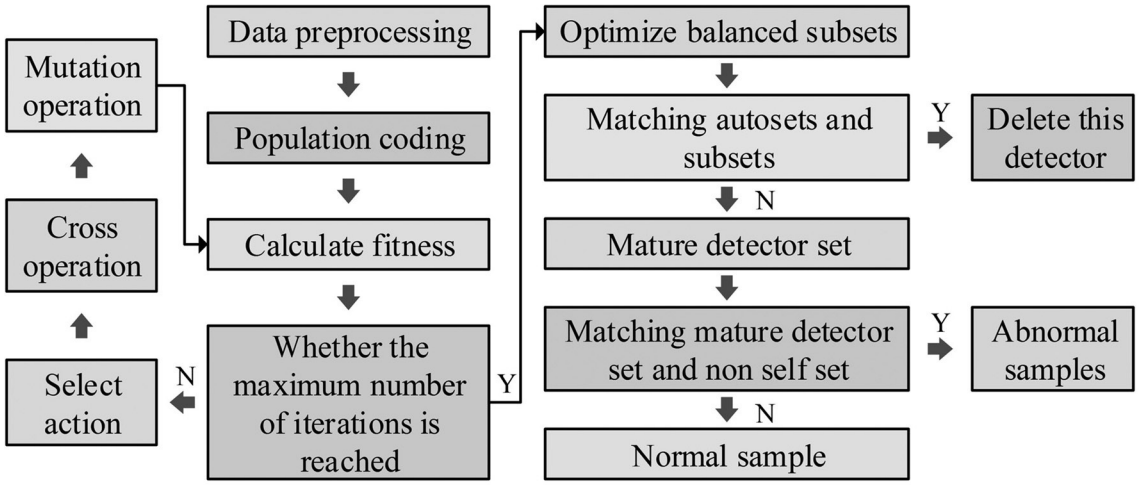


Figure 2. The flow of GA-INSA NID model.

In Figure 2, the first step of the GA-INSA NID model is data preprocessing, the second step is population encoding, and the third step is fitness calculation. The fourth step is to determine whether the number of iterations has reached its maximum [19]. If the result is yes, the algorithm proceeds to the next step. Otherwise, it is required to perform the selection, crossover, and mutation operations in sequence, and then return to the third step. The fifth step is to optimize and balance the subsets. The sixth step is to match the self set with the optimized balanced subset. If the matching result is yes, the detector is deleted, otherwise, the detector is placed into the mature detector set. The seventh step is to match the mature detector set with the non self set. If the matching result is yes, it is determined that the data is abnormal, *i.e.* it represents intrusion data, otherwise it represents normal data. Due to the use of the NSL-KDD dataset in the study, which has the characteristics of multiple feature attributes and is relatively complex overall [20], data processing is conducted to facilitate the operation of the model. There are three main steps in data preprocessing, with the first step being normalization. The specific normalization formula is equation (1).

$$\partial' = \frac{\partial - \min_c}{\max_c - \min_c} \tag{1}$$

In equation (1), ∂ represents the data being processed. c is the serial number of the record attribute. $\min_c$ and $\max_c$ are the min and max values of each record attribute, respectively. Step

2 is to digitize the character data, and step 3 is to use Principal Component Analysis (PCA) to reduce the dimensionality of the dataset. The main process of PCA is shown in Figure 3.

In Figure 3, the first step of PCA is to standardize the dataset. The second step is to obtain the feature vectors and eigenvalues. The third step is to sort the feature values in descending order and select the feature vectors. The fourth step is to construct the projection matrix. The fifth step is to generate feature sub-spaces. Among them, the data in the second step is derived from the covariance matrix, and the calculation of this matrix is equation (2).

$$CM = \frac{1}{n-1}\left(\left(x-x'\right)^{T}\left(x-x'\right)\right) \tag{2}$$

In equation (2), x' represents the mean vector. n represents the quantity of samples, and x is a random variable. The solution of the mean vector is equation (3).

$$x' = \frac{1}{n}\sum_{k=1}^{n} x_i \tag{3}$$

In equation (3), k represents the dimension of the generated feature space. x_i represents the i-th random variable. The covariance calculation between two features is equation (4).

$$C_{vjk} = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_{ij}-x_j'\right)\left(x_{ik}-x_k'\right) \tag{4}$$

In equation (4), x_{ij} and x_{ik} are the x_j and x_k variable values of the i-th sample point. x_j' represents the average value of the x_j variable. x_k' represents the average value of the x_k variable. The generated feature subspace is equation (5).

$$Y = x * W \tag{5}$$

In equation (5), W represents the projection matrix. The advantages of GA are high coverage and low consumption, which can effectively solve the problem of NSA not being able to fully cover abnormal samples. The calculation of fitness is equation (6).

$$Fit\left(f\left(x\right)\right) = \begin{cases} 1-0.5x\left[\left|\dfrac{f(x)-b}{a}\right|\right]^{\alpha}, & \left|f(x)-b\right| < a \\ \dfrac{1}{1+\left[\left|\dfrac{f(x)-b}{a}\right|^{\beta}\right]}, & \left|f(x)-b\right| \geq a \end{cases} \tag{6}$$

In equation (6), $f(x)$ is the objective function. a and b are both parameters. α and β are both custom parameters. a and b can follow the GA for continuous correction. Matching a mature detector set with a non self set requires the use of Euclidean distance, and the distance between the sample and detector is calculated as shown in equation (7).

$$L = \sqrt{\sum_{i=1}^{n}\left(Q_{\Gamma}-Z_{\Gamma}\right)^{2}} \tag{7}$$

In equation (7), Z_{Γ} represents the Γ-th detector. Q_{Γ} is the Γ-th sample.

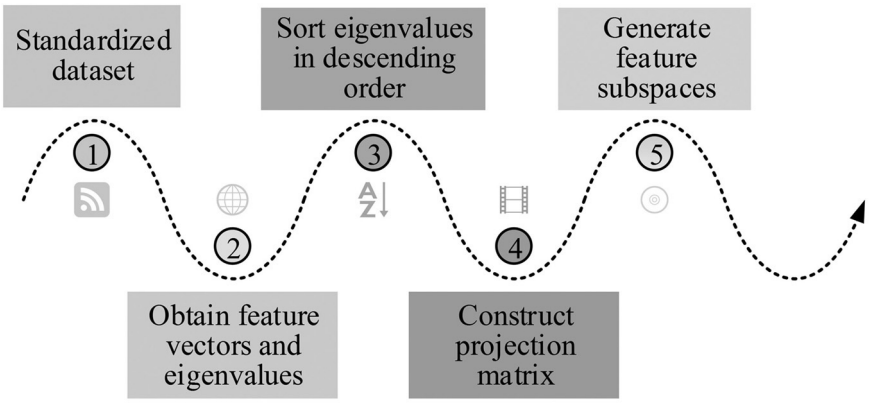


Figure 3. The main process of principal component analysis.

3.2. Design of NID Model Based on Improved LeNet-5

In the previous chapter, an intrusion detection model based on GA-improved NSA was studied and designed. The GA was used to replace the generation of detectors, and the non-self spatial distribution of detectors was optimized. When there are many feature attributes recorded in the dataset and the overall dataset is complex, directly using a GA-INSA NID model may result in incorrect test results. To address this issue, in addition to using PCA to reduce the dimensionality of the dataset, CNN can also be used for processing [21]. The advantage of CNN is that it can reduce computational complexity, reduce the difficulty of preprocessing high-dimensional data, and quickly learn different data features based on unlabeled data [22]. When facing a large amount of intrusion detection data, CNN has obvious advantages. LeNet-5 belongs to a type of CNN. It has made some progress in NID, but there is also a shortage of low recall rate on small sample data. Therefore, to avoid the shortcomings of the GA-INSA model, LeNet-5 was improved and used as an effective supplement to the GA-INSA model. The calculation of input features in CNN is equation (8).

$$x_j^i = f\left(\sum_{i\in\rho_j} x_i^{l-1} * k_{ij}^l + b_j^l\right) \tag{8}$$

In equation (8), ρ_j represents the local receptive field. x_i^{l-1} represents the feature map output on layer $l-1$. k_{ij}^l represents the convolutional kernel on layer l. b_j^l is the offset amount on the j channel of the l layer. f represents the activation function. j represents the channel. The pooling of input features is equation (9).

$$x_j^l = f\left[\omega_j^l D\left(x_j^{l-1}\right) + b_j^l\right] \tag{9}$$

In equation (9), D represents down sampling. ω_j^l represents the weight on the j channel. x_j^{l-1} is the output characteristics of the j-th channel on layer $l-1$ [23]. The common processing methods for the pooling layer include maximum pooling sampling and evaluation pooling sampling. The output of the fully connected layer (FCL) is shown in equation (10).

$$x^l = f\left(\omega^l x^{l-1} + b^l\right) \tag{10}$$

In equation (10), x^{l-1} is the input. ω^l represents weight. b^l represents the offset amount. There are three common activation functions, namely sigmoid, hyperbolic tangent (Tanh), and corrected linear unit (ReLU) [24]. The LeNet-5 model includes an input layer, convolutional layer, pooling layer, FCL, and output layer. The convolutional and pooling layers are two. The specific structure of the NID model based on LeNet-5 is shown in Figure 4.

In Figure 4, the first step of the LeNet-5 based intrusion detection model is to normalize the data. The second step is to standardize the CNN input layer. The content of data normalization involves the digitization and standardization of symbol features. The third step is to alternately perform convolution and pooling, where convolution needs to be performed 3 times and pooling needs to be performed 2 times. The fourth step is to transfer the data to the output layer. Step 5 is to transfer the data to the Softmax function. The sixth step is to output the intrusion detection results. When the number and distribution of samples exhibit an uneven pattern, the dataset is called an imbalanced dataset. This will also affect the accuracy of the experimental results. The calculation of dataset imbalance is given in equation (11).

$$IR = \frac{U}{\Pi} \tag{11}$$

In equation (11), U is the majority class sample, and Π means the minority class sample. To process imbalanced datasets, the study adopted the SMOTE algorithm to enhance minority class samples and optimized the network through CNN, forming a iLeNet-5 NID model. The specific model structure is shown in Figure 5.

In Figure 5, the model is mainly divided into three sections: data preprocessing section, model training section, and intrusion detection section. The first step of this model is to use the SMOTE algorithm to process the original dataset. The second step is to digitize and normalize symbol features and form a test set. The third step is to use the training set to train the NID model, and then output the predicted values. The fourth step is to compare the predicted values with the true values of the training set. The fifth step is to determine the loss value. When the loss value is small, the mode proceeds to the

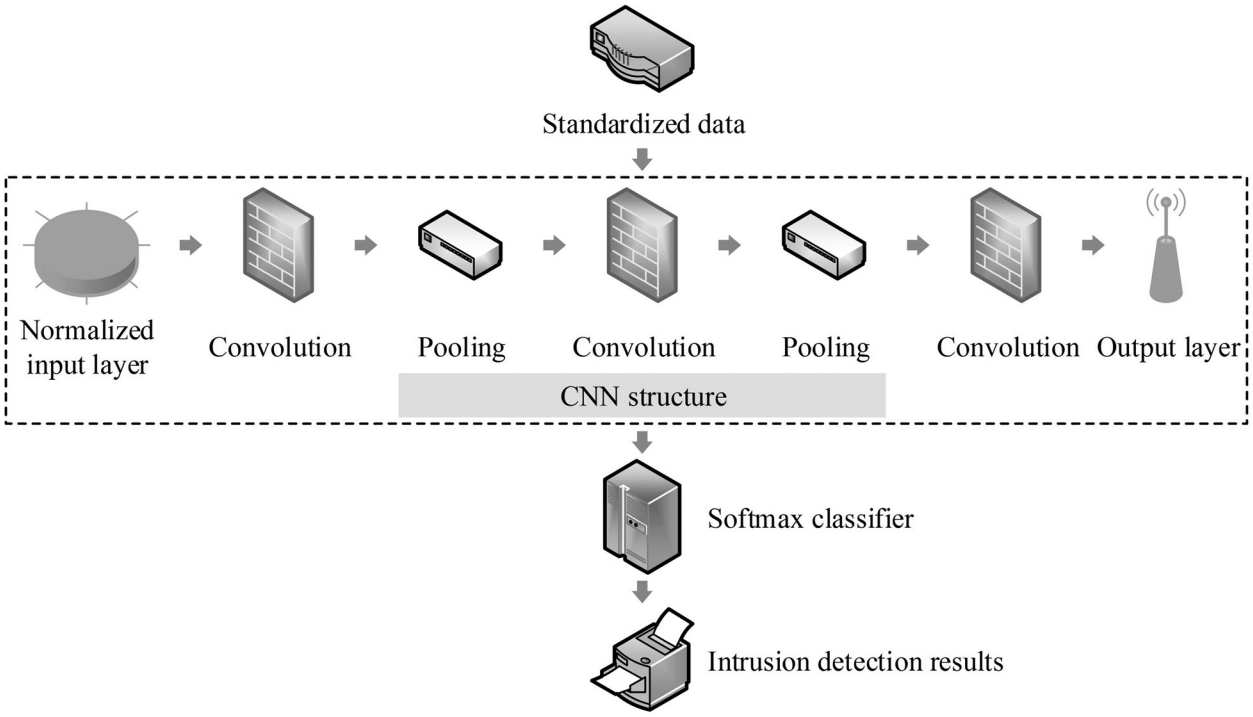


Figure 4. The specific structure of NID model based on LeNet-5.

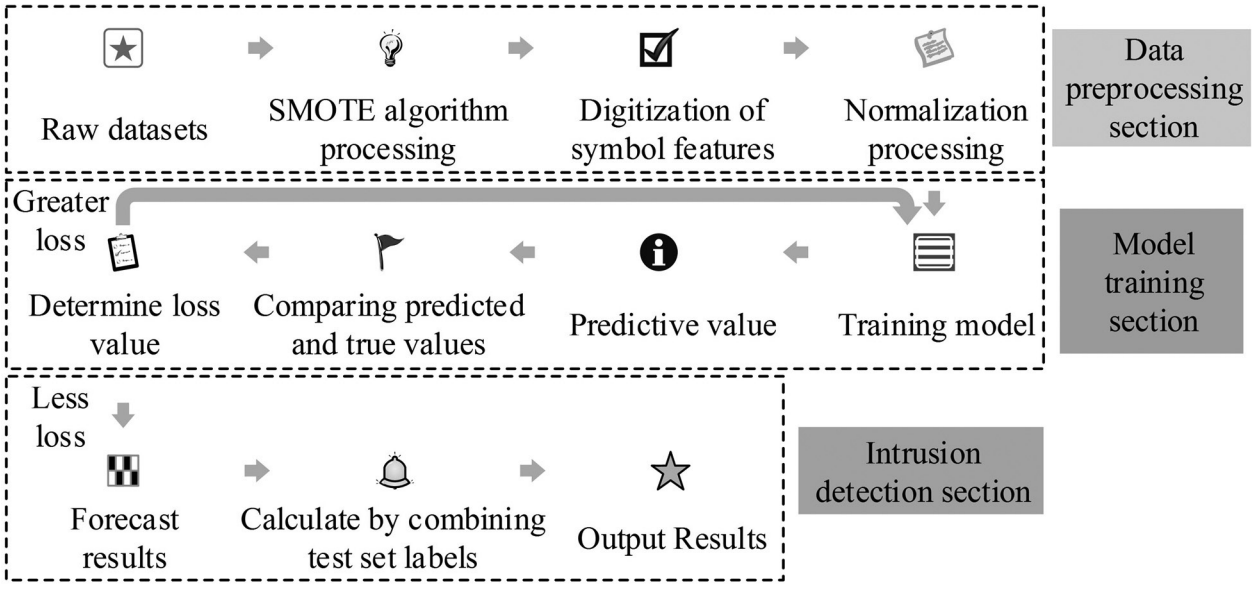


Figure 5. The iLeNet-5 NID model.

next step, otherwise ti proceeds to the third step. The sixth step is to use an intrusion detection model to predict the results and combine them with test set labels for calculation. The seventh step is to output the results. The specific presentation of the SMOTE algorithm involved in the data preprocessing section is shown in Figure 6.

In Figure 6, the green circle represents minority class data. x_δ represents a minority sample point. $\hat{x}_\delta$ represents the nearest sample point. The nearest neighbor sample points are located around the minority sample points. The purple triangle x_{new} is the newly generated sample, and its calculation method is given in equation (12).

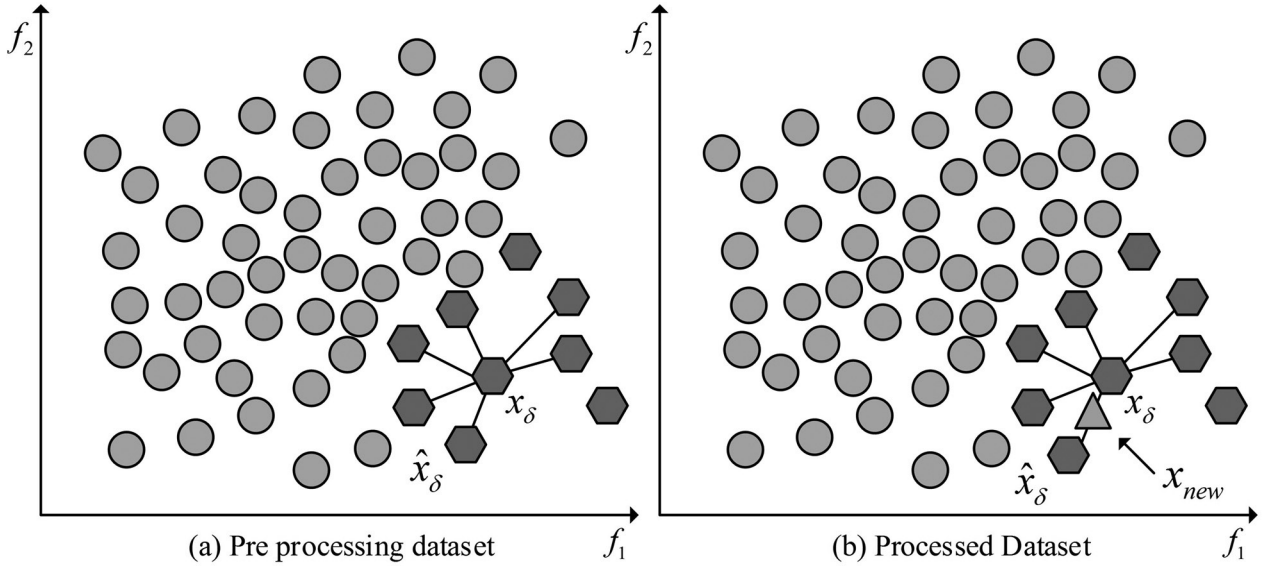


Figure 6. The specific expression of SMOTE algorithm.

$$x_{new} = x_\delta + rand(0, 1) \times (\hat{x}_\delta - x_\delta) \tag{12}$$

In equation (12), *rand*(0, 1) represents a randomly selected number within the range of 0 to 1. The iLeNet-5 NID model replaces the 1 * 1 convolutional layer with a FCL. In addition, the third, fourth, and sixth convolutional layers are 3 * 3 convolutional layers, and batch normalization (BN) has been introduced in these layers to enhance the network's generalization ability. The BN algorithm has four steps. The average value of a small portion of the data in the dataset is expressed as equation (13).

$$\mu_\beta = \frac{1}{m}\sum_{\delta=1}^{m} x_\delta \tag{13}$$

In equation (13), m represents the amount of small pieces of data. δ is the serial number of the data. The variance calculation of some data is equation (14).

$$\sigma_\beta^2 = \frac{1}{m}\sum_{\delta=1}^{m}(x_\delta - \mu_\beta)^2 \tag{14}$$

After preprocessing the batch of data and placing it between [0, 1], the nearest neighbor sample points can be obtained, as shown in equation (15).

$$\hat{x}_\delta = \frac{x_\delta - \mu_\beta}{\sqrt{\sigma_\beta^2 + \varepsilon}} \tag{15}$$

In equation (15), ε represents a very small positive value. When accessing the pooling layer after the fourth layer, the maximum pooling structure needs to be used. Then, the data is mapped into one-dimensional data through full connectivity and classified through softmax classification. Finally, the network intrusion type is output.

4. Result and Discussion

For the performance verification of the improved NSA network intrusion model, different evaluation indicators were selected, and different comparative algorithms were examined. In addition, this study also set the data volume for the subset of data required for the experiment. For the verification of the performance of the LeNet-5 model, different evaluation indicators and comparison objects were selected, and the experimental environment and parameters were set.

4.1. Analysis of NID Model Results Based on GA Improved NSA

To verify the performance of the GA-INSA NID model, two evaluation indicators were selected, namely accuracy and FAR. Accuracy and FAR are important indicators for evaluating the performance of network intrusion models. Three

other algorithms were selected for comparison, namely the traditional NSA, support vector machine (SVM), and GA back propagation (GA-BP). The experiment used four subsets of data from NSL-KDD, with corresponding data volumes of 60000, 12000, 18000, and 24000, respectively. The operating system used in the verification experiment is Windows 10 (64 bit), with an Intel Core i7 9750H CPU, an Intel Supercore Graphics Card 630 integrated graphics card, and a maximum memory of 128 GB. The mature detector count is 1100, with 220 genetic iterations and probabilities of crossover and mutation equal to 0.9 and 0.09, respectively. Due to the influence of the self radius on the performance of the detector, the accuracy and other indicators under different radii were compared in the study. The comparison of changes in different evaluation indicators under different self radii is shown in Figure 7.

In Figure 7 (a), as the radius of the self gradually increases, the trend of the accuracy index is to first increase and then decrease, with the corresponding radius of the self being 0.5 when decreasing. In Figure 7 (b), FAR decreases with the increase of self radius. When the self radius is less than 0.4, the decrease of FAR is faster. When the radius of the self is greater than 0.4, the decrease of FAR significantly slows down. In Figure 7 (c), as the self radius increases, the detection rate continuously decreases. When the self radius is less than 0.6, the decrease rate of detection rate is relatively slow. When the self radius is greater than 0.6, the decrease rate of detection rate begins to accelerate. To validate the selected public dataset, a comparison was made among commonly used NID datasets, and the comparison results are shown in Table 1.

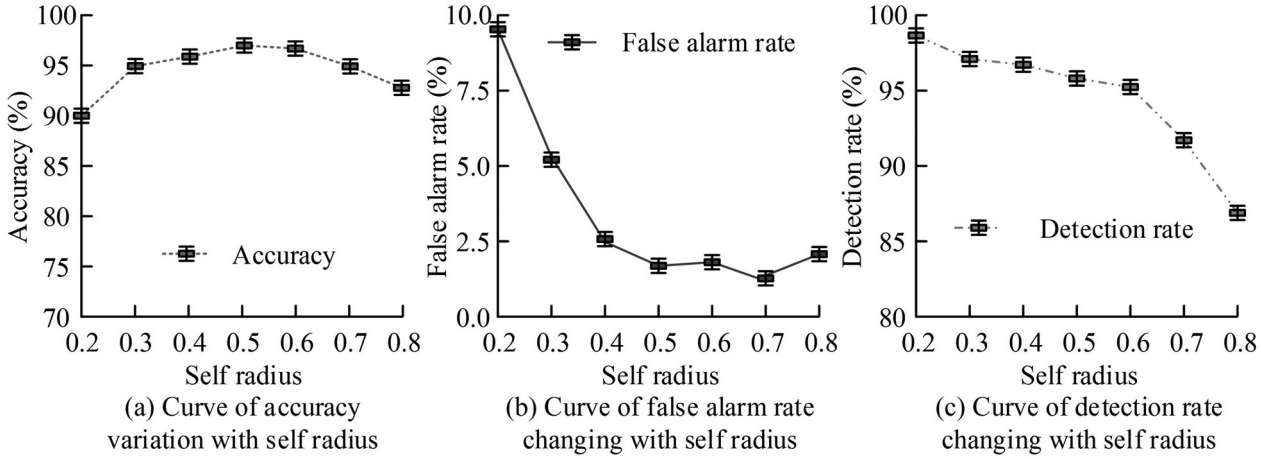


Figure 7. Comparison of changes in different evaluation indicators under different self radii.

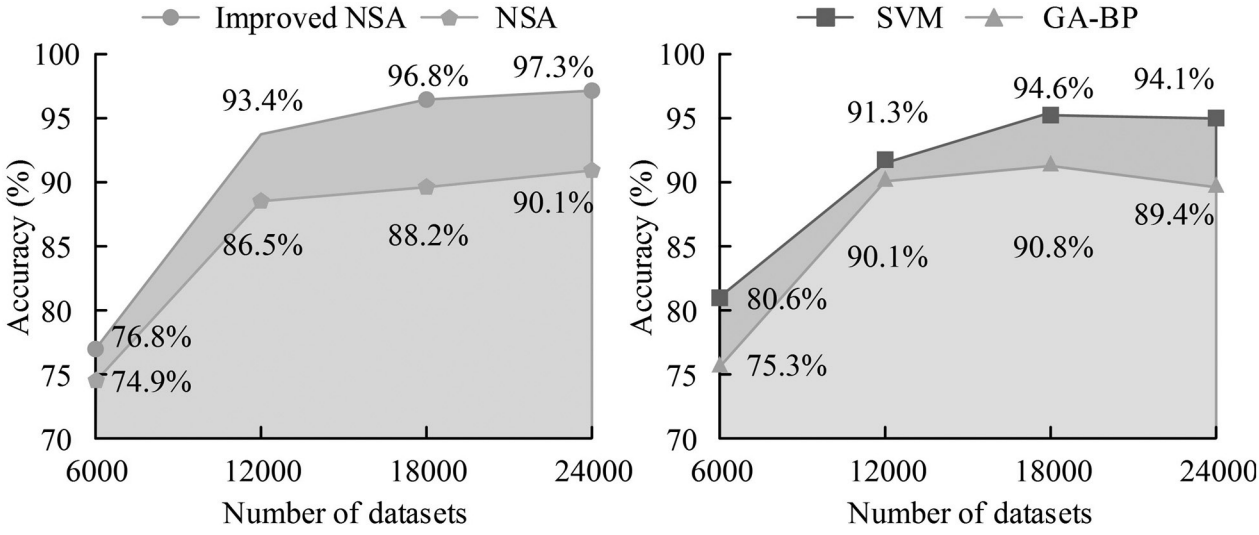
Table 1. Comparison of different network intrusion detection datasets.

Dataset name	Types of cyber attacks
MIT LL DARPA	Denial-of-service (Dos), Remote to Local (R2L), User to Root (U2R), surveillance or Probe (Probe) and date compromise (data)
CIC-IDS-2017 and CIC-IDS-2018	File Transfer Protocol (FTP) Brute Force, Secure SHell (SSH) Brute Force, Dos, Heartbleed, Web Attack, Infiltration, Botnet and Distributed Denial of service (DDos)
NSL-KDD and KDD99	Dos, R2L, U2R, Probe, Normal
UNSW-NB15	Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

From Table 1, the NSL-KDD dataset mainly involves five types of attacks, namely Dos, R2L, U2R, Probe, and Normal. The UNSW-NB15 dataset mainly contains nine types of attacks, namely Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. The CIC-IDS-2017 and CIC-IDS-2018 datasets mainly cover eight types of attacks, namely FTP Brute Force, SSH Brute Force, Dos, Heartbleed, Web Attack, Infiltration, and DDos. The MIT LL DARPA dataset mainly includes five types of attacks, namely Dos, R2L, U2R, Probe, and data. Therefore, the study mainly selected the NSL-KDD dataset and the UNSW-NB15 dataset. The accuracy comparison of different algorithm models under different data subsets is given in Figure 8.

In Figure 8 (a), when the data volume is 6000, 12000, 18000, 24000, the accuracy of improved NSA is 76.8%, 93.4%, 96.8%, 97.3%, and NSA is 74.9%, 86.5%, 88.2%, and 90.1%. In Figure 8 (b), when the data volume is 6000, 12000, 18000, 24000, the accuracy of SVM and GA-BP algorithms is 80.6% and 75.3%, 91.3% and 90.1%, 94.6% and 90.8%, 94.1% and 89.4%. Therefore, the GA-INSA model performs better, and the classification accuracy of the model is better. Figure 9 shows the comparison of FAR among different algorithm models under different data subsets.

In Figure 9 (a), FAR of the improved NSA and NSA are 8.21% and 9.98% respectively when the data volume is 6000, with 4.13% and 5.93% at 12000, 2.82% and 4.33% at 18000, and 1.32% and 1.93% at 24000. In Figure 9 (b), FAR of SVM and GA-BP are 6.37% and 7.12% for 6000 data, 5.04% and 5.75% for 12000 data, 3.98% and 4.11% for 18000 data, and 3.81% and 3.02% for 24000 data. Therefore, the model detection performance of GA-INSA model is better. To better validate the performance of the GA-INSA NID model, the response time of this model and the comparison model in the face of network attacks was analyzed. Response time is one of the important indicators reflecting the performance of the system itself. In addition, the study also selected additional NID models for comparative verification and selected the UNSW-NB15 dataset for testing. The newly added NID models include BiGRU-SVM combining bidirectional gate recurrent unit (BiGRU) and SVM, RF-XGBoost combining random forest and eXtreme gradient boosting (XGBoost), PCA-RNN combining PCA and recurrent neural network (RNN). A total of 5 response time tests were conducted. The operating system used for testing is Windows 10 (64 bit).



(a) Comparison of classification accuracy between improved NSA and NSA algorithms

(b) Comparison of classification accuracy between GA-BP and SVM algorithms

Figure 8. Comparison of accuracy of different algorithm models under different data subsets.

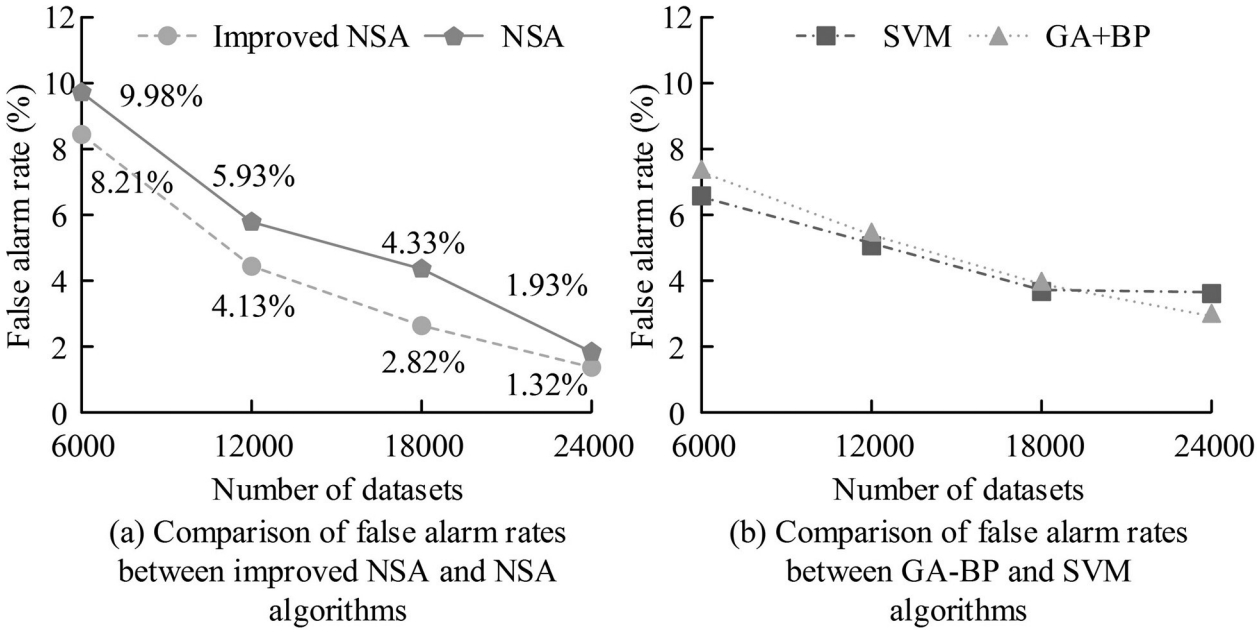


Figure 9. Comparison of FAR between different algorithm models under different data subsets.

The central processor is Intel Core i7 9750H. The integrated graphics card is an Intel ultra core graphics card 630. The maximum memory is 128 GB. Table 2 compares the response times of different intrusion detection models when facing network attacks.

In Table 2, the maximum response times of the improved NSA model, NSA model, SVM algorithm, and GA-BP algorithm are 60 ms, 125 ms, 96 ms, and 102 ms, while the minimum values are 45 ms, 109 ms, 73 ms, and 82 ms. The maximum response times of BiGRU-SVM, RF-XGBoost, and PCA-RNN detection models are 76 ms, 78 ms, and 79 ms, respectively, while the minimum values are 68 ms, 64 ms, and 62 ms, respectively. Therefore, the response time of the improved NSA model has always been better than other comparative detection models, which also indicates that the performance of this detection model is better.

Table 2. Comparison of response time of different intrusion detection models in the face of network attacks.

Model	**Number of experiment**				
	1	**2**	**3**	**4**	**5**
Improved NSA	60 ms	58 ms	45 ms	51 ms	55 ms
NSA	120 ms	111 ms	125 ms	117 ms	109 ms
SVM	88 ms	96 ms	73 ms	82 ms	77 ms
GA-BP	102 ms	98 ms	82 ms	94 ms	89 ms
BiGRU-SVM	70 ms	76 ms	68 ms	73 ms	69 ms
RF-XGBoost	78 ms	65 ms	71 ms	67 ms	64 ms
PCA-RNN	73 ms	79 ms	66 ms	70 ms	62 ms

4.2. Analysis of NID Model Results Based on Improved LeNet-5

To validate the performance of the iLeNet-5 NID model, two evaluation indicators were selected, namely accuracy and recall. The comparative model is an iLeNet-5 NID model. The experimental model is mainly installed on the Tensor Flow framework, and the training, testing, and validation set is implemented on the NVIDIA GTX 2080T GPU. The optimizer used in the experiment is Adam, the environment is Python 3.8, and epochs is set to 300. The operating system used in the experiment is also Windows 10 (64 bit). Table 3 shows the specific data.

In Table 3, the number of type data on the training set, Normal is 97278, Dos is 391458, Probe is 4107, Remote to Local attach (R2L) is 18016, and User to Root attach (U2R) is 10400. The total number of training set data is 521259. The proportions of the five types of data are 18.66%, 75.04%, 0.81%, 3.47%, and 2.01%, respectively. On the test set, the number of five types of data is 60593, 229853, 4166, 16189, and 228, respectively. The total number of test set data is 311584. The proportions of the five types of data are 19.48%, 73.80%, 1.37%, 5.23%, and 0.11%, respectively. The accuracy comparison of the LeNet-5 NID model before and after improvement is shown in Figure 10.

Table 3. Number of different data types in the dataset processed using SMOTE.

Data type	Number of training sets	Account for percentage	Number of test sets	Account for percentage
Normal	97278	18.66%	60593	19.48%
Dos	391458	75.04%	229853	73.80%
Probe	4107	0.81%	4166	1.37%
R2L	18016	3.47%	16189	5.23%
U2R	10400	2.01%	228	0.11%
Total quantity	521259	/	311584	/

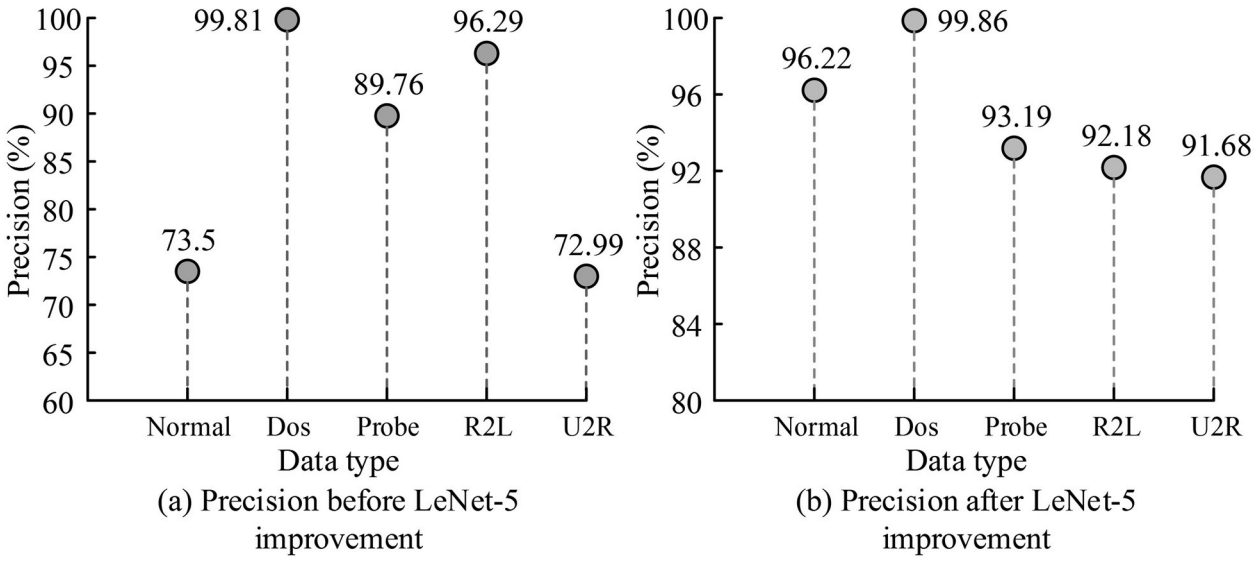


Figure 10. Comparison of accuracy of LeNet-5 NID Models before and after improvement.

In Figure 10 (a), the accuracy of the data type before improving LeNet-5 is 73.50% for Normal, 99.81% for Dos, 89.76% for Probes, 96.29% for R2L, and 72.99% for U2R. In Figure 10 (b), after improving LeNet-5, the accuracy rates of the five types of data are 96.22%, 99.86%, 93.19%, 92.18%, and 91.68%, respectively. The accuracy of the iLeNet-5 NID model on different types of data is significantly higher than before the improvement. Therefore, the improved LeNet-5 has better performance and better classification performance. The comparison of recall rates of the LeNet-5 NID model before and after improvement is shown in Figure 11.

In Figure 11 (a), the recall rate of the data type before improving LeNet-5 is 99.59% for Normal, 98.41% for Dos, 76.48% for Probe, 4.04% for R2L, and 11.2% for U2R. In Figure 11 (b), after improving LeNet-5, the recall rates of the five types of data are 98.57%, 99.94%, 90.93%, 84.91%, and 85.32%, respectively. The iLeNet-5 NID model has a higher recall rate on most data types than before, with the greatest improvement on R2L and U2R data types. Therefore, the performance of the iLeNet-5 NID model is significantly better than before. To further validate the model performance, the defense success rate (DSR) and FAR of the NID system were selected for comparison. DSR and FAR are key indicators reflecting the detection performance of intrusion detection systems. The DSR and FAR of the NID were tested a total of 5 times. The testing environment is consistent with the response time of the system. The comparison of DSR and FAR of the NID model before and after the improvement of LeNet-5 is shown in Figure 12.

In Figure 12 (a), the max-DSR of the LeNet-5 system before improvement is 82.5%, and the min value is 73.6%. The max-DSR of the improved LeNet-5 system is 98.7%, and the min value is 91.2%. In Figure 12 (b), the max-FAR of the LeNet-5 system before improvement is 5.87%, and the min value is 4.85%. The max-FAR of the improved LeNet-5 is 2.37%, and the min value is 1.33%. Therefore, based on the iLeNet-5 NID system, its performance in DSR and FAR is greater than before, which also indicates that the improved intrusion detection system has better performance. To better validate the performance of the iLeNet-5 NID model, other comparative indicators were used in the study, and the UNSW-NB15 dataset was selected for testing. Other comparative indicators include central processing unit (CPU) utilization and memory usage, both of which are important indicators for measuring system performance. The testing environment and response time testing environment are consistent, and the testing frequency is also 5 times. The comparison of CPU utilization and memory utilization of the NID model before and after LeNet-5 improvement is shown in Table 4.

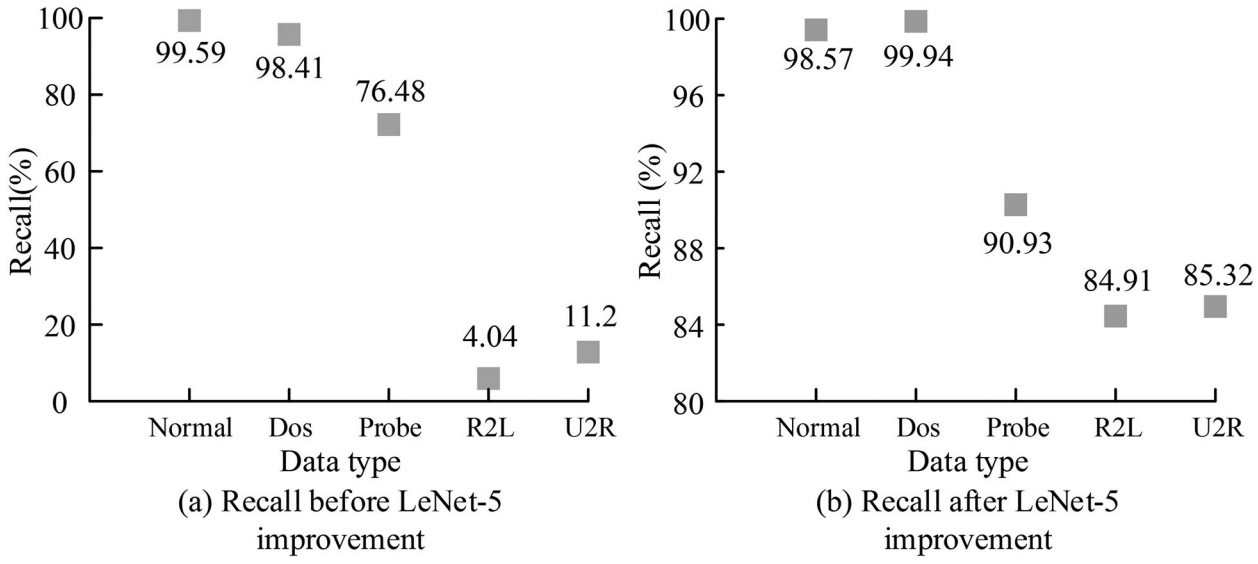


Figure 11. Comparison of recall rate of LeNet-5 NID model before and after improvement.

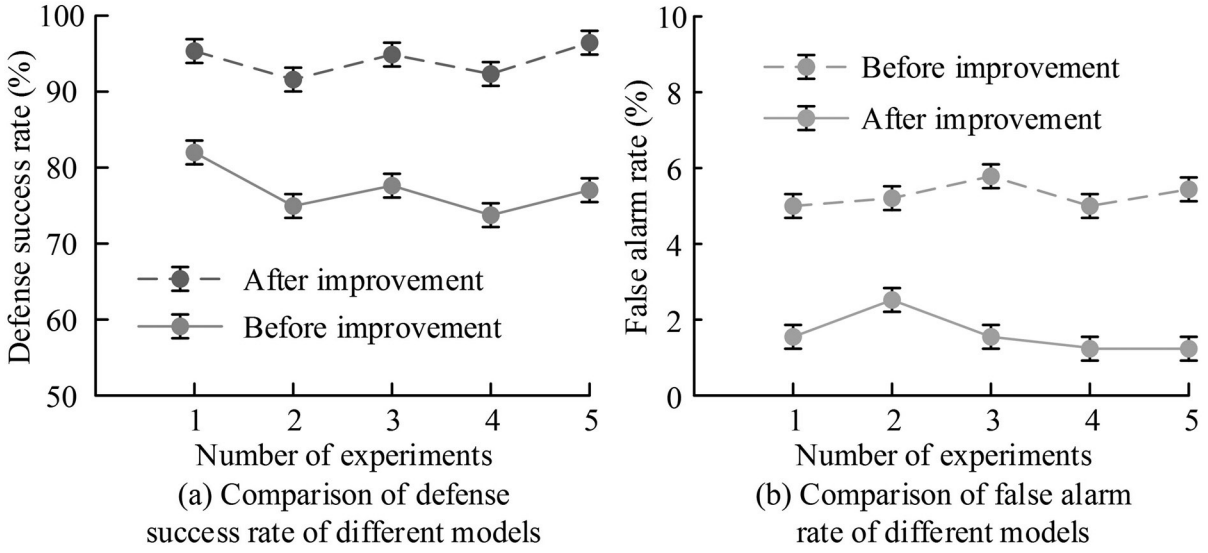


Figure 12. Comparison of DSR and FAR of NID model before and after improvement of LeNet-5.

Table 4. Comparison of CPU utilization and memory utilization of NID models before and after LeNet-5 improvement.

Model	CPU utilization					Memory usage rate				
	Number of experiments					Number of experiments				
	1	2	3	4	5	1	2	3	4	5
Before LeNet-5 improvement	17%	19%	23%	21%	18%	42%	48%	52%	56%	55%
Improved LeNet-5	10%	12%	11%	13%	15%	35%	37%	40%	36%	33%

From Table 4, the maximum CPU utilization of the intrusion detection system before LeNet-5 improvement is 23%, and the minimum is 17%. The maximum memory utilization is 56%, and the minimum is 42%. The maximum CPU utilization of the improved intrusion detection system in LeNet-5 is 15%, and the minimum is 10%. The maximum memory utilization is 40%, and the minimum is 33%. It can be inferred that the improved intrusion detection system performs better.

5. Conclusion

To optimize NID technology, this study innovatively designed a GA-INSA and an iLeNet-5 NID model. The latter one was regarded as an effective supplement to the GA-based improved NSA. The research results showed that when the data volume was 24000, the accuracy of the improved NSA was 97.3%, NSA was 90.1%, SVM algorithm was 94.1%, and GA-BP algorithm was 89.4%. When the data volume

was 24000, the FAR values of the four algorithms were 2.82%, 4.33%, 3.81%, and 3.02%, respectively. Based on the improved NSA, the maximum response time of the detection model was 60 ms and the minimum value was 45 ms. Therefore, the improved NSA algorithm performed better, and the GA-INSA intrusion detection model performed better. Regarding the accuracy of data types before and after LeNet-5 improvement, Normal was 73.50% and 96.22%, Dos was 99.81% and 99.86%, and U2R was 72.99% and 91.68%, respectively. The recall rates of data types before and after LeNet-5 improvement were 99.59% and 98.57% for Normal, 98.41% and 99.94% for Dos, and 11.2% and 85.32% for U2R, respectively. The maximum DSR and FAR values of the improved LeNet-5 system were 98.7% and 2.37%, respectively, and the minimum values were 91.2% and 1.33%, respectively. Therefore, the improved LeNet-5 performed better, and the performance based on the iLeNet-5 NID model was also more advantageous. However, this study also has certain shortcomings. In response to the improvement of the NSA, the study adopted the GA to generate detectors and optimized the non autogenous spatial distribution of detectors, which to some extent improved the detection efficiency of the NSA. However, there is room for improvement in the detection efficiency of NSA. Future research can optimize the NSA based on the findings to enhance its performance and detection efficiency. Furthermore, deploying artificial intelligence-based intrusion detection systems in the real world presents various challenges due to the ever-changing types of network attacks encountered. Currently, most network intrusion systems are built based on existing types of network attacks, and there is still room for improvement in identifying and detecting new types of network attacks.

Acknowledgement

2023 Henan Province Key R&D and Promotion Project (Soft Science): Research on fault recording data analysis of flexible HVDC transmission valve control based on deep learning (232400410357).

References

[1] X. Zhao, "A Network Security Algorithm Using SVC and Sliding Window", *Wireless Networks*, vol. 29, no. 1, pp. 345–351, 2022. http://dx.doi.org/10.1007/s11276-022-03064-z

[2] Y. Liu *et al.*, "Identifying Important Nodes Affecting Network Security in Complex Networks", *International Journal of Distributed Sensor Networks*, vol. 17, no. 2, pp. 1560–1571, 2021. http://dx.doi.org/10.1177/1550147721999285

[3] I. Hidayat *et al.*, "Machine Learning-Based Intrusion Detection System: An Experimental Comparison", *Journal of Computational and Cognitive Engineering*, vol. 2, no. 2, pp. 88–97, 2022. http://dx.doi.org/10.47852/bonviewJCCE2202270

[4] N. Tran *et al.*, "Effect of Class Imbalance on the Performance of Machine Learning-based Network Intrusion Detection", *International Journal of Performability Engineering*, vol. 17, no. 9, pp. 741–755, 2021. http://dx.doi.org/10.23940/ijpe.21.09.p1.741755

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

[6] X. Kan *et al.*, "A Novel IoT Network Intrusion Detection Approach Based on Adaptive Particle Swarm Optimization Convolutional Neural Network", *Information Sciences*, vol. 568, pp. 147–162, 2021. http://dx.doi.org/10.1016/J.INS.2021.03.060

[7] H. Qiu *et al.*, "Adversarial Attacks Against Network Intrusion Detection in IoT Systems", *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10327–10335, 2020. http://dx.doi.org/10.1109/JIOT.2020.3048038

[8] M. N. Injadat *et al.*, "Multi-stage Optimized Machine Learning Framework for Network Intrusion Detection", *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1803–1816, 2020. http://dx.doi.org/10.1109/TNSM.2020.3014929

[9] Y. He, "Identification and Processing of Network Abnormal Events Based on Network Intrusion Detection Algorithm", *International Journal of Network Security*, vol. 21, no. 1, pp. 153–158, 2019. http://dx.doi.org/10.6633/IJNS.201901 21(1).19

[10] R. H. Dong *et al.*, "An Intrusion Detection Model for Wireless Sensor Network Based on Information Gain Ratio and Bagging Algorithm", *International Journal of Network Security*, vol. 22, no. 2, pp. 218–230, 2020. http://dx.doi.org/10.6633/IJNS.202003 22(2).05

[11] M. Wei *et al.*, "An Intrusion Detection Mechanism for IPv6-based Wireless Sensor Networks", *International Journal of Distributed Sensor Networks*, vol. 18, no. 3, pp. 103053–33770, 2022. http://dx.doi.org/10.1177/15501329221077922

[12] X. Zhou *et al.*, "Hierarchical Adversarial Attacks Against Graph-neural-network-based IoT Network Intrusion Detection System", *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9310–9319, 2021. http://dx.doi.org/10.1109/JIOT.2021.3130434

[13] H. Jagruthi *et al.*, "Network Intrusion Detection Using Fusion Features and Convolutional Bidirectional Recurrent Neural Network", *International Journal of Computer Applications in Technology*, vol. 69, no. 1, pp. 93–100, 2022. http://dx.doi.org/10.1504/ijcat.2022.126095

[14] R. K. R. Rajesh Kanna *et al.*, "Improved Random Forest Algorithm for Cognitive Radio Networks' Distributed Channel and Resource Allocation Performance", *Journal of Logistics, Informatics and Service Science*, vol. 10, no. 3, pp. 98–106, 2023. http://dx.doi.org/10.33168/JLISS.2023.0308

[15] M. Liu *et al.*, "A Modified Real-value Negative Selection Detector-based Oversampling Approach for Multiclass Imbalance Problems", *Information Sciences*, vol. 556, pp. 160–176, 2021. http://dx.doi.org/10.1016/j.ins.2020.12.058

[16] E. Dandil, "C-NSA: A Hybrid Approach Based on Artificial Immune Algorithms for Anomaly Detection in Web Traffic", *IET Information Security*, vol. 14, no. 6, pp. 683–693, 2020. http://dx.doi.org/0.1049/iet-ifs.2019.0567

[17] K. D. Gupta and D. Dasgupta, "Negative Selection Algorithm Research and Applications in the Last Decade: A Review", *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 110–128, 2021. http://dx.doi.org/10.1109/TAI.2021.3114661

[18] S. Ji el al., "Parallel Sparse Filtering for Intelligent Fault Diagnosis Using Acoustic Signal Processing", *Neurocomputing*, vol. 462, pp. 466–477, 2021. http://dx.doi.org/10.1016/j.neucom.2021.08.049

[19] L. Zhu *et al.*, "Social Media Communication Network Analysis and Influence Propagation Model: A Case Study", *Journal of Logistics, Informatics and Service Science*, vol. 10, no. 3, pp. 264–279, 2023. http://dx.doi.org/10.33168/JLISS.2023.0320

[20] J. Li *et al.*, "FIMF Score-CAM: Fast Score-CAM Based on Local Multi-feature Integration for Visual Interpretation of CNNS", *IET Image Processing*, vol. 17, no. 3, pp. 761–772, 2022. http://dx.doi.org/10.1049/ipr2.12670

[21] S. Gurumurthy *et al.*, "Hybrid Pigeon Inspired Optimizer-gray Wolf Optimization for Network Intrusion Detection", *Journal of System and Management Sciences*, vol. 12, no. 4, pp. 383–397, 2022. http://dx.doi.org/10.33168/JSMS.2022.0423

[22] F. Guo *et al.*, "Automatic Rail Surface Defects Inspection Based on Mask R-CNN", *Transportation Research Record*, vol. 2675, no. 11, pp. 655–668, 2021. http://dx.doi.org/10.1177/03611981211019034

[23] Q. Zhang *et al.*, "The Detection of Hyperthyroidism by the Modified LeNet-5 Network", *Indian Journal of Pharmaceutical Sciences*, vol. 82, no. 5, pp. 108–114, 2020. http://dx.doi.org/10.36468/pharmaceutical-sciences.spl.108

[24] Z. Xu *et al.*, "An Oversampling Algorithm Combining SMOTE and k-means for Imbalanced Medical Data", *Information Sciences*, vol. 572, pp. 574–589, 2021. http://dx.doi.org/10.1016/j.ins.2021.02.056

Received: October 2023
Revised: December 2023
Accepted: December 2023

Contact address:
Long Li
Department of information engineering
Xuchang Electrical Vocational College
Xuchang
Henan
China
e-mail: longforpaper@163.com

Long Li is currently an associate professor in Xuchang Electrical Vocational College and also the deputy director of the Research and Foreign Affairs Center (department level). His research interests include computer intelligence information technology,computer network security protection and algorithm theory research.