Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9956371, 11 pages
https://doi.org/10.1155/2021/9956371

Research Article

Intrusion Detection for Network Based on Elite Clone Artificial Bee Colony and Back Propagation Neural Network

Guohong Qi, Jie Zhou, Wenxian Jia, Menghan Liu, Shengnan Zhang, and Mengying Xu

College of Information Science and Technology, Shihezi University, Shihezi, China

Correspondence should be addressed to Jie Zhou; jiezhou@shzu.edu.cn

Received 1 April 2021; Revised 1 July 2021; Accepted 11 August 2021; Published 13 September 2021

Academic Editor: Rubén González Crespo

With the rapid development of Internet technology, network attacks have become more frequent and complex, and intrusion detection has also played an increasingly important role in network security. Intrusion detection is real-time and proactive, and it is an indispensable technology under the diversified trend of network security issues. In terms of network security, neural networks have the characteristics of self-learning, self-adaptation, and parallel computing, which are very important in intrusion detection. This paper combines back propagation neural network (BPNN) and elite clone artificial bee colony (ECABC) to propose a new ECABC-BPNN, which updates and optimizes the settings of traditional BPNN weights and thresholds. Then, apply ECABC-BPNN to network intrusion detection. Use the attack data samples of KDD CUP 99 and water pipe for attack classification experiments using GA-BPNN, PSO-BPNN, and ECABC-BPNN. The results show that the ECABC-BPNN proposed in this paper has an accuracy rate of 98.08% on KDD 99 and 99.76% on water pipe data. ECABC-BPNN effectively improves the accuracy of network intrusion classification and reduces classification errors. In addition, the time complexity of using ECABC-BPNN to classify network attacks is relatively low. Therefore, ECABC-BPNN has superior performance in network intrusion detection and classification.

1. Introduction

The innovation of information technology has brought convenience to people, and the Internet is indispensable in all aspects of work, study, and life. With the rapid development of network resources and users, network intrusions have become more frequent and complex. Network security has become a topic of general concern among researchers.

Cyberattacks include a variety of methods, such as personal data theft and data pollution [1]. Firewalls and antivirus software are necessary components of security protection in various fields. However, the use of a firewall alone cannot meet the needs of ensuring network security. In March 2019, Norsk Hydro, one of the world's largest aluminum manufacturers, suffered a large-scale cyberattack on its operations in several factories in Europe and the United States, rendering its IT systems unusable. The company had to temporarily close the factory and change the operating mode to manual mode to continue the company's operations. In July, a ransomware virus attacked City Power Company in Johannesburg. The applications and databases were maliciously encrypted by hackers, which paralyzed its external services. On October 22, the cloud service provider Amazon's AWS DNS server was attacked by DDoS. The attacker blocked the system through junk network traffic and made the service inaccessible. Targeted cyberattacks continue to occur, posing a serious threat to the security of cyberspace.

Intrusion detection technology monitors network attacks through real-time detection and analysis of various behavioral data on the network [2–6]. It can detect different types of network intrusions, including illegal access, malicious attacks, stealing confidential information, stealing system accounts, and DoS [7–9]. To better identify cyberattacks,

network intrusion behaviors need to be classified through network intrusion detection. When detecting network attacks, it is important to extract the characteristics of network intrusion behavior. These characteristics can effectively provide information and describe attacks.

Due to the interference of various factors, network intrusion behavior is very complicated, and there are many characteristics of network intrusion. To effectively detect network intrusions, many scholars have invested in research. Common algorithms include Boyer-Moore string search, fast search string, and pattern matching [10–12]. Although these algorithms have their advantages, they are not novel enough.

To this end, this paper proposes a new type of neural network that combines elite cloned artificial bee colony (ECABC) and BP neural network (BPNN), called ECABC-BPNN. ECABC-BPNN can efficiently and quickly detect normal behaviors and attacks in network traffic in cyberspace. It can be used as an auxiliary to the network firewall to help predict the key protection direction of cyberspace in a certain period and ensure the security of the network operating environment. Simulation results show that the ECABC-BPNN proposed in this paper can effectively classify network attack data. Compared with PSO-BPNN and GA-BPNN, ECABC-BPNN has advantages in classification accuracy, precision, and recall.

The structure of this paper is as follows. Section 2 describes the research done to improve the efficiency of network intrusion detection and the use of artificial bee colony (ABC) in different fields. In Section 3, we propose an ABC optimized based on elite and clone operators to improve the weights and thresholds and introduced in detail the design and implementation of data network intrusion detection based on BPNN. Section 4 discusses the experimental results, and then, Section 5 concludes.

2. Related Work

Intrusion detection collects data from the network, analyzes the gathered information, and judges whether attacks have occurred in the network.

Work [13] designed an improved k-dependent Bayesian network structure model, which accurately described the dependence between system variables. In addition, the authors introduced a maximum posterior criterion and a small sample virtual expansion method to construct an improved intrusion detection classification model. The model was validated using the KDD CUP 99 data set, which showed that the model has high detection accuracy and stability. In work [14], researchers use a novel nonparametric Bayesian model to establish detection models for known and unknown intrusions. In the author's method, the activity mode of the infinitely bounded generalized Gaussian mixture model is understood through Bayesian Markov Chain Monte Carlo inference. To obtain better clustering performance, the researchers evaluated the method they developed using popular data sets, and the results obtained proved the effectiveness of the method in detecting various attacks. To improve the performance of industrial network intrusion detection, work [15] proposed an industrial network intrusion detection algorithm optimized by multifeature data clustering. The algorithm classifies the weighted distance and safety factor of the data by the priority of the data feature. Researchers show that the algorithm effectively improves the detection rate and real-time performance of abnormal behavior of multifeature data in an industrial network. Compared with other algorithms, this algorithm has better superiority in detection rate and time. In [16], a K-means/K-modes architecture based on reconfigurable FPGA was proposed to accelerate data clustering in network intrusion detection. The author evaluated the proposed method on the NSL-KDD data set and showed that K-means and K-mode can achieve 15 times and 994 times more operation than the parallel software version. However, the traditional classifiers in [13–16] have their shortcomings. For example, K-means has strict requirements on the integrity of the training data and the rationality of the scaling factor. When the massive training data set is semistructured, it is difficult for K-means to achieve efficient classification.

In contrast, the neural network has the advantages of large-scale parallelism, distributed processing, and self-learning. These advantages enable the neural network to extract the deep attributes of the data set and solve the problem of low training efficiency of ordinary intrusion detection classifiers and the need for accurate label data [17, 18].

In [19], a support vector machine optimization based on GA is proposed. The authors use a feature selection method based on genetic algorithms to reduce the error rate of support vector machines and increase the true positive rate. The authors prove through experiments that the proposed algorithm improves the classification accuracy and shortens the classification time. Work [20] proposed an intrusion detection classifier based on the epigenetic algorithm to classify network attack types. The authors prove through experimental comparison that the performance of the proposed new algorithm has a higher detection rate than the GA classifier, and the processing time is faster than GA and other classification algorithms. An intrusion detection model based on an improved genetic algorithm and deep belief network is proposed in work [21]. Researchers use the NSL-KDD data set to evaluate their algorithms. The results show that the improved intrusion detection model improves the recognition rate of intrusion attacks and reduces the complexity of network attacks. Although GA combined with learning tools such as support vector machines can classify intrusion detection more effectively. However, support vector machines also have problems such as difficulty in determining appropriate function parameters. Moreover, the GA space exploration ability combined with it is limited, and it is easy to fall into a state of evolutionary stagnation. Therefore, the neural network optimized with GA still has the possibility of improvement.

Work [22] proposed a back propagation neural network combined with particle swarm optimization (PSO) based on an entropy model. The neural network introduces the entropy model to obtain the search characteristics of PSO, and PSO based on the entropy model algorithm absorbs the weights and thresholds. Therefore, the algorithm can

obtain higher solution accuracy and intrusion detection rate. Work [23] takes the wavelet neural network as the object and uses the quantum PSO for training. The results show that the neural network performance trained by the improved quantum PSO is better than other intelligent algorithms such as traditional PSO and GA. Although PSO has the characteristics of continuity, it is more suitable for neural network training than GA. Due to its single-item information sharing feature, although PSO speeds up its convergence, it is more likely to fall into a local optimum. Therefore, using PSO to optimize neural networks is not the best choice.

Work [24] proposed a method through using ABC to optimize BPNN to find the best weights and thresholds. This method utilizes bees to effectively find high-quality food. The optimized neural network has the generalized mapping ability of the BPNN and has the global iteration and local search capabilities of the ABC. Through comparative experiments, it proved that the prediction model of a neural network optimized based on ABC has effectively improved the accuracy of PM2.5. In [25], the authors proposed a short-term wind speed prediction method based on improved ABC-based BPNN. The weights and thresholds of the BPNN are optimized by the improved ABC and proved that the algorithm has the characteristics of high accuracy. In addition, based on the characteristics of ABC and the superior performance of the neural network in classification [26], it is combined with improved ECABC and BPNN to classify network intrusion detection. Compared with PSO and GA, the proposed ECABC has better global search ability, thus speeding up the training speed of BPNN and improving the classification accuracy of network intrusion.

3. Network Attack Detection Based on ECABC-BPNN

For BPNN, the learning result needs to converge to the global minimum of mean square error. ECABC has a strong convergence performance in the overall situation. It uses the greedy selection of nonemployed bees and the selection of following bees to gradually improve the iterative efficiency. Meanwhile, to avoid falling into the local optimum, explored bees are added to supply and increase the diversity of solutions. The most prominent advantage of ECABC is that it performs global and local searches together during each iteration to improve the possibility of finding the best solution.

This section proposes an ECABC-BPNN and applies it to network intrusion detection. Using ECABC to optimize BPNN has the following advantages:

(1) We propose a new ECABC-BPNN weight and threshold selection method. In ECABC-BPNN, the initial weights and thresholds are obtained according to the algorithm. There are many uncertainties in BPNN. If weights and thresholds are not appropriate, it will affect the training of the network and convergence speed and cause the train to fall into local optimal. On the premise of ensuring the colony number, the initial bees' population is evenly distributed in the problem connection weight space. ECABC can effectively avoid falling into a local minimum due to the singular initial value during the BPNN learning and training process. ECABC-BPNN can set appropriate weights and thresholds through powerful global search capabilities to make the network converge relatively quickly

(2) The overall concept of ECABC changes the status quo that BPNN only generates a set of values during initialization and can only adjust the error in one dimension. In ECABC, each initial individual of the population corresponds to a network, and the parameters of the network can be adjusted in multiple dimensions at the same time, thereby speeding up the convergence

(3) In the optimization process of BPNN parameters, ECABC can control the update of each value set in the population by comparing the size of the fitness value in each iteration by using the elite and clone operators and always record the position with better fitness value. By comparing the optimization results, the algorithm can find the global minimum set in each iteration to avoid the optimization process from falling into the local optimal state

(4) One aspect of network structure design needs attention, namely, the number of hidden layers in the network structure. Therefore, we design a novel ECABC-BPNN structure. To improve the operating efficiency of the neural network without affecting the performance, we studied the effect of the number of hidden layers on the performance of the neural network and selected appropriate parameters through experimental comparison when designing ECABC-BPNN parameters

ECABC-BPNN improves the defects and shortcomings of BPNN. It takes the weights and thresholds of BPNN as the target output value of the ECABC. Moreover, ECABC improves the global search capability of BPNN and reduces the probability of falling into local optimal, making the convergence speed faster.

The pseudocode of ECABC-BPNN is shown in Pseudocode 1.

The following is a detailed introduction to the steps of ECABC-BPNN.

Step 1. Initialize the parameters of BPNN. Use the result as the initial artificial bee population of ECABC, which is also the initial feasible solution.

Step 2. Create the initial population in ECABC according to the initial parameters.

Step 3. According to the decoding strategy, decode the discrete population to obtain the corresponding decimal value, which means that the relevant data of the weights and

```
ECABC-BPNN () {
    parameter = BPNN initialize();
    ECABC (parameter);
    BPNN classify();
    BPNN train(best_value);
}
ECABC (parameter) {
    ECABC initialize();
    for (generation=0; generation<Max_generation; generation++) {
        ECABC search for the best fitness value;
    }
    return (best_value);
```

Pseudocode 1: Pseudocode of ECABC-BPNN.

thresholds required by the BPNN in the process of learning iteration are obtained.

Step 4. Pass the decoded data obtained after ECABC optimization to BPNN to get the fitness value. Consider the misclassification result of the network attack test data as the adaptability of a group in ECABC at this time.

Step 5. Use the training sample to optimize the BPNN parameters according to the ECABC rules.

Step 6. Determine whether the ECABC iteration has reached the end condition of the algorithm. If it reaches the end condition, terminate the optimization process and output the learning result to BPNN.

Step 7. Input test samples to BPNN to test current learning results after optimization.

Step 8. Calculate the test error of learning results from ECABC optimization on the test data.

Step 9. The weights and thresholds optimized by ECABC are input into BPNN as initial parameters. Train and classify the neural network, and compare the differences.

Figure 1 shows the ECABC-BPNN execution process proposed in this paper.

The following is the basis and improvement principle of ECABC.

3.1. Initialize the Bee Population. When using ECABC to optimize BPNN for network attack data classification, it is very important to set appropriate parameters. ECABC first initializes the basic parameters, including the total number of bees, the number of employed bees, the number of onlooker bees, the number of explored bees, and the number of nectar sources. Among them, the number of employed bees and follow bees is equal, and they each account for half of the total. The number of nectar sources equals the number of hired bees. Besides, it is necessary to initialize the number of population iterations, the number of restrictions required when discarding nectar sources, etc. The nectar source is composed of the initial parameters generated when constructing BPNN.

3.2. Employed Bees Perform Searching Tasks. In the employed bees, to improve the correct rate of attack detection classification, use equation (1) to find a new candidate solution.

$$V_i = B_{id} + r(B_{id} - B_{kd}), \quad k \in \{1, 2, \cdots, N, k \neq i\}, d \in \{1, 2, \cdots, D\}, \tag{1}$$

where k represent the employed bees and d represent the dimensions of the employed bees searching for nectar source B. r is a disturbance item randomly distributed between [-1,1], which determines the magnitude of the disturbance. V_i means a new nectar source which corresponds to the employed bees.

Equation (1) shows that as the iteration progresses, the values of B_{id} and B_{kd} are getting closer and closer, so the disturbance amplitude of the position update is gradually reduced. Meanwhile, when the fitness value of the new nectar source selected by the employed bee is better than the old nectar source, the nectar source has a certain probability of cloning. In this way, when the algorithm is close to the optimal solution, the artificial bee can dynamically adjust the step length when searching for nectar sources. The employed bees choose between the original B_i and the newly generated V_i according to the greedy rule and compare the fitness value between the original B_i and the new V_i. The number of misclassifications of the optimized parameters in the BPNN classification test is used as the fitness. If fit(V_i) < fit(B_i, replace B_i with V_i, otherwise the original B_i is still retained.

3.3. Onlooker Bees Perform Following Tasks. Onlooker bees can increase the weight of better weights and thresholds in BPNN, make the optimization more effective, and improve the BPNN attack classification performance. At this stage, the onlooker bees make judgments and choices by themselves. In the beginning, the employed bees first share the nectar source information they hold. Then, the onlooker bees judge whether the message is valid according to the observation and choose whether to follow the nectar according to the calculation.

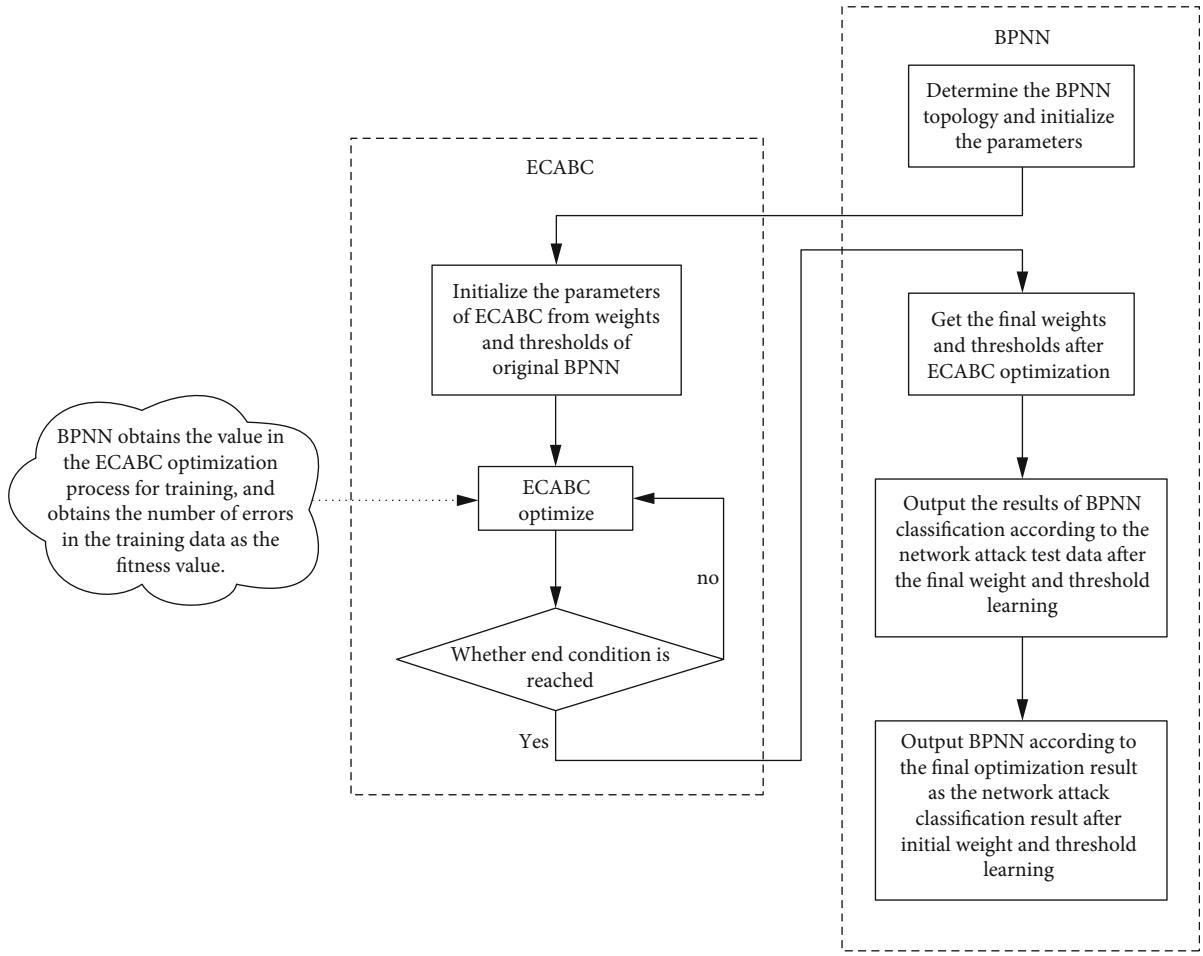

Figure 1: ECABC-BPNN optimization and network intrusion data classification process.

To find a better nectar source solution and obtain a better return, ECABC uses roulette to determine which nectar source to choose. The roulette equation is

$$P_b = \frac{1 - F(\delta_b)}{\sum_{d=1}^{B} F(\delta_d)}. \tag{2}$$

where P_b represent the probability of selecting the b corresponding nectar sources. B represents the total number of nectar sources. δ means the position of the b employed bee near the nectar source. F represents the fitness function of the nectar source. The smaller the nectar source's fitness value, the higher the probability that it will be selected by the roulette.

Onlooker bees search for an area corresponding to the message provided by the employed bees. The collection process of the onlooker bees is the same as that of the employed bees. Use equation (1) to find a new nectar source result, and compare and obtain a better solution. Make an elite selection of new nectar sources. If the fitness value of the new nectar source is higher than that of the old nectar source, it is judged that the selection is invalid. Besides, there is a parameter for the nectar to indicate the number of times the nectar is not updated. When the nectar source is updated, the parameter is recorded as 0 and will not change. Conversely, if the nectar source information has not been updated, the parameter value is increased by 1.

3.4. Explored Bee's Judgement. The explored bees are the key for ECABC to jump out of the local optimum when optimizing BPNN. Setting appropriate exploration bee parameters can improve the efficiency of BPNN intrusion classification. They can determine whether a nectar source needs to be replaced. According to the parameters provided by the bees, when the parameter value reaches the set limit range, a new nectar source will be set. At this time, the originally employed bees abandon the old nectar source and look for new sources around them. Its role is changed to explored bees, showing the negative feedback and bumpy properties of self-organization in ECABC. In this stage, the explored bees will use equation (3) to randomly find a new source of nectar to replace the old.

$$V_j = \text{rand}\,(B_d), \quad d \in \{1, 2, \cdots, D\}, \tag{3}$$

where V_j is the new nectar source chosen by the explored bees and d represent the dimensions of the bee searching for nectar source B. Elite and clone operators play a role in preventing the optimization process from regressing at this moment.

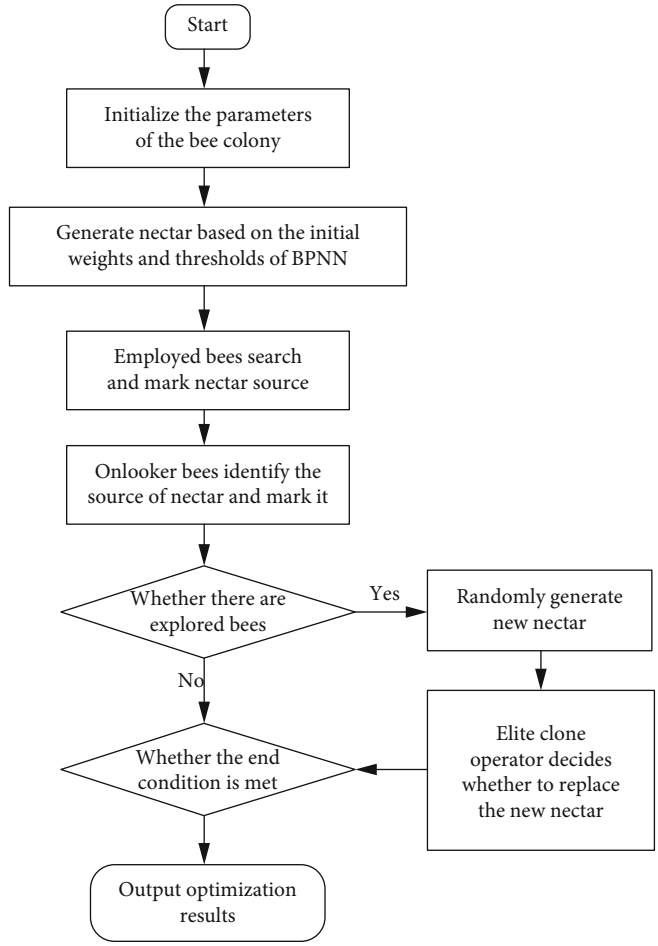

Figure 2: The process of ECABC optimization.

3.5. Elite Operator. The elite operator can promote global optimization performance in the ECABC-BPNN intrusion detection data classification training process. Its main idea is to save part of the high-quality solutions in the group, provide guidance for all groups, accelerate the convergence speed of ECABC, and improve the search efficiency. In each iteration, ECABC uses the elite operator to retain the elite nectar sources and accelerates the convergence of the algorithm through the clone of elite sources.

3.6. Clone Operator. The core idea of the cloning operator is to only increase individuals with high performance in cloning and reproduction, and other individuals with poor performance will not be cloned. The performance of the clone depends on the mechanism of individual proliferation. If the proliferation range is too small, it will affect the diversity of the population species. In the iterative optimization and solution of the colony, ECABC clones and mutates towards the optimal population according to the distribution of the current solution set, thus reflecting the advantages of ECABC and ensuring the effectiveness after cloning and mutation. Figure 2 illustrates the process of optimizing ECABC network intrusion parameters.

4. Experimental Simulation and Discussion

The main parameters of the ECABC are the size of the bee colony, the limit value, the number of employed bees, the number of following bees, the number of solutions, and the maximum number of iterations. The number of employed bees is equal to the number of the following bees, and both are equal to the solutions. There are big differences in the parameter values of the net attack data in ECABC-BPNN. Obviously, the larger the bee colony, the greater the possibility of obtaining the optimal solution. However, increasing the number of bees will make the calculation more complicated and consume lots of network resources. Therefore, the parameters in the ECABC need to set through simulation tests.

To prevent the deviation of the simulation experiment results, this paper performs the data preprocessing before the neural network training, and the neural network input value is normalized. This simulation uses the maximum-minimum difference method as the normalization processing method, as shown in the following equation:

$$m_j = \frac{m_j - m_{\min}}{m_{\max} - m_{\min}}, \tag{4}$$

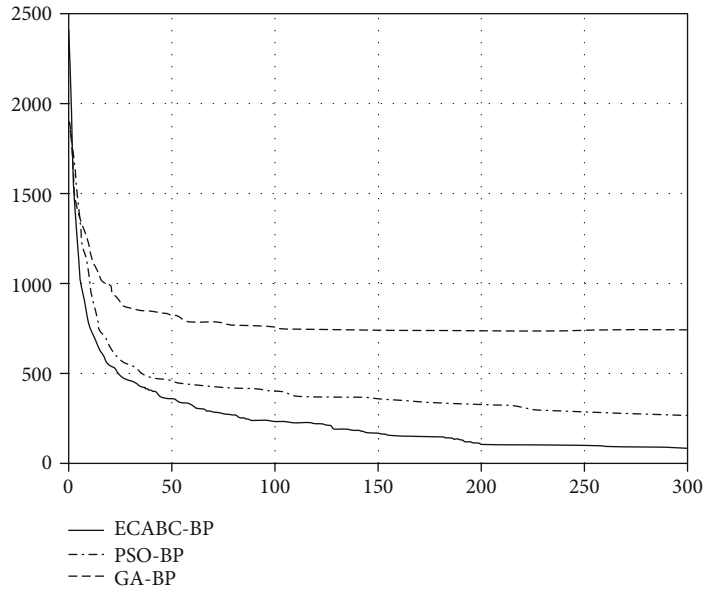

Figure 3: The number of misclassifications of the KDD 99 training data optimized by ECABC varies with the number of iterations.

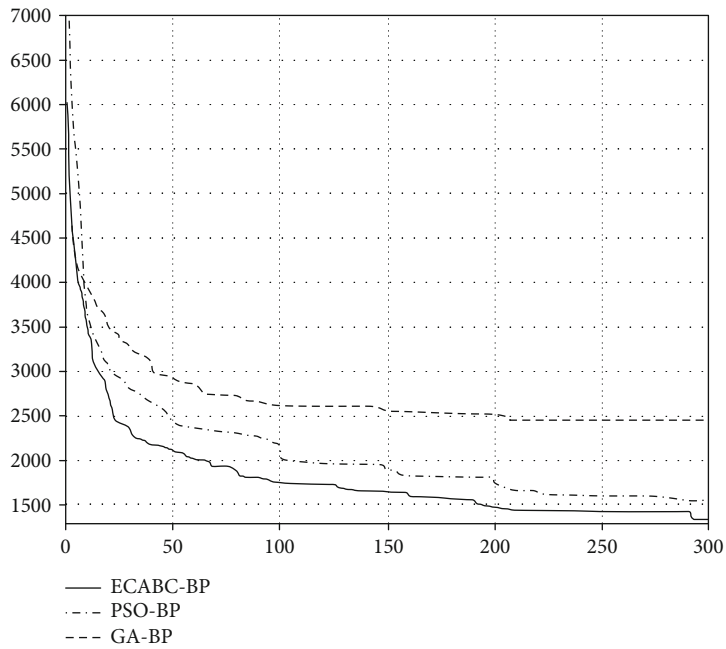

Figure 4: The number of misclassifications of the industrial water pipe training data optimized by ECABC varies with the number of iterations.

where m_j is the j input parameter value in ECABC-BPNN, $m_{\max}$ is the maximum value in the input value, and $m_{\min}$ is the minimum value in the input value.

In selecting data sets for testing and improving algorithm performance, the sample data used in this paper are the KDD CUP 99 data set and the industrial water pipe data set. The KDD CUP 99 data set is a data set used for network attack detection, which is very typical and highly recognized [27]. It is a de facto benchmark in the field of network attack detection and prepares for the exploration of network attack detection based on intelligent computing. Intrusion types include DOS, probing, R2L, and U2R. The industrial water pipe data set is collected from the factory, and the attack types include NMRI, CMRI, MSCI, MPCI, MFCI, DoS, and reconnaissance [28].

Table 1: The classification accuracy of the three BPNN on the network attack test data.

Accuracy (%)	ECABC-BPNN	PSO-BPNN	GA-BPNN
KDD 99	98.08	94.18	83.16
Water pipe	99.76	99.53	99.28

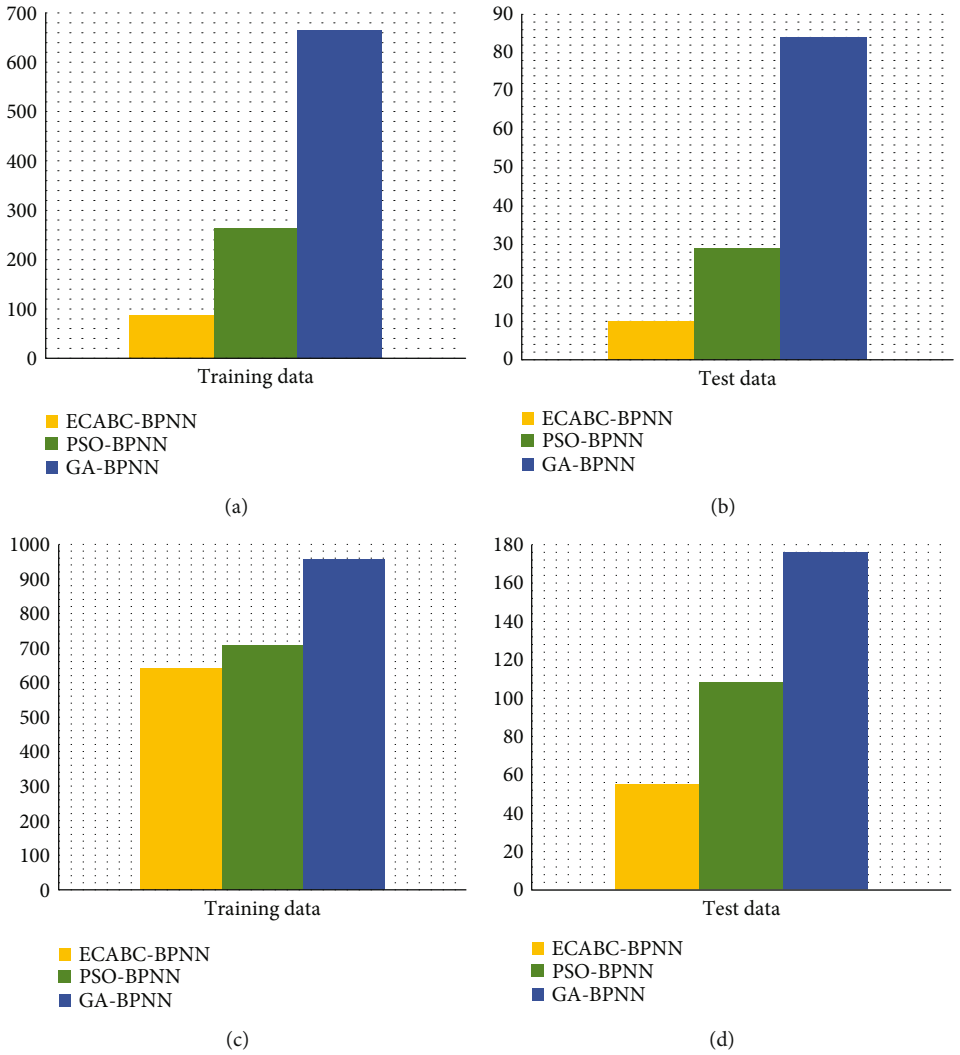

Figure 5: Comparison of errors between training data and test data optimized by ECABC-BPNN, PSO-BPNN, and GA-BPNN: (a) KDD 99 errors between training data optimized by three algorithms; (b) KDD 99 errors between test data optimized by three algorithms; (c) water pipe errors between training data optimized by three algorithms; (d) water pipe errors between test data optimized by three algorithms.

Figure 3 indicates how the optimization results of ECABC, PSO, and GA for BPNN weights and thresholds vary with the number of iterations when there are 4000 training data in KDD CUP 99. The optimization result is the average of 20 optimizations. The simulation results show that GA has an advantage at the beginning of the algorithm iteration. However, it quickly falls into evolutionary stagnation during the iteration process, and the final optimization gets the most training error data. The performance of PSO is relatively stable, and the final error data obtained is much less than that of GA. But PSO cannot dynamically adjust parameters during the evolution process, which results in slower algorithm evolution. ECABC can dynamically adjust the direction of optimization during the optimization process, which enhances the global search capability and prevents the algorithm from falling into the local optimum. Besides, elite operator and clone operator ensure the forward evolution of the optimization process, making ECABC avoid evolutionary stagnation and premature convergence. Under the same conditions, the classification error rate of ECABC for intrusion detection data is reduced by 4.534% compared with PSO and by 16.300% compared with GA. That is, ECABC has a lower error rate in the classification of training data than PSO and GA.

Figure 4 shows how the optimization results of ECABC, PSO, and GA on BPNN weights and thresholds vary with the number of iterations when the water pipe has 10,000 training data. The simulation results show that ECABC has always had an advantage in algorithm iteration. GA quickly falls into a local optimum in the iterative process, and the final optimization result is the worst. The performance of PSO is relatively stable, and the final error data obtained is much smaller than GA but still worse than ECABC. Under the same conditions, the error rate of ECABC in training data classification is lower than that of PSO and GA. The error rate of ECABC's detection and classification of water pipe data is reduced by 2.197% compared with PSO and 11.266% compared with GA.

After ECABC, PSO, and GA optimize the initial weights and thresholds of BPNN, the optimized BPNN performs

simulation experiments on the performance of KDD 99 and water pipe test data, respectively. Table 1 shows the accuracy of three types of BPNN to classify network attack test data.

As can be seen from Table 1, in general, ECABC-BPNN has higher network attack classification accuracy than PSO-BPNN and GA-BPNN by 3.90% and 14.92%, respectively. The rate of ECABC-BPNN classification on water pipe is also the highest. To clearly distinguish the effects of ECABC, PSO, and GA optimization, we use Figure 5 to show the proportion of misclassifications when BPNN uses the optimization results of the three algorithms as weights and thresholds to classify network attacks.

Figure 5 shows that when using GA-BPNN to classify network attacks in training data and test data, the number of classification errors is the largest. Compared with GA-BPNN, PSO-BPNN has reduced the number of misclassifications but still has higher errors than ECABC-BPNN. Using ECABC-BPNN to classify training and test data has the lowest misclassification number in KDD 99 and water pipe.

Table 2 shows the precision rate of BPNN using ECABC, PSO, and GA optimized parameters as weights and thresholds. It can be seen from Table 2 that ECABC-BPNN has the highest classification accuracy for the four detection categories of normal, DoS, probing, and U2R compared to PSO-BPNN and GA-BPNN. In normal, the classification accuracy of ECABC-BPNN is 8.453% higher than PSO-BPNN and 10.483% higher than GA-BPNN. In DoS, the accuracy is 0.316% better than that of PSO-BPNN and 8.500% better than that of GA-BPNN. In probing, the classification accuracy of ECABC-BPNN is 15.967% and 62.796% well than that of PSO-BPNN and GA-BPNN, respectively. In U2R, ECABC-BPNN is 50.340% outstanding than PSO-BPNN and 75.279% outstanding than GA-BPNN. For R2L, the classification accuracy of ECABC-BPNN is only 0.180% lower than that of PSO-BPNN, but it is still 15.243% higher than that of GA-BPNN.

Table 3 shows the precision rate of BPNN on water pipes using ECABC, PSO, and GA optimized parameters. From the table, it can be watched out that the classification accuracy of ECABC-BPNN in the eight detection categories is basically higher than or equal to that of PSO-BPNN and GA-BPNN. ABC has better performance than PSO and GA in the detection and classification of water pipes.

The comparison of KDD 99 recall rates in Table 4 shows that in the four categories of normal, probing, R2L, and U2R, the recall rates of ECABC-BPNN are 1.418%, 26.097%, 20.841%, and 10.834% better than PSO-BPNN, respectively. And in DoS, the recall rate of both reached 100%. Compared with GA-BPNN, ECABC-BPNN has better recall rates in normal, DoS, probing, and R2L by 10.957%, 1.589%, 54.339%, and 58.186%. Only in the recall rate of U2R ECABC-BPNN is 4.166% lower than GA-BPNN.

The comparison of the recall rate of water pipes in Table 5 shows that in the seven categories of normal, NMRI, CMRI, MSCI, MFCI, DoS, and reconnaissance, the recall rate of ECABC-BPNN is better than that of PSO-BPNN and GA-BPNN. Only in MPCI the recall rate of ECABC-BPNN is lower than that of PSO-BPNN.

Table 2: The classification precision rate of the three BPNN on the KDD 99 test data.

Precision (%)	ECABC-BPNN	PSO-BPNN	GA-BPNN
Normal	98.969	90.516	88.486
DoS	98.562	98.246	90.062
Probing	86.709	70.742	23.913
R2L	97.196	97.376	81.953
U2R	83.673	33.333	8.394

Table 3: The classification precision rate of the three BPNN on the water pipe test data.

Precision (%)	ECABC-BPNN	PSO-BPNN	GA-BPNN
Normal	99.999	99.999	99.999
NMRI	99.999	99.853	99.663
CMRI	99.071	98.670	98.311
MSCI	97.035	96.774	96.941
MPCI	99.999	99.999	99.353
MFCI	97.983	98.039	83.951
DoS	96.552	95.758	91.848
Reconnaissance	99.999	99.929	99.964

Table 4: The classification recall rate of the three BPNN on the KDD 99 test data.

Recall (%)	ECABC-BPNN	PSO-BPNN	GA-BPNN
Normal	99.310	97.892	88.353
DoS	100.000	100.000	98.411
Probing	88.387	62.308	34.048
R2L	96.131	75.290	37.945
U2R	34.167	23.333	38.333

Table 5: The classification recall rate of the three BPNN on the water pipe test data.

Recall (%)	ECABC-BPNN	PSO-BPNN	GA-BPNN
Normal	99.621	99.448	99.462
NMRI	99.999	99.999	99.999
CMRI	99.688	99.379	98.980
MSCI	99.999	99.999	95.814
MPCI	98.625	99.059	98.462
MFCI	99.999	99.999	99.999
DoS	98.000	90.286	73.478
Reconnaissance	99.999	99.999	99.999

5. Conclusion

In this paper, to better classify network attacks, we propose the ECABC-BPNN method. It selects the best threshold and weight by combining ECABC and BPNN to improve

the performance of the neural network and the quality of network attack detection. At the same time, the method of this paper has good antinoise performance and user effect. In KDD 99, compared with PSO-BPNN and GA-BPNN, ECABC-BPNN has increased the detection accuracy of network attack test data by 3.90% and 14.92%, respectively. This method also improves the classification speed. In water pipe data, ECABC-BPNN also has a better performance. In short, ECABC-BPNN has a good effect on improving the accuracy of network intrusion classification.

Data Availability

The experiment data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was funded by the Corps innovative talents plan, grant number 2020CB001; the project of Youth and Middle-aged Scientific and Technological Innovation Leading Talents Program of the Corps, grant number 2018CB006; the China Postdoctoral Science Foundation, grant number 220531; Funding Project for High Level Talents Research in Shihezi University, grant number RCZK2018C38; Project of Shihezi University, grant number ZZZC201915B; and Postgraduate Education Innovation Program of the Autonomous Region.

References

[1] G. Abhilash and G. Divyansh, "Intrusion detection and prevention in software defined networking," in *2018 IEEE international conference on advanced networks and telecommunications systems (ANTS)*, pp. 1–4, Indore, India, 2018.

[2] Z. Zhang, H. Zhu, S. Luo, Y. Xin, and X. Liu, "Intrusion detection based on state context and hierarchical trust in wireless sensor networks," *IEEE Access*, vol. 5, pp. 12088–12102, 2017.

[3] J. Camacho, G. Maciá-Fernández, N. M. Fuentes-García, and E. Saccenti, "Semi-supervised multivariate statistical network monitoring for learning security threats," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2179–2189, 2019.

[4] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: a comparison study," *IEEE Access*, vol. 8, pp. 106576–106584, 2020.

[5] M. Zhou, Y. Li, L. Xie, and W. Nie, "Maximum mean discrepancy minimization based transfer learning for indoor WLAN personnel intrusion detection," *IEEE Sensors Letters*, vol. 3, no. 8, pp. 1–4, 2019.

[6] G. Nguyen, S. Dlugolinsky, V. Tran, and Á. L. García, "Deep learning for proactive network monitoring and security protection," *IEEE Access*, vol. 8, pp. 19696–19716, 2020.

[7] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[8] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[9] A. A. Cárdenas, P. K. Manadhata, and S. P. Rajan, "Big data analytics for security," *IEEE Security & Privacy*, vol. 11, no. 6, pp. 74–76, 2013.

[10] R. M. Yousufi, P. Lalwani, and M. B. Potdar, "A network-based intrusion detection and prevention system with multi-mode counteractions," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1–6, Coimbatore, 2017.

[11] E. Papadogiannaki, D. Deyannis, and S. Ioannidis, "Head(er) Hunter: fast intrusion detection using packet metadata signatures," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–6, Pisa, Italy, 2020.

[12] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: deep learning methods on network intrusion detection using NSL-KDD dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020.

[13] H. Yin, M. Xue, Y. Xiao, K. Xia, and G. Yu, "Intrusion detection classification model on an improved k-dependence Bayesian network," *IEEE Access*, vol. 7, pp. 157555–157563, 2019.

[14] W. Alhakami, A. Alharbi, S. Bourouis, R. Alroobaea, and N. Bouguila, "Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection," *IEEE Access*, vol. 7, pp. 52181–52190, 2019.

[15] W. Liang, K. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2020.

[16] L. A. Maciel, M. A. Souza, and H. C. de Freitas, "Reconfigurable FPGA-based K-means/K-modes architecture for network intrusion detection," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 8, pp. 1459–1463, 2020.

[17] M. Á. López, J. M. Lombardo, M. López et al., "Intelligent detection and recovery from cyberattacks for small and medium-sized enterprises," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 3, pp. 55–62, 2020.

[18] N. Manju, B. S. Harish, and N. Nagadarshan, "Multilayer feedforward neural network for internet traffic classification," *International Journal of Interactive Multimedia and Artificial I ntelligence*, vol. 6, no. 1, pp. 117–122, 2020.

[19] P. Tao, Z. Sun, and Z. Sun, "An improved intrusion detection algorithm based on GA and SVM," *IEEE Access*, vol. 6, pp. 13624–13631, 2018.

[20] M. Ezzarii, H. E. Ghazi, H. E. Ghazi, and F. E. Bouanani, "Epigenetic algorithm-based detection technique for network attacks," *IEEE Access*, vol. 8, pp. 199482–199491, 2020.

[21] Y. Zhang, P. Li, and X. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, pp. 31711–31722, 2019.

[22] T. Liu, J. Yao, and Q. Sun, "Intrusion detection algorithm of EPSO combined with BP neural network," in *2020 International conference on intelligent transportation, big Data & Smart City (ICITBS)*, pp. 893–896, Vientiane, Laos, 2020.

[23] L. Guo, "Research on anomaly detection in massive multimedia data transmission network based on improved PSO algorithm," *IEEE Access*, vol. 8, pp. 95368–95377, 2020.

[24] Q. Wang, Z. Xu, Z. Xu, Y. Shi, H. Wu, and G. Shen, "PM2.5 prediction model based on ABC-BP," in *2019 International conference on communications, information system and computer engineering (CISCE)*, pp. 140–143, Haikou, China, 2019.

[25] G. Jia, D. Li, L. Yao, and P. Zhao, "An improved artificial bee colony-BP neural network algorithm in the short-term wind speed prediction," in *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, pp. 2252–2255, Guilin, China, 2016.

[26] M. S. B. M. Kasihmuddin, M. A. B. Mansor, S. A. Alzaeemi, and S. Sathasivam, "Satisfiability logic analysis via radial basis function neural network with artificial bee colony algorithm," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp. 164–173, 2021.

[27] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, Ottawa, ON, Canada, 2009.

[28] T. Morris and W. Gao, "Industrial control system traffic data sets for intrusion detection research," in *International conference on critical infrastructure protection*, pp. 65–78, Berlin, Heidelberg, 2014.