Intrusion Detection based on Neural Networks and Artificial Bee Colony Algorithm

Quan Qian, Jing Cai, Rui Zhang
School of Computer Engineering & Science, Shanghai University, Shanghai, China
Email: qqian@shu.edu.cn

Abstract—**Intrusion detection, as a dynamic security protection technology, is able to defense the internal and external network attacks. Using Artificial Bee Colony algorithm to optimize the parameters of neural network is to avoid the neural network falling into a local optimum, can solve the problem of slow convergence speed of the neural network algorithm. Also Artificial Bee Colony algorithm can deal with the problem of finding the optimal solutions in a very short period of time. In this paper, An Artificial Bee Colony optimized neural network algorithm is applied to intrusion detection. And the experimental results shows that the optimized method has better detection accuracy and efficiency than the single BP neural network.**

Keywords—Intrusion detection; Neural network; Artificial bee colony algorithm

I. INTRODUCTION

With the rapid development of network technology, especially the widely use of Internet, computers suffer more and more threats from viruses, trojan horses and other malware attacks. According to the network security threat report from Symantec Company, it shows that they blocked a total over 247,350 web malware attacks per day in 2012, 20.7% increase over 2011, Web-based attacks increased by 30% and the number of phishing sites(spoofing social networking sites) increased 125% over 2011[1].

Intrusion detection has been viewed as a very effective method in computer security, mainly by collecting and analyzing some information points such as network behavior, security logs and application audit data, to detect whether the network is under attack. According to detection mechanisms, Intrusion detection can be divided into two main categories: misuse detection and anomaly detection [2].

Misuse detection that is analyzing the matching degree with those unacceptable known behaviors. In misuse detection, we define abnormal system behavior at first, and then decide any other behavior, as normal behavior. It needs to collect the behavioral characteristics of abnormal behavior, and if the detected user's or system behavior matched the records in the misuse library, the behavior is considered an abnormal one. This detection approach has low positive false rate but relatively high false negative rate. For known attacks, this method can detect the type of attack detailed and accurately, but it cannot be accurate to detect those unknown attacks. Moreover, with the increasing of attack variants, the misuse signature library must also be updated constantly.

Anomaly detection is to recognize the difference among acceptable behaviors. First, we should collect the behavioral characteristics of the normal operations, and then establishes the behavioral model by analyzing the normal behaviors to find out those anomaly ones. This detection model has low false negative rate but high positive rate. As it does not need to predefine the behavior signature for each attack, it shows a certain degree of recognition ability for unknown attacks.

Concerning the intrusion detection process, it mainly includes three phases: information collection, analysis and results processing. At the phase of information collection, the collected information mainly includes network traffic, host audit data and user's operating logs. During the information analysis stage, it mainly analyses the collected data. Generally there are three techniques for information analysis: pattern matching, statistical analysis and integrity analysis. During processing the results, if abnormal behavior was found through analyzing, the detection system will give alarm or implement some other pre-defined anti-measures, for instance, shutdown the network, lock the host, etc.[2,3].

Intrusion detection model first proposed by Dorothy Denning[2]. Most current detection systems are based on the basic model. Some popular products are SRI IDES, IBM's IERS, ISS's RealSecure and etc. Although there are so many intrusion detection systems, they are still not mature enough and the speed and accuracy still exist many problems[4]. Most of the commercial detection systems are usually based on signature recognition detection technology. This kind of detection approach shows great advantages for known attacks, but has limitations for those unknown intrusion behaviors. In recent years, neural network has been applied to network intrusion detection by many researchers. Because of its strong self-learning ability, neural network is an effective solution to the rule-based or signature based system problems. For example, BP neural network is classical which shows strong searching ability, but the convergence speed of a single BP network is slow, and being easy to fall into local optimum. While some evolutionary algorithms are good at global searching, it is suitable for optimizing the BP neural network[5]. So, in this paper we try to optimize the BP neural network and use it for network intrusion detection.

The organization of the paper is as follows: Artificial Bee Colony Algorithm(*ABC*) is introduced in part 2. Part 3 is the main contribution of optimizing Neural Network based on *ABC* algorithm. Experiments and related evaluations are shown in part 4. Part 5 gives a conclusion and future work.

ICIS 2014, June 4-6, 2014, Taiyuan, China

II. ARTIFICIAL BEE COLONY ALGORITHM

Artificial bee colony algorithm(*ABC*) is proposed by Karaboga in 2005[6]. The algorithm was originally developed in order to solve the multivariable function optimization problems. The *ABC* algorithm simulates the intelligent foraging behavior of honey bee swarms to achieve intelligence through communication and cooperation between bees with different roles [6,7].

In *ABC* algorithm, there are two basic behaviors with bees. One is when a bee finds an abundant food source, it leads other bees to the food source. The other is bees abandoning a food source and looking for another one. The colony of artificial bees contains three groups: employed bees, onlookers and scouts. For employed bee, they are associated with a particular food source which they are currently exploiting. They record the information about this particular source, for instance, its distance and direction from the nest, the profitability of the source and share this information with a certain probability. The onlookers and scouts, belonging to unemployed bees, are continually at look out for a food source to exploit. For onlooker bees, they wait in the nest and get the food source information from employed bees, and then according to a certain rule, to choose their own employed bees to follow. For scout bee, it is responsible for searching the environment surrounding the nest for new food sources. And the three parts of *ABC* algorithm can be described as Table 1.

TABLE I. THREE PARTS OF ARTIFICIAL BEE COLONY ALGORITHM

Employed bee	Sending the employed bees onto the food sources and then measuring their nectar amounts;
Onlooker bee	Choosing the food sources after sharing the information from employed bees and measuring the nectar amount of the foods;
Scout Bee	Sending the scout bees onto possible food sources and random searching for food sources.

In *ABC* algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. In the algorithm, the number of the employed bees or onlooker bees is equal to the number of solutions in the population. In other words, for every food source, there is only one employed bee.

In *ABC* algorithm, at the initialization stage, a set of food source positions are randomly generated and their nectar amounts are determined. The algorithm generates SN solutions (food source positions), where SN denotes the size of population, calculated by Eq.(1).

$$X_i^j = X_{\min}^j + rand(0,1)(X_{\max}^j - X_{\min}^j) \qquad (1)$$

where X_i^j represents a possible solution to the optimization problem and $j \in \{1,2,...,D\}$, *max* and *min* are the maximum and minimum elements in $\{1,2,...,SN\}$ except for i. $rand(0,1)$ means that it is a random number between $[0,1]$. For each food source, $X_{ij}(i=1,2,...,SN; j=1,2,...,D)$ is a D-dimensional vector. Here, D is the number of optimization parameters. After initialization, supposing the repeated cycles is MCN. An artificial employed or onlooker bee probabilistically produces a modification on the position in its memory for finding a new food source and tests the fitness. The fitness value fit_i is calculated by Eq.(2).

$$fit_i = \begin{cases} \dfrac{1}{1+f_i} & f_i > 0 \\ 1+abs(f_i) & f_i < 0 \end{cases} \qquad (2)$$

where f_i is the objective function value of the solution i. Supposing that if the fitness value of the new food source is higher than the previous one, the bee memorizes the new position and forgets the old one. Otherwise, it keeps the position of the previous one in its memory. After all employed bees finish the searching process, they share the nectar information of the food sources and their position information with the onlooker bees on the dance area. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its fitness value.

An onlooker bee chooses a employed bee to follow a food source, depending on the probability associated with the fitness value P_i, calculated by:

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \qquad (3)$$

where fit_i is the fitness value of the solution i evaluated by its employed bee, which is proportional to the nectar amount of the food source in the position i, and SN is the number of food sources which is equal to the number of employed bees. In this way, the employed bees exchange their information with the onlookers in the dance area.

In order to produce a candidate food position from the old one, employed and onlooker bees use the following equation(4) to calculate the new food source position.

$$V_{ij} = X_{ij} + \varphi_{ij}(X_{ij} - X_{kj}) \qquad (4)$$

where $k = \{1,2,...,SN\}$ and $j = \{1,2,...,D\}$ are determined randomly. The $\varphi_{ij} \in [-1,1]$ is a random number. It controls searching rang of the neighbor food source position around X_{ij}. With the value of $(X_{ij} - X_{kj})$ decreasing, the search rang around X_{ij} decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is adaptively reduced.

In the ABC algorithm, if a position cannot be improved further through a pre-determined number of iterations called *Limit* then that food source is assumed to be abandoned. If the abandoned solution is X_i, the Scout bee produces a new solution by equation (1) to replace the old one. The process of the ABC algorithm is given below:

a) Initialize the population of solutions X_i, set parameters SN, MCN and Limit;

b) Employed bees find new food sources V_{ij} around X_{ij} according to the equation(4);

c) Choose the better solution between V_{ij} and X_{ij} using the greedy policy;

d) Calculate the choosing probability P_i according the equation (2) and (3);

e) Onlooker bees choose a employed bee to follow according to the probability P_i, and product a new solution by equation (4);

f) Onlooker bees choose a best solution between X_{ij} and V_{ij} by greedy criterion;

g) Employed bee abandons a solution, becomes a Scout Bee, then produce a new solution by the equation (1);

h) Record the best solutions, continue until the number of iterations is equal to MCN.

As can be seen from the above process, the *ABC* algorithm has the features of roles division and conversion and a strong ability between individuals in a group team work together.

The *ABC* algorithm is able to find the optimal solution faster, mainly because of the positive feedback mechanism when Employed bees and Onlooker bees finding the best solutions in the algorithm. The positive feedback mechanism can accelerate the convergence speed greatly[6-11].

III. *ABC* OPTIMIZED *BP* NEURAL NETWORK

The ABC algorithm is good at global search with fast convergence speed, and does not require the continuity of the objective function. So in this paper, we use *ABC* algorithm to optimize the neural network. We use *ABC* algorithm to optimize weights and thresholds of BP neural network, not only can improve the neural network generalization mapping capability, but also make the neural network with a faster convergence and strong learning ability. This integration method can take advantage of the strong learning ability of neural network, but also embodies the artificial bee colony algorithm for global optimization features.

The Artificial Neural Network is an intelligent algorithm by simulating the structure of the nervous system and information delivery of biological neural networks[12]. The neural network with a strong self-learning ability, be able to do adaptive calculations, is a large scale nonlinear adaptive systems. For every kinds of neural networks, BP Network is relatively mature. In general, a BP network consists of an input layer, one or more hidden layers and an output layer. The structure of a typical BP neural network is shown in Figure 1.

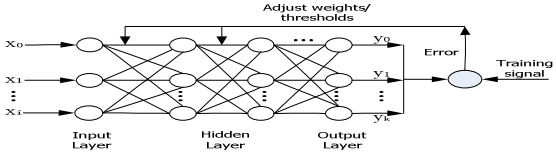

Fig. 1. The architecture of BP neural network

In a *BP* network, the input vector is $X=(x_1,x_2,...,x_n)$, and the input in hidden layer is calculated by:

$$t_j=\sum_{i=1}^{n} w_{ij}x_i-\theta_j \qquad (5)$$

where w_{ij} is the weight, and θ_j is the threshold between the input layer and the hidden layers. The actual output in hidden layer is calculated by:

$$l_j=f\left(t_j=\frac{1}{1+e^{-t_j}}\right) \qquad (6)$$

Similarly, we can get the output of Output layer y_k .The error between the actual and expected output is calculated by the following equation:

$$e^k=\sum_{k=1}^{q}(d_k-y_k)^2 \qquad (7)$$

where d_k is the expected output of the output node k, and q is the number of the output nodes. Moreover, we can derive the adjustment of weights and threshold between the hidden layer and output layer by:

$$\Delta w_{ij}=\partial(d_k-y_k)y_k(1-y_k)y_i \qquad (8)$$

$$\Delta\theta_k=\partial(d_k-y_k)y_k(1-y_k) \qquad (9)$$

Where $\partial\in(0,1)$ is the constant of proportionality, reflecting learning rate when training.

Similarly, we can get the adjustment of weights and threshold between the input layer and hidden layer Δw_{ij} and $\Delta\theta_j$.

If we cannot get the desired results from the output layer, it needs to take backward propagation, adjusting the weights and threshold in the direction of error decreasing, gradually making the error smaller. The detailed process can be found in [12]. The BP neural network has a very strong self-learning ability, and can quickly obtain the optimal solution for unknown data sets. Therefore, the neural network applied to the intrusion detection is a very effective way [13].

However, a single neural network has its limitations when training samples is small and not sufficient, which decrease its accuracy for intrusion detection. More importantly, the neural network obtains the optimal solution by global approximation method with slow convergence speed, and it is easy to fall into the local optimal problems. These factors will affect the effect and accuracy for intrusion detection.

To solve the above problems, we can optimize the neural network. More current research is using evolutionary algorithms. Through combining evolutionary algorithms with neural networks, the neural network can obtain those hardly determined parameters, such as interconnect weights, network architecture and learning rules. It is so called evolutionary neural networks.

Evolutionary Algorithms are a kind of algorithms based on biological evolution, typically Genetic Algorithm (*GA*), Ant Colony Optimization(*ACO*), Particle Swarm Optimization (*PSO*), and Artificial Bee Colony (*ABC*) and so on. They are the probabilistic search algorithm based on the population of iterations to complete the optimization process[14]. Among these evolutionary algorithms, *GA* not only has fast convergence speed but also very versatile, is suitable for solving combinatorial optimization problems. For high-dimensional and high-precision optimization problems, *GA* is easy to reach premature convergence, especially in later iterations[15]; *ACO* is through a positive feedback mechanism between individuals sharing pheromone, which can improve the efficiency of global optimization. However, in *ACO* ants are searching blindness, slow convergence and the lack of pheromone, easily fall into local optimal solution[15]. *PSO* has fast convergence speed and small parameters, but is easy to fall into local optimal solution in later iterations[16].

Comparing with the above relatively sophisticated algorithms, *ABC* algorithm, has a positive feedback mechanism, that can effectively enhance the global searching process. So, ABC algorithm is flexible that easily combined with other technologies to get better algorithm [17,18].

In this paper, based on *BP* neural network and the *ABC* algorithm, we propose the method of the ABC optimized Neural Network algorithm, and apply it to the intrusion detection. In the process of the *ABC* optimization BP neural network, the optimized objects are network weights and thresholds of *BP* neural network. The method can improve *BP* neural network self-learning ability and accelerate convergence speed, so that the neural networks can perform better with higher accuracy in intrusion detection. *ABC* optimized Neural Network process is shown in Figure 2.

The basic idea of using ABC algorithm to optimize BP neural network is that when the BP neural network self-learning training is not able to meet the accuracy requirements, we use ABC algorithm to optimize the BP network operating parameters (including network weights and thresholds).Then BP network utilize the ABC algorithm to optimize its initial values for further training. Thus run *ABC* algorithm and *BP* neural network alternately, until it reaches the required accuracy. Detailed algorithm is given below:

a) Select sample data for training, generate weights W_{ij} *between the Input and Hidden layer, weight* W_{jk} *between the Hidden and Output layer;*

b) Calculate the actual output y_k *by equation (5) (6);*

c) Calculate the Error by equation (7), if it meets the required error, end training and goto e); else goto d);

d) Adjust the weights and thresholds between the Input and Hidden layer, the Hidden and Output layer according to equation (8);

e) According to the results from step d), calculate the weights, W_{ij} *and* W_{jk} *, between the Input and Hidden layer, the Hidden and Output layer respectively;*

f) According to the new weights and sample data, goto a~c), if it meets error requirement, end training; else continue;

g) Using the current weights and thresholds as the input of the ABC, set MCN and Limit. Set the Error of neural network as the target function of equation (2);

h) Run ABC algorithm until iterations reaching MCN. Set the weights and thresholds from ABC as new initial data for BP neural network training and then goto step d);

i) End training; output weights W_{ij} *and* W_{jk} *. Use the obtained model for testing data to get the result.*

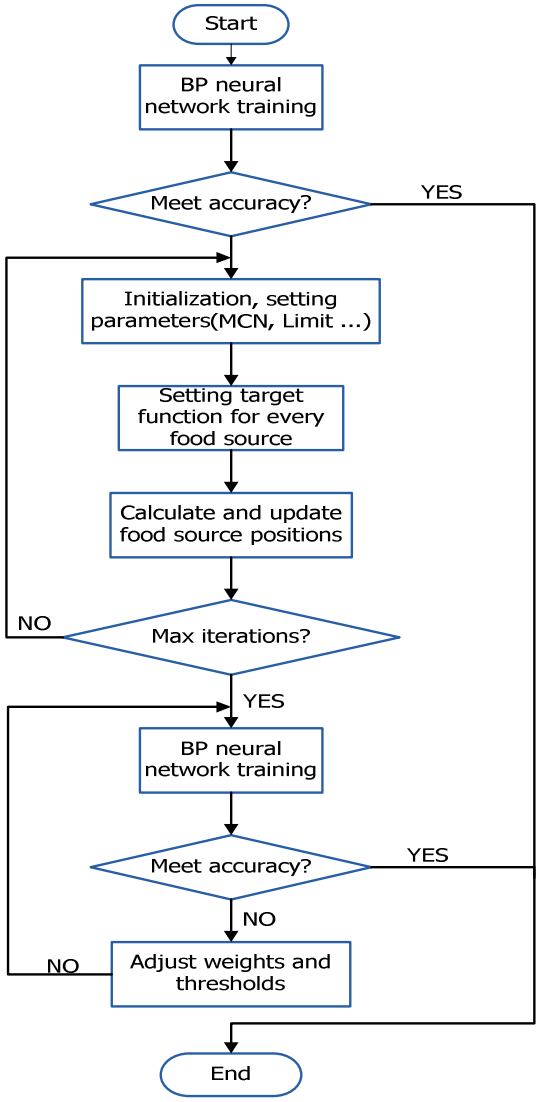

Fig. 2. The integration model of *ABC* optimized *BP* neural network

IV. EXPERIMENTS

A. Experimental Dataset

We use *Matlab* 7.0 simulation platform to evaluate the optimized neural network and conduct two experiments. The first experiment use the optimized neural network. The main purpose is to observe the convergence speed. And the second experiment is comparing the optimized BP algorithm with the original *BP* to evaluate which one is better.

Both experiments use the KDD CUP99 dataset[19], which is provided by United States Massachusetts Institute of Technology Lincoln Laboratory in 1999, a widely use competition data for intrusion detection. There are about 5,000,000 records of total data set, 39 types of attacks, each record described by 41dimentional features.

B. The Convergence Speed of ABC optimized BP neural network

We use 10% KDDCUP99 data set with about 490,000 records and each data record with 41 dimensions. The total data set is divided into 31 categories and each category represents an abnormal type of intrusion detection respectively.

During the experiment, the initial bee colony size is 20, the number of iterations is 200 generations. About the *BP* neural network, input layers are 41 nodes, one hidden layers with 20 nodes and the output layer is 1. The output result is the square error.

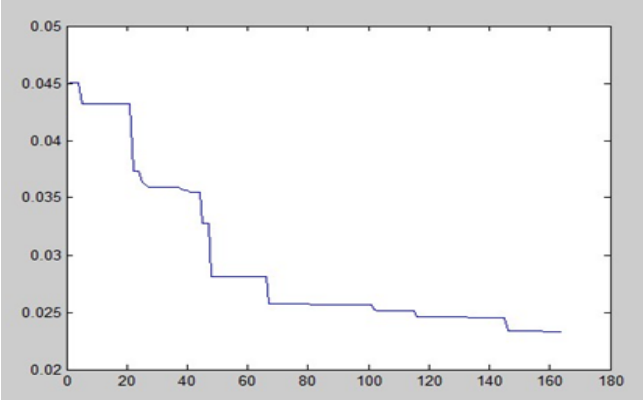

Fig. 3. The result of ABC optimized BP neural network

In figure 3, the *X*-axis represents the number of iterations and the *Y*-axis represents the squared error. From Fig.3, it is obvious that the convergence speed of *ABC* optimized neural network is very fast.

C. ABC optimized BP neural network

In this experiment, we select 4000 data record from the KDDCUP dataset. Each data record has 41 dimensions. There are three kinds of classification results: *Normal*, *Neptune*, S*murf*. Through data smoothing each category has 800, 880 and 2320 records respectively. We set the model parameters as following: the initial bee colony size is 20, the number of iterations is 1000 generations. The BP are the same as the above experiment.

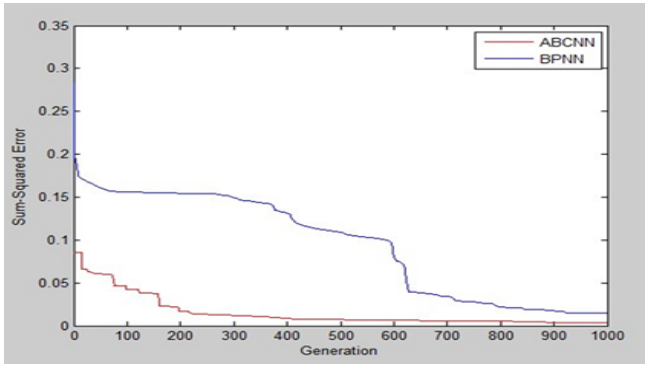

Fig. 4. Comparing the ABC optimized BP neural network with BP neural network

As can be seen from the figure 4, the ABC optimized BP neural network has significantly faster convergence than BP neural network, which shows the advantages of the optimized BP approach.

V. CONCLUSIONS AND FUTURE WORK

Although there are several evolutionary algorithms combined with neural networks used in intrusion detection field, but in our knowledge, ABC optimized BP algorithm has not widely used. We combine the ABC algorithm with *BP* network and the experimental results show its effectiveness.

The *BP* neural network has a very strong self-learning ability, and can obtain the optimal solution for unknown data sets. Therefore, applying the neural network to the intrusion detection is a very effective way. However, single neural network has its limitations when training samples is small and not sufficient, it may decrease its detection accuracy. More importantly, the neural network obtains the optimal solution by global approximation method with slow convergence speed, and it is easy to fall into local optimal solutions. These factors will affect the speed and accuracy of intrusion detection. In this paper, using the *ABC* algorithm with a global search ability and fast convergence to optimize the neural network, can overcome the shortcomings of a single neural network. And the future work include using more data set to evaluate the approach and try to use *ABC* algorithm for other neural networks, such as *RBF*, *Hopfield*, etc.

ACKNOWLEDGMENT

Acknowledgements: This work is partially supported by Shanghai Municipal Natural Science Foundation (13ZR1416100), and Key Principle Project of Shanghai Educational Committee(J50103).

REFERENCES

[1] Symantec. Internet Security Threat Report, Volume 17. In http://www.symantec.com/threatreport/.

[2] D.E. Denning. An intrusion detection model. IEEE Transactions on Software Engineering – Special Issue on Computer Security and Privacy, 1987, 13(2):222-232.

[3] R. Bace, P. Mell. Intrusion detection systems. NIST Special Publication Intrusion Detection Systems, National Institute of Standards and Technology, 2000.

[4] W. Y. Quan. Overview and Forecast of Intrusion Detection System. Communications Technology(In Chinese with English abstract), 2008, 41(11):139-146.

[5] Z. Quan, W. C. Jun, Z.X Min, etc. Several Approaches Used in Intrusion Detection Based on Artificial Intelligence. Application Research of Computers(In Chinese with English abstract), 2007, 24(5): 144-148.

[6] D. Karaboga, C. Ozturk. A novel clustering approach artificial bee colony (ABC) algorithm. Applied Soft Computing, 2011, 11(1):652-657.

[7] M. Magdalene, M. Yannis, Z. Constantin. Honey bees mating optimization algorithm for financial classification problems. Applied Soft Computing, 2010, 10(3): 806-812.

[8] D. Karaboga, B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization. 2007, 39(3):459 -171.

[9] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Computer Engineering Department, Oct. 2005. In http://mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf.

[10] Z. C. Qun. Z. J. Guo, W. Xiang. Overview of research on bee colony algorithms. Application Research of Computers(In Chinese with English abstract), 2011, 28(9):3202-3208.

[11] Y. Jin, M. Liang. Efficient tool for complex optimization problems: bee colony optimization algorithm. Application Research of Computers(In Chinese), 2010, 27(12):4410-4413.

[12] H. L. Qun. The artificial neural network theory, design and Application. Beijing: Chemical Industry Press(In Chinese), 2002.

[13] X. S. Hua. Design of Intrusion Detection System Based on BP Neural Network. Computer and Modernization(In Chinese), 2011:47-50.

[14] W. C. Qiong, Research of Intrusion Detection based on the Neural Networks and Genetic Algorithm. Computer Security(In Chinese), 2010:25-28.

[15] M. Dorigo, V. Maniezzo, A. Colorni. et al. Positive feedback as a search strategy, Technical Report, TR91-16. Milano: Politecnico di Milano, 1991.

[16] J. Kennedy, R. C. Eberhart. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks. IEEE Xplore Digital Library, 1995, Vol.4: 1942-1948.

[17] D. Karaboga, B. Akay, C. Ozturk. Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks. Modeling Decisions for Artifical Intelligence, Lecture Notes in Computer Science, 2007, Vol.4617, 318-329.

[18] D. Karaboga, C. Ozturk. Neural networks training by artificial bee colony algorithm on pattern classification. Neural Network World, 2009, 19(3):279-292.

[19] KDD Cup 1999 Data. The UCI KDD Archive, Information and Computer Science, University of California, Irvine. In http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.