CHEMICAL ENGINEERING TRANSACTIONS
VOL. 51, 2016
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at www.aidic.it/cet
Guest Editors: Tichun Wang, Hongyang Zhang, Lei Tian

ISBN 978-88-95608-43-3; **ISSN** 2283-9216
DOI: 10.3303/CET1651066

Research on Key Technology of Network Intrusion Detection System Based on Improved GA-BPNN Algorithm

Haifeng Hu[a], Weina He[b*]

[a]College of Information Technology, Pingdingshan University, Pingdingshan,Henan 467000, China;
[b]Computer college, Pingdingshan University, Pingdingshan,Henan 467000, China;
heweina_lvyin@163.com

In recent years, with the rapid development of information and network technology, computer and network infrastructure has become a popular target of hacker attacks. The intense demand for electronic commerce has intensified the growth of hacking incidents. Network security is a systematic concept, and the effective security policy or scheme is the primary goal of network information security. So, the intrusion detection is one of the main research directions in the field of network security. Because the intrusion detection system(IDS) has its own limitations and technical difficulties, how to apply all kinds of intelligent artificial intelligence algorithms to intrusion detection technology is the key to improve the efficiency of intrusion detection. In order to solve the problem of traditional intrusion detection algorithm in the presence of high false negative rate and high false positive rate, combined with the advantages of BP neural network algorithm, this paper puts forward a kind of intrusion detection algorithm which is used the genetic algorithm to optimize the BP neural network algorithm. Firstly, we find the most suitable weights of BP neural network by genetic algorithm. Then, we use the optimized BP neural network for model learning and testing. Simulation results show that compared with the traditional network intrusion detection algorithm, the training time is shorter, and the algorithm has better recognition rate and detection rate.

1. Introduction

In recent years, with the rapid development of information and network technology, computer and network infrastructure has become a popular target of hacker attacks. The intense demand for electronic commerce has intensified the growth of hacking incidents. Network security is a systematic concept, and the effective security policy or scheme is the primary goal of network information security. So, the intrusion detection is one of the main research directions in the field of network security. Because the intrusion detection system(IDS) has its own limitations and technical difficulties, how to apply all kinds of intelligent artificial intelligence algorithms to intrusion detection technology is the key to improve the efficiency of intrusion detection(Koc, et al., 2012; Shon, et al., 2006; Zhang, et al., 2008; Srinivas, 2002).

At present, many effective intrusion detection classification models are proposed. Guan Jian and Liu Daxin(2004) use genetic algorithm to search the characteristic space of network data. Through genetic operations, the algorithm produces individuals with a high degree of adaptability, thereby automatically summing up the common attributes of the invasion. Zhang Fengbin, Yang Yongtian, Jiang Ziyang(2004) use the properties of the system as feature vectors, so the normal state of system is distributed in the n-dimensional space. Then, they use the genetic algorithm to cover the abnormal space, so as to establish the anomaly detection model. Ludovi M(1993) also uses the genetic algorithm to analyze the safety audit. The research of intrusion detection technology based on neural network is mainly focused on the selection of intrusion features and the establishment of the detection model. K.Fox et al(1990) establish an intrusion detection model based on the multilayer perceptron and multilayer back propagation neural network, which has achieved good results. Based on the traditional intrusion detection algorithm, the intrusion detection algorithm based on the cloud environment has also been developed rapidly. Mazzariello C et al(2010) propose an intrusion detection system based on the network in the cloud environment, and they define a series of intrusion rules, so as to determine whether the corresponding behaviour belongs to the intrusion behaviour.

Please cite this article as: Hu H.F., He W.N., 2016, Research on key technology of network intrusion detection system based on improved ga-bpnn algorithm, Chemical Engineering Transactions, 51, 391-396 DOI:10.3303/CET1651066

Chirag N. Modil et al (2012) apply the Bias classifier and Snort based network intrusion detection system to the cloud environment, and achieve very good classification results. Pardeep Kumar et al (2011) propose hidden Markov model(HMM) based on clustering method, the cloud intrusion detection technology reduces the amount of data which is detected, then they use the HMM to track the state transition. Anand Kannan et al(2012) propose an intrusion detection technology based on genetic algorithm and support vector machine in cloud network. Vieira et al(2010) propose an intrusion detection system based on grid and cloud computing. This system can be used to quickly analyze the behaviour of users by using feed forward artificial neural network.

2. Intrusion detection system

Distributed intrusion detection system is called hybrid intrusion detection system. Most traditional intrusion detection system (IDS) based on network or host to identify and avoid attacks. In either case, the product is looking for an attack flag, which is a pattern that represents a malicious or suspicious intent. On one hand, when IDS is looking for these patterns in the network, it is based on the network. On the other hand, when IDS is looking for an attack flag in a record file, it is based on the host computer. Distributed intrusion detection system combines the two functions. From a functional point of view, whether the type of IDS is a network or host, it can be divided into two parts which are engine and control center. The engine is used to read the original data and generate events, and the control center is used to display and analyze events. The architecture of intrusion detection system as shown in Figure 1.

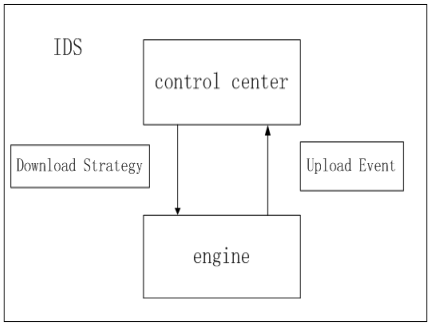

Figure 1: The architecture of intrusion detection system

The engine's main function is to read the original data, analyze the data, match the strategy and process the event, as shown in Figure 2.

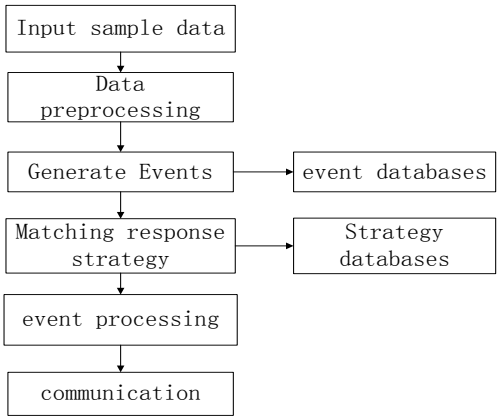

Figure 2: The engine working flow

3. The classification process of GA-BPNN

Error back propagation learning algorithm referred to as BP algorithm, the basic idea is to use the gradient search technology to achieve the minimum mean square error between the network output value and the actual value. The training steps of intrusion detection classifier based on BP neural network are as follows:
(1) Initialization of weights.

(2) Input samples to train the BP neural network.
(3) The first layer is the input layer, and the q th layer is the output layer. The number of neurons in the q th layer is n_q, and the connection weight coefficient of the q th neuron in the i th layer is $w_{ij}^q, (i=1,2,\cdots,n_q)$. Then, the relationship between input and output is:

$$s_i^q = \sum_{i=0}^{n_{q-1}} w_{ij}^q x_i^{q-1} \tag{1}$$

$$x_i^q = f(s_i^q) = \frac{1}{1+e^{-\mu s_i^q}} \tag{2}$$

(4) Calculation of the reverse propagation error of each layer. To adjust the weight of each layer by using the delta learning algorithm. Then, according to the gradient descent method, the weight of the learning algorithm is as follows:

$$w_{ij}^q(k+1) = w_{ij}^q + \alpha D_{ij}^q(k) \tag{3}$$

$$D_{ij}^q = \sum_{p=1}^{p} \delta_{pi}^q x_{pj}^{q-1} \tag{4}$$

$$\delta_{pi}^q = (\sum_{k=1}^{q+1} \delta_{pi}^{q+1} w_{ki}^{q+1}) \mu x_{pi}^q (1-x_{pi}^q) \tag{5}$$

(5) Record the number of samples x that have been learned. If $x<X$, go to step (2) and continue to calculate, otherwise continue the following steps.
(6) Update of weights and thresholds.
(7) Using the new weights to calculate. If the error condition is satisfied or the maximum number of learning is reached, stop learning, otherwise jump to step (2) to continue a new round of learning.
In this paper, genetic algorithm is used to optimize the weights and thresholds of BP neural network, so that the optimized BP neural network can better predict the output value of the function. The selected individual is a string that contains all the weights between input and hidden layers, and the hidden layer and the output layer. At this point, let n be the number of nodes in the network output; y_i is the expected output of the i th node of the BP neural network; o_i is the predicted output of the i th node, and the fitness value is $F = k\sum_{i=1}^{n} |y_i - o_i|$. The smaller the individual fitness value is, the better the individual is. In this way, the preparation of genetic algorithm is completed. Next, we introduce the basic steps of genetic algorithm.
The selective probability of each individual is:

$$cp_i = (k/F_i) \Big/ \sum_{j=1}^{N} f_i \tag{6}$$

As the fitness value is smaller the better, we should calculate the reciprocal of fitness value before selecting individual. Where, N is the number of individuals. We use real number for coding the individual, so the crossover operation is also using the real intersection method. The specific method for exchanging a fragment of the chromosome is:

$$\begin{cases} a_{kj} = a_{kj}(1-b) + a_{ij}b \\ a_{il} = a_{lj}(1-b) + a_{kj}b \end{cases} \tag{7}$$

The mutation process acts on a single individual, and the procedure uses a very small probability to randomly change the value of a bit string. The specific operation method for gene mutating is:

$$a_{ij} = \begin{cases} a_{ij} + (a_{ij} - a_{\max}) f(g) & r \geq 0.5 \\ a_{ij} + (a_{\min} - a_{ij}) f(g) & r < 0.5 \end{cases} \tag{8}$$

In the formula, a_{max} is the upper bound of the a_{ij}, a_{min} is the lower bound of the a_{ij}, m is a random number, g is the number of the current iteration, G_{max} is the maximum number of evolution, and $f(g)$=m(1-g/G_{max},).

4. Simulation experiment and result analysis

4.1 The experimental data

The training and testing datasets are stored by using text format in the KDD CUP99. In this data set, each row represents a record that contains 41 characteristic values and 1 the decision attribute value. Decision attributes represent the categories of each data, and they are only used as judgment conditions in the test. The remaining 41 attributes can be divided into the basic attribute set, the content attribute set, the traffic attribute set and the host traffic attribute set. Among them, there are 8 attributes that are discrete variables, and the remaining attributes are continuous variables. In order to make the data conform to the experimental requirement, the data need to be preprocessed. Data preprocessing includes two steps that are the data processing of symbolic data and the standard processing of numerical data. Since each data contains 3 symbolic attributes which are protocol_type, service, and flag, we should convert them to numeric attributes. The protocol_type property has 3 properties, we can use 1-3 to represent them. The service property contains 70 kinds of values, through the reduction of attributes, we can use 1-9 to represent them. The flag property contains 11 kinds of attributes. We can use 1-4 to represent them by means of attribute reduction. Next, we give the sample data table. It consists of 1000 vectors, each data has 41 characteristics.

Table 1: The sample data set

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X⋯	X40	X41	Y
0	tcp	http	SF	215	45076	0	0	0	0	0	⋯	0	0	Normal
0	tcp	http	SF	162	45076	0	0	0	0	0	⋯	0	0	Normal
43	tcp	telnet	SF	756	369448	0	0	0	0	0	⋯	0	0	Normal
38	tcp	telnet	SF	755	417500	0	0	0	0	0	⋯	0	0	Normal
18	tcp	telnet	SF	154	187774	0	0	0	0	0	⋯	0	0	Normal
3	tcp	smtp	SF	833	345	0	0	0	0	0	⋯	0	0	Normal
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	⋯	0	0	smurf
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	⋯	0	0	smurf
0	udp	dom	SF	35	0	0	0	0	0	0	⋯	0	0	Normal
0	tcp	smtp	SF	571	330	0	0	0	0	0	⋯	0	0	Normal
0	tcp	finger	SF	9	366	0	0	0	0	0	⋯	0	0	Normal
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮
0	tcp	private	S0	0	0	0	0	0	0	0	⋯	0	0	Neptune
0	tcp	private	S0	0	0	0	0	0	0	0	⋯	0	0	Neptune
0	tcp	private	S0	0	0	0	0	0	0	0	⋯	0	0	Neptune

Table 1 reflects the KDD CUP99 data set from different angles. As the dimensions of the various indicators are different, so we cannot make a direct comparison. In order to make the index have comparability, and to speed up the convergence rate of the neural network, this paper has carried on the normalized processing to each index:

For quantitative indicators: the following formula is used to normalize.

$$x_i = \frac{x_i - x_{i\min}}{x_{i\max} - x_{\min}} \tag{9}$$

Where, the normalized values for the *i* th indicator is x_i , the minimum value of the *i* th indicator is x_{imin}, and the maximum value of the *i* th indicator is x_{imax}.

In this paper, the characteristic values are used as the input values of the GA-BPNN model. As the neural network model of this paper is a 41-X-1 model, we carry out the training of the sample according to principle. The principle is that the number of nodes in the hidden layer is $n_i = \sqrt{n+m} + a$. We try to set the number of nodes in the hidden layer to 8, 12, 16, and 20. From the results of training, it can be known that the number of hidden layer nodes is X=16 when the system fitting residual is the smallest.

Next, we use the GA-BPNN model, SVM model and clustering model to detect the 15 sets of data in the table 2, and the experiment is repeated 3 times. The correctness of the classification is shown in Figure 3.

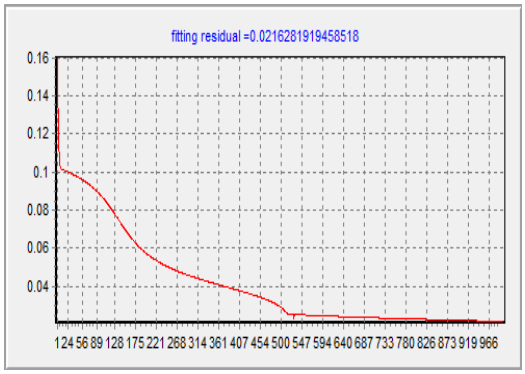

Figure 3: The number of hidden layer nodes is 16 in neural network training

Table 2: The test data set

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X⋯	X40	X41
0	tcp	http	SF	236	622	0	0	0	0	0	1	⋯	0	0
0	tcp	http	SF	253	5230	0	0	0	0	0	1	⋯	0	0
0	tcp	http	SF	232	757	0	0	0	0	0	1	⋯	0	0
0	tcp	http	SF	250	3267	0	0	0	0	0	1	⋯	0	0
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	⋯	0	0
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	⋯	0	0
0	tcp	http	SF	168	36438	0	0	0	0	0	1	⋯	0	0
0	tcp	http	SF	232	337	0	0	0	0	0	1	⋯	0	0
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	⋯	0	0
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	⋯	0	0
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	⋯	0	0
0	tcp	http	SF	248	4739	0	0	0	0	0	1	⋯	0	0
0	tcp	http	SF	236	1667	0	0	0	0	0	1	⋯	0	0
0	tcp	http	SF	222	622	0	0	0	0	0	1	⋯	0	0
0	tcp	http	SF	245	3432	0	0	0	0	0	1	⋯	0	0

4.2 The experiment steps and the result analysis

In order to evaluate the advantages and disadvantages of detection technology, we need a series of quantitative evaluation index. The main evaluation indicators include total detection accuracy, missed alarm rate, false alarm rate, detection delay and learning ability. The total detection accuracy refers to the ratio of the number of correct classification and the total number of samples.

$$\text{total detection accuracy(TDA)} = \frac{\text{The number of correct classification}}{\text{The total number of samples}} \tag{10}$$

This is an evaluation index of the distinguishing ability of sample of intrusion detection technology, which reflects the overall detection ability of intrusion detection technology in a certain extent. We hope that the total detection accuracy as high as possible.

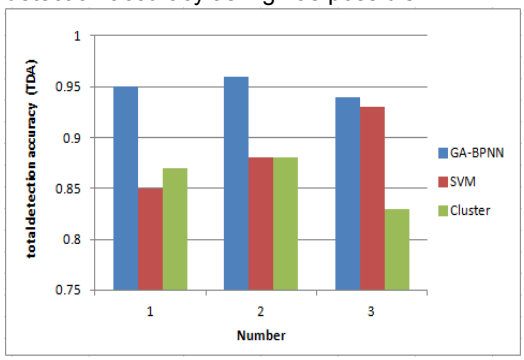

Figure 4. The comparison of detection accuracy

From Figure 4 we can see that the detection accuracy of the type of DoS attack is the highest, and the type of R2L attack is the lowest. This is because the type of R2L attack is often disguised as a legitimate user identity. At the same time, we found that the number of experiments for the analysis of the final results have a greater impact. Through many tests, the final test results are the highest detection accuracy which are obtained from the experiment. In summary, compared with the results of classification of CU99 KDD data set by clustering, SVM algorithm and other methods, the GA-BPNN algorithm can get better classification results. Therefore, it is feasible and effective to use this classification theory in intrusion detection model.

5. Conclusions

In order to solve the problem of traditional intrusion detection algorithm in the presence of high false negative rate and high false positive rate, combined with the advantages of BP neural network algorithm, this paper puts forward a kind of intrusion detection algorithm which is used the genetic algorithm to optimize the BP neural network algorithm. Firstly, we find the most suitable weights of BP neural network by genetic algorithm. Then, we use the optimized BP neural network for model learning and testing. Simulation results show that compared with the traditional network intrusion detection algorithm, the training time is shorter, and the algorithm has better recognition rate and detection rate.

References

Fox K., Henning R., Reed, J. et al., 1990, A neural network approach towards intrusion detection[C]. Proceedings of the 13th National Computer Security Conference, 125-134.

Guan J., Liu D.X, 2004, Study of building misuse detection models based on genetic algorithms[J]. Journal of Harbin Engineering University, 25(1): 80-84.

Kannan A., Maguire Jr G.Q., Sharma A., et al., 2012, Genetic Algorithm based Feature Selection Algorithm for Effective Intrusion Detection in Cloud Networks[C],Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference, 416-423. Doi: 10.1109/ICDMW.2012.56.

Koc L., Mazzuchi T.A., Sarkani S., 2012, A network intrusion detection system based on a Hidden Naive Bayes multiclass classifier [J]. Expert Systems with Applications, 39(18): 13492-13500.

Kumar P., Sehgal V., Shah K., et al., 2011, A novel approach for security in Cloud Computing using Hidden Markov Model and clustering[C]. Information and Communication Technologies (WICT), 2011 World Congress on. IEEE, 810-815. Doi: 10.1109/WICT.2011.6141351.

Ludovi M., 1993, Genetic algorithm, A biologically inspired approach for security audit trails analysis[C]. In: Proc. of the 12th Int Conf. On Computer Safety.

Mazzariello C., Bifulco R., Canonico R., 2010, Integrating a network IDS into an open source Cloud Computing environment[C]. Information Assurance and Security (IAS), 2010 Sixth International Conference on. IEEE, 265-270. Doi: 10.1109/ISIAS.2010.5604069.

Modi C.N., Patel D.R., Patel A., et al., 2012, Bayesian Classifier and Snort based network intrusion detection system in cloud computing[C]. Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on. IEEE, 1-7. Doi: 10.1109/ICCCNT.2012.6396086.

Shon T., Kovah X., Moon J., 2006, Applying genetic algorithm for classifying anomalous TCP/IP packets [J]. Neurocomputing, 69(16-18):2429-2433. Doi: 10.1016/j.neucom.2006.01.023.

Srinivas M., Guadalupe J., Andrew S., 2002, Intrusion Detection Using Neural Networks and Support Vector Machines [J]. Neural Networks, IJCNN 02. Proceedings of the 2002 International Joint Conference on, 2(3): 1702-1707. Doi: 10.1109/IJCNN.2002.1007774.

Vieira K., Schulter A., Westphall C.B., et al., 2010, Intrusion detection for grid and cloud computing[J]. It Professional, 12(4): 38-43. Doi: 10.1109/MITP.2009.89.

Zhang F.B., Yang Y.T., Jiang Z.Y., 2004, Genetic Algorithms in Intrusion Detection Based on Network Anomaly [J]. ACTA ELECTRONICA SINICA, 32(5):875-877.

Zhang J., Zulkernine M., Haque A., 2008, Random-Forests-Based network intrusion detection systems [J]. IEEE Transactions on System, Man, and Cybernetics part B: Cybernetics, 38(5): 649-659. Doi: 10.1109/TSMCC.2008.923876.