ATLANTIS PRESS

Advances in Engineering Research, volume 114

International Conference on Advances in Materials, Machinery, Electrical Engineering (AMMEE 2017)

Research on Improved Genetic Algorithm for Virus Intrusion Detection Model

Peng Zhang

Suzhou Industrial Park Institute of Services Outsourcing, Suzhou, 215123, China

zhangp@siso.edu.cn

Keywords: improved genetic algorithm; intrusion detection model; traditional security model; computer virus.

Abstract. The purpose of this study is to solve the computer security problems. Based on the traditional security model, the hidden dangers of computer security and the advantages and disadvantages of various intrusion detection techniques are analyzed. A computer intrusion detection system based on an improved genetic algorithm is designed. The results show that the optimized genetic algorithm can improve the efficiency of intrusion detection and reduce the false alarm rate. Therefore, we conclude that the application of genetic algorithm in intrusion detection has important theoretical and practical significance.

1. Introduction

With the rapid development of computer technology and communication technology, the application of computer is becoming more and more popular [3]. At the same time, the computer security problem becomes more and more complex. In the late 1960s, people realized the vulnerability of computer system and began to study its security [2]. In the1980s, the performance of computers has been greatly improved. All the isolated computer systems are connected to form a network to realize mutual communication and resource sharing [5]. Especially since the 1990s, the rapid development of the Internet has greatly promoted the social informatization [1]. Based on the computer network environment, it has realized a variety of applications such as electronic banking, e-commerce, e-government, e-household, financial network, manufacturing resource management and network virtual community [4]. However, the network also provides an opportunity for illegal acts of information. The storage, processing and transmission of information is attacked by the intruder or the virus, which results in the paralysis of the system or the loss of important data. For example, hackers break new industrial security mechanisms, and millions of computers are paralyzed for viruses in hours [6]. In a word, computer security becomes more and more important. It is related to business interests, personal privacy and national secrets. Therefore, the computer security issues need to be resolved urgently. It is of great significance for China's aerospace manufacturing industry [8].

From the initial experimental research to today's commercial products, intrusion detection system (IDS) has been developed for more than 20 years. In 1980, Anderson first proposed audit tracking, which could be used to monitor intrusion threats [9]. Although the importance of the idea was not understood at the time, his report was considered groundbreaking work for intrusion detection. From 1984 to 1986, Dorothy Denning and Peter Neumann jointly proposed a real-time intrusion detection system model, which was named IDES (Intrusion Detection Expert System). It is a host based intrusion detection system, which uses statistical methods to detect user abnormal behavior. Their hypothesis is the basis of intrusion detection research. In 1987, Dorothy Denning introduced a generic intrusion detection model [10]. The model consists of three components: event generator, activity recorder and rule set. The event generator is part of the model that provides system activity information. The activity recorder holds the status of the system or network in the monitoring. The rule set is an ordinary check event for checking events and status data. In 1980s, inspired by Anderson and IDES, a large number of IDS systems emerged [11]. However, until the 1990s, the

Copyright © 2017, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

commercial IDS appeared. In foreign countries, commercial intrusion detection systems include ISS's Secure Secure, Nai Cybercop, SVC Net Prowler, CA's Session Wall-3, Cisco's Secure IDS and IBM's IERS (Internet Emergency Response Service). In China, there are few network intrusion software [13]. It mainly includes intrusion detection systems of Netpower, hacker intrusion detection and early warning system. There are still a big gap with foreign. In 1994, Biswanath Mukherjee made a complete review and analysis of previous IDS studies. He analyzed and commented on various IDS prototype systems. After 1995, there were many different new IDS research methods, especially intelligent IDS, which included neural networks, fuzzy recognition, genetic algorithms, immune systems, and data mining [12]. These methods are currently in the theoretical research stage. At present, most commercial intrusion detection systems are similar in principle to virus detection. They have a certain size and number of intrusion feature pattern libraries, which can be updated regularly [7].

2. Materials and Methods

2.1 The Concept of Genetic Algorithm

Genetic algorithm (GA) is a global search algorithm, which randomly generates a population. Then. it simulates the process of natural evolution in a better direction, and evaluates the merits of individuals through fitness. In each generation, the superior individual produces offspring by genetic manipulation, and the inferior individual is eliminated. Individuals in a population are usually composed of a string of binary or real numbers of length. An individual is a solution to the problem, and genetic algorithms are widely used to solve optimization problems.

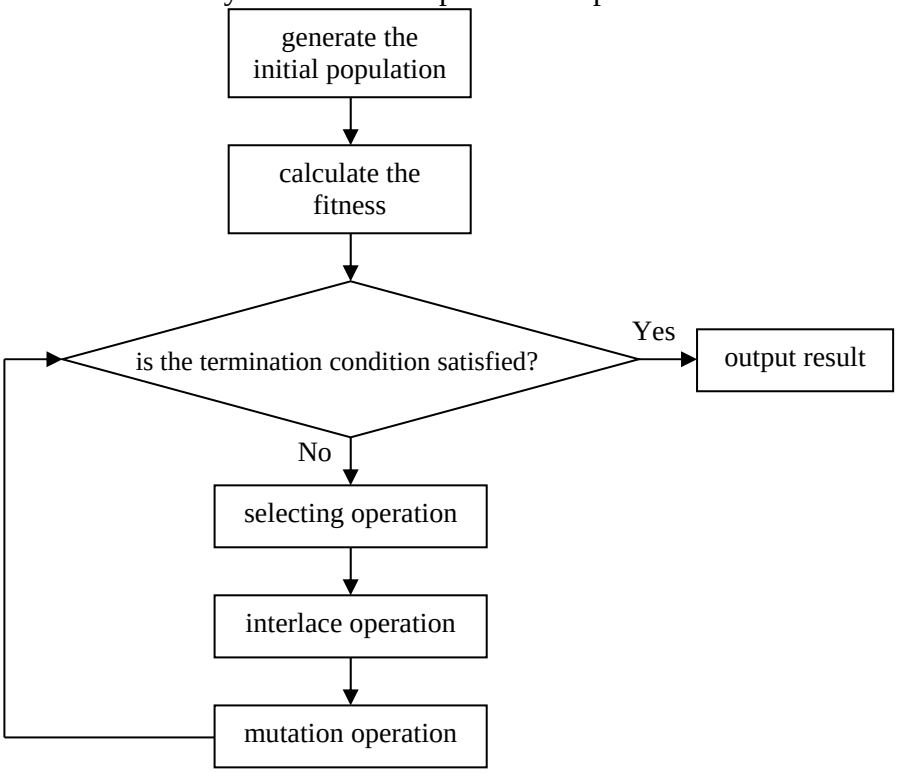

Figure 1. The flow chart of simple genetic algorithm

The genetic algorithm was first proposed by American professor John. Holland of the University of michigan. It originated from the study of adaptive behavior of natural and artificial systems in the 60s. It is a global optimization random search algorithm, which imitates the principles of natural biology, heredity and evolution. It uses bit string coding technique to generate initial population for the problem. Then, the fitness of the population is evaluated, followed by selection, crossover, and mutation operations. A selection strategy based on fitness ratio is used to select individuals in the current population, and crossover and mutation are used to produce the next generation. This generation goes on until it meets the expected conditions. The genetic algorithm can simultaneously

search multiple regions in the solution space by using the method of population organization, and it is especially suitable for large-scale parallel processing. Genetic algorithm has the characteristics of self-organization, self-adaptation and self-learning. The natural selection of the fittest and the simple genetic operations make the computation free from the constraints of the search space (e.g., differentiable, continuous, unimodal) and the absence of additional supporting information (such as guidance). Therefore, the genetic algorithm not only can achieve high efficiency, but also has a simple, easy to operate and common features. At present, with the development of computer technology, genetic algorithm has been paid more and more attention, and has been successfully applied in machine learning, pattern recognition, image processing, combinatorial optimization, VLSI design and optimization control. Genetic algorithm is not a simple random comparison search algorithm. By assessing the fitness of chromosomes and the role of genes in chromosomes, it effectively uses existing information to guide the search for the most promising states to improve the quality of the optimization. The flow chart of simple genetic algorithm is shown in Figure 1.

2.2 Improvement of Genetic Algorithm

In the genetic algorithm, the generation of new generation population mainly depends on the stochastic crossover, recombination and mutation of the previous generation. Therefore, it takes a great deal of cost to obtain the optimal solution. Therefore, genetic algorithm has the disadvantages of premature maturity and weak local search ability, which seriously affect the efficiency of genetic algorithm. Nowadays, the commonly used methods of improvement can be broadly divided into the following categories. First, genetic operators are improved appropriately, including selection, crossover and mutation, etc. in addition, in order to improve the performance of the algorithm, a number of high-level genetic operations have been developed and applied. Second, combining genetic algorithm with traditional heuristic search techniques based on problem knowledge (such as hill climbing, simulated annealing, tabu search, etc.), a framework of hybrid search algorithm based on genetic algorithm is constructed. Third, the genetic algorithm is parallelized. At present, the most basic parallel schemes are synchronous master-slave, asynchronous, synchronous and network. Fourth, the dynamic adaptive technology is used to adjust the algorithm control parameters and coding granularity in the evolution process. The last one is an organic combination of several previous approaches.

This paper mainly studies the use of genetic algorithm to train the neural network, optimize the network connection right, and use it for intrusion detection. Due to the shortcomings of the local search ability of the basic genetic algorithm, it is necessary to improve the genetic algorithm.

3. Results

3.1 Implementation of Improved Genetic Algorithm

The main idea of genetic algorithm to optimize the weights of BP neural network is as follows. First of all, the genetic algorithm is used to optimize the initial weight (threshold) distribution, and a better search space is found in the solution space. Then, BP algorithm is used to search the optimal solution in this smaller solution space. The improved genetic algorithm can optimize the weights of neural networks and prevent the search from being trapped into local minima.

The global characteristics of genetic algorithm and the fast and efficient characteristics of neural networks in local optimization are comprehensively utilized. A local improved GABP algorithm is used to train the neural network. A locally improved genetic algorithm is used to optimize the weights of neural networks. The adaptive learning rate momentum gradient descent algorithm is used to train the neural network, and the fitness function is calculated. Finally, the output of neural network is computed by the weight corresponding to the maximum fitness function. The algorithm flow is shown in Figure 2.

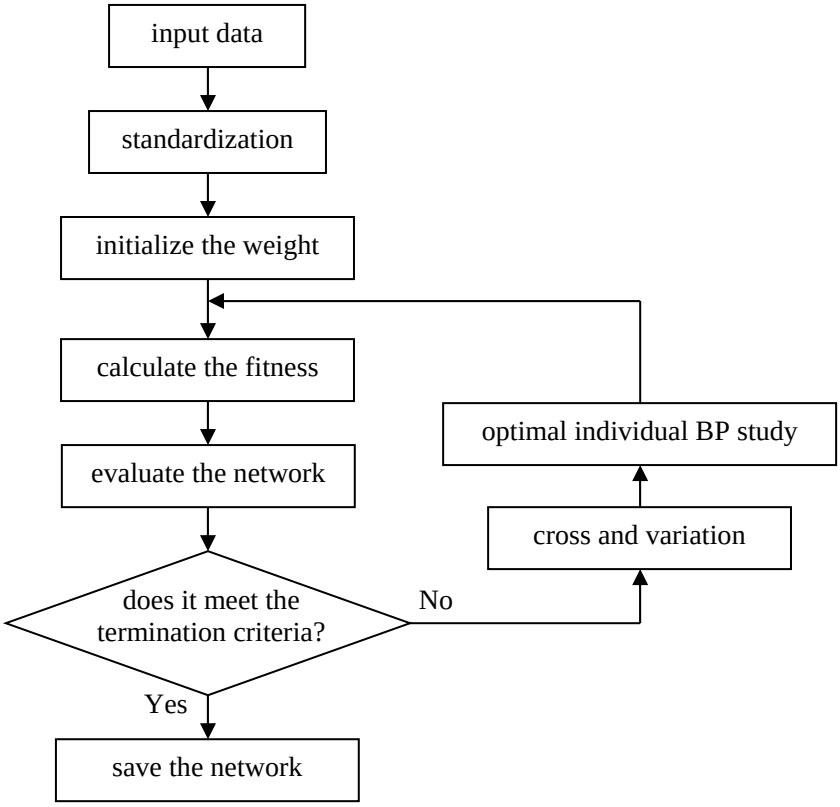

Figure 2. The algorithm flowchart

3.2 Detector Sets

The generation of the detector is calculated and analyzed in a probabilistic manner. If Nx is used to represent the size of the set X, then: N_{R0}: the number of candidate detectors, that is, the size of the detector set prior to review; N_R: the number of valid detectors, that is, the size of the set of detectors after the review; N_S: the number of strings in the "own" collection; P_M: the probability of matching between any two random strings; P_f: false dismissal probability, that is, $P_f=(1-P_M)^{N_R}$; f: the probability that any random string does not match the N_S string in the "own" set, that is, $f=(1-P_M)^{N_S}$.

Because $f=(1-P_M)^{N_S}$, we can get $Inf = N_S \cdot In(1-P_M)$. When the PM is small enough, according to the Taylor expansion:

$$N_{R0} = \frac{-InP_f}{P_M} \cdot e^{P_M N_S} \quad (1)$$

This formula can estimate the number of candidate detectors N_{R0}. N_{R0} is the function of P_f, N_S and P_M.

The minimum set of detectors is obtained by the formula (1). Then, by calculating the minimum, we can get:

$$P_M = \frac{1}{N_S} \quad (2)$$

Therefore, when matching rules $P_M = \frac{1}{N_S}$, there is the minimum value of N_{R0}. At this point, the required set of candidate detectors is minimal, that is, the number of randomly generated strings is minimized.

Formula (1) gives an estimate of N_{R0}, which is related to the size of the protected content N_S, the required detection reliability P_f, and the matching rule P_M. Through further analysis, we can see that the algorithm has the following characteristics. The formula (2) is introduced into formula (1), and the function relation between N_{R0} and N_S is obtained: $N_{R0} = (-e \cdot InP_f) \cdot N_S$. When choosing a suitable detection probability P_f, $(-e \cdot InP_f)$ is a constant. Therefore, N_{R0} is proportional to N_S. Figure 3 shows the minimum N_{R0} dynamic results for each different N_S value. It can also be seen from equation (1) that the size N_{R0} of the detector set is related to the detection capability P_f of the detection system.

The elements of the detector set should have better distribution characteristics, so that neither of the two detectors will match the same "non-self" string. The string space that the detector set R is covered in the collection U is: $P_M \times N_R \times N_U$. For a given failure probability P_f, it requires at least $(1-P_f)\times(N_U - N_S)$ detector in order to cover the "non-self" string space $(1-P_f)\times(N_U - N_S)$ that can be detected. Therefore, we can get:

$$P_M \times N_R \times N_U \geq (1-P_f)\times(N_U - N_S)$$

That is,

$$N_R \geq \frac{(1-P_f)}{P_M} \qquad (N_U > N_S) \tag{3}$$

If P_f and P_M are fixed, then N_{R0} grows exponentially with N_S. Therefore, N_{R0} is very sensitive to N_S, that is, N_S has changed a little, it will lead to a large set of candidate detector changes. For this index factor, we can consider from both positive and negative aspects: the negative side is that it will increase the computational cost of the algorithm; its advantage is that it makes it harder for the intruders to cover up their malicious behavior. For example, if a collection of detectors is generated (possibly using a supercomputer), an attempt to mask its malicious behavior is difficult to succeed if the malicious agent wants to modify the "own" collection, and then modify the set of detectors accordingly.

3.3 Algorithm Test

The system runs on Red Hat Linux 9.0 and windows2000 server platform. The simulation experiments use MATLAB 7.4.0. The trial CPU is AMD Athlon XP 1700+, memory 1G and 120G hard drive. The database uses SQL 2000 Server. The actual network data packet is used as the data source, and the experimental results of GABPIDS in the simulation are obtained, and the detection rate, false alarm rate and false negative rate are calculated. This experiment uses the more authoritative KDD CUP 99 data set for intrusion detection.

The data set contains normal data and a variety of anomaly data, involving a total of four major categories of 22 categories of intrusion attacks: DOS (denial of service attacks); R2L (remote non-authorized access attacks); U2R (unauthorized access to super user rights attacks); PROBE (vulnerability detection and scanning attacks). The kddcup.data.corrected file provided by the dataset contains 4,898,431 raw data, SQL2000 delete duplicate records, get 1,074,992 valid data, the distribution is: Normal (812,814), DOS (247,267), R2L (999), U2R (52), PROBE (13,860).

The data used in the experiment are pretreated by PCA technology. After the 41-dimensional data is processed, the data dimension is reduced to 7, which greatly reduces the dimension of the input data of the neural network and the training and testing time of the neural network, and improves the system effectiveness.

In order to explain the performance of genetic algorithm to optimize the design of neural network more clearly, this paper compares BP algorithm with GABP algorithm. Each algorithm was trained 100 times and compared with the same test data. The convergence times, convergence speed and average detection results of each algorithm are shown in Table 1 and Table 2.

Table 1. The convergence effect statistical

	The number of convergence in 100 training sessions	The convergence speed
Neural network algorithm	66	Slower
Improved genetic algorithm	100	From fast to slow

Table 2. Detection effect statistical

Attack type	Training / test samples	Neural network algorithm		Improved genetic algorithm	
		Average false alarm rate	Average detection rate	Average false alarm rate	Average detection rate
neptune (DOS)	5000/1000	2.5%	95.6%	1.6%	99.2%
smurf (DOS)	5000/1000	4.8 %	92.1%	3.6%	96.5%
ipsweep (PROBE)	5000/1000	5.1%	89.1%	4%	95%
portsweep (PROBE)	5000/1000	7.3%	87.5%	6.8%	94.8%
satan (PROBE)	5000/1000	8.1%	89.3%	6.5%	95.3%
Mixed attack	10000/2000	12.3%	73.5%	10%	85.3%
Mixed attack, test data for new attacks	10000/2000	12.8%	60.5%	10.9%	74.3%

As can be seen from Table 1 and Table 2, compared with the neural network algorithm, the optimized genetic algorithm not only converges fast and converges easily, but also has higher detection rate for various attacks and lower false positive rate. At the same time, compared with other current methods, the detection rate has also improved significantly. It also proves that the performance of the optimized network is improved obviously after the improved genetic algorithm.

The improved genetic algorithm is used to optimize the neural network algorithm, and the simulation test is carried out on the MATLAB platform. Through the preprocessing of real network packets, the input data of the improved genetic algorithm is obtained. The reduction of dimension is reduced by introducing PCA technology, and the structure of neural network is simplified. Neural networks and improved genetic algorithms are used to test the same data. It is found that the improved genetic algorithm is easier to converge, and the convergence speed is faster. It has high classification and recognition rate for normal data. For the data of mixed attack and new attack, although the detection rate of the improved genetic algorithm has been improved significantly, it is still around 75%. For single attack data recognition, especially for Neptune (DOS) attack data, the detection rate is between 98%~100%, and the recognition effect is very good.

4. Conclusion

Combined with the actual data of computer network, an intrusion detection model based on genetic neural network is proposed. Through the data acquisition, preprocessing and feature extraction, the model transforms the computer network data into neural network data. In order to improve the deficiency of the basic genetic algorithm, the genetic algorithm is optimized by the fitness value calibration and the expected value selection method. In this way, the improved genetic algorithm is more suitable for intrusion detection system. Through analysis and research, the following conclusions are obtained. Compared with the traditional genetic algorithm, the improved genetic algorithm has been improved in terms of system efficiency, detection rate and omission rate. At the same time, it improves the ability to detect unknown attacks to a certain extent.

References

[1]. Ahmad, I., Hussain, M., Alghamdi, A., & Alelaiwi, A. (2014). Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components. Neural Computing and Applications, 24(7-8), 1671-1682.

[2]. Bhattacharjee, P. S., Fujail, A. K. M., & Begum, S. A. (2017). Intrusion Detection System for NSL-KDD Data Set using Vectorised Fitness Function in Genetic Algorithm. Advances in Computational Sciences and Technology, 10(2), 235-246.

[3]. Chen, M. H., Chang, P. C., & Wu, J. L. (2016). A population-based incremental learning approach with artificial immune system for network intrusion detection. Engineering Applications of Artificial Intelligence, 51, 171-181.
[4]. Dastanpour, A., Ibrahim, S., & Mashinchi, R. (2014). Using genetic algorithm to supporting artificial neural network for intrusion detection system. In The international conference on computer security and digital investigation (ComSec2014) (pp. 1-13). The Society of Digital Information and Wireless Communication.
[5]. Elhag, S., Fernández, A., Bawakid, A., Alshomrani, S., & Herrera, F. (2015). On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems. Expert Systems with Applications, 42(1), 193-202.
[6]. Ganapathy, S., Kulothungan, K., Muthurajkumar, S., Vijayalakshmi, M., Yogesh, P., & Kannan, A. (2013). Intelligent feature selection and classification techniques for intrusion detection in networks: a survey. EURASIP Journal on Wireless Communications and Networking, 2013(1), 271.
[7]. Hosseinpour, F., Amoli, P. V., Farahnakian, F., Plosila, J., & Hämäläinen, T. (2014). Artificial immune system based intrusion detection: Innate immunity using an unsupervised learning approach. International Journal of Digital Content Technology and its Applications, 8(5), 1.
[8]. Jahromy, B. M., Honarvar, A. R., Saif, M., & Jahromy, M. A. M. (2016). A New Method for Detecting Network Intrusion by Using a Combination of Genetic Algorithm and Support Vector Machine Classifier. Journal of Engineering and Applied Sciences, 100(4), 810-815.
[9]. Mukesh, K. G., Khanna, H. P., & Velvizhi, R. V. (2015). An anamoly based Intrusion Detection System for mobile ad-hoc networks using genetic algorithm based support vector machine. Advances in Natural and Applied Sciences, 9(12), 40-45.
[10].]Moorthy, V. P., & Kumar, S. A. (2014). MCKELM-IDS: efficient feature transformation & optimal feature subset selection based intrusion detection approach using MCKELM. Advances in Natural and Applied Sciences, 8(19), 88-100.
[11]. Shamshirband, S., Amini, A., Anuar, N. B., Kiah, M. L. M., Teh, Y. W., & Furnell, S. (2014). D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks. Measurement, 55, 212-226.
[12]. Singh, T., Verma, S., Kulshrestha, V., & Katiyar, S. (2016, March). Intrusion Detection System Using Genetic Algorithm for Cloud. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies (p. 115). ACM.
[13]. Swaminathan, A., Vivekanandan, P., & Sivajothi, E. (2016). Anomaly Detection Model Based on Multivariate Correlation Analysis Technique to Detect Covert Communication in Wireless Sensor Network. Journal of Computational and Theoretical Nanoscience, 13(8), 5281-5287.