A Back Propagation Neural Network Intrusion Detection System Based on KVM

Jiazuo Wang
Computer Science Department, Bowling Green State University
Bowling Green, OH, United States

Yan Wu
Computer Science Department, Bowling Green State University
Bowling Green, OH, United States

ABSTRACT

A Network Intrusion Detection System (NIDS) monitors a network for malicious activities or policy violations [1]. The Kernel-based Virtual Machine (KVM) is a full virtualization solution for Linux on x86 hardware virtualization extensions [2]. We design and implement a back-propagation network intrusion detection system in KVM. Compared to traditional Back Propagation (BP) NIDS, the Particle Swarm Optimization (PSO) algorithm is applied to improve efficiency. The results show an improved system in terms of recall and precision along with missing detection rates.

***Keywords**—NIDS, KVM, back-propagation, PSO algorithm*

I. INTRODUCTION

The rapid development of cloud computing provides a new computing model for users with powerful and cheap customized services including networks, servers, storage, and applications [2].

A. Background

The continuous improvement of cloud computing technology and competition among cloud providers help users spend less in exchange for enhanced cloud resources. This enables legitimate users to experience greater benefits by using cloud services with better quality. On the other hand, it simultaneously enables malicious users to pay less to contract a substantial configuration of cloud resources for launching an attack, which threatens other legitimate users on the same cloud platform. The existing security technology in cloud computing environments is facing various problems, such as large amounts of data, concurrent access, hardware resource sharing, and network compatibility. Compared with traditional computer systems, cloud computing is more prone to large-scale, dangerous attacks. Users store important information in the cloud platform from which malicious users may launch attacks or steal information. The intruder may also use the powerful computing capabilities of the cloud platform for malicious attacks on its own virtual machine. Typical attacks include the Resident attack, Trojan, and Distributed Denial-of-Service (DDoS) attacks. In other words, with the rapid development of cloud services, cloud computing applications are facing enhanced security threats.

B. Virtual Network Environment Security

The critical structural differences between cloud computing and traditional systems result in traditional intrusion detection techniques not fitting in the cloud computing virtualization environment. All services are provided by the server virtual machine where the security mechanism is deployed. To ensure the safe operation of the virtual machine, the intrusion detection system must be installed on all virtual machines on the same server in the cloud detection system. There are usually many virtual machines on each cloud server, so the above scheme will require a significant amount of the cloud service provider's computer resources, which greatly reduces the overall performance of the platform. To solve this problem, the intrusion detection system is deployed on a privileged virtual machine, which is responsible for the intrusion detection of all other virtual machines on the same server.

Today, mainstream network security technologies include encryption, firewall, and intrusion detection systems. As a common intrusion detection technology, the artificial neural network (ANN) includes capabilities such as nonlinear elastic modeling, strong generalization, learning, and large-scale parallel computing [3]. The forward neural network is one of the most widely used neural networks, and the BP neural network is one of the most commonly used feedforward neural networks. The BP network, also known as the error back propagation network, is a multi-layer mapping network that transmits information forward with the minimum error propagating backward. A single hidden layer BP neural network can approximate any nonlinear function with arbitrary precision. This characteristic makes the BP neural network a common nonlinear detection system.

II. LITERATURE REVIEW

A significant amount of literature exists on cloud computing virtualization security research. Ficco et al. [4] introduce cloud computing in virtual environments and a variety of safety-related research results from different aspects, such as IDS and honeypot. Patel et al. [5] and Su et al. [6] list and analyze a variety of cloud computing environment intrusion detection technologies and the detection strategies. Lee and Yu [7] summarize the detection and defense model in a cloud environment. To ensure the security of a cloud computing network, an intrusion detection system (IDS) acts as the second line of defense in the computer network. It is responsible for processing and analyzing key information from internal and external computer networks to collect and then raise alarms for any violations of the security policy. A new virtual self-checking system is proposed in [8] to protect Kernel-based Virtual Machines (KVM) from a malicious attack on the virtual machine. However, the study needs to establish a complete set of rules in advance. Nantes et al. proposed a way for IDS to establish an efficient model to gain the optimal number of features with reduced usage of computer resources including memory and CPU time [9]. To deal with a large number of network access streams, control data, and applications in the cloud, Dhage and Meshram proposed a new multi-thread distributed intrusion detection model, which effectively integrates knowledge and behavior analysis into intrusion detection while processing, analyzing, and generating a large number of data streams [10]. However, this model is more complex and offers low efficiency.

Also, Rocha and Correia [11] show how malicious insiders can steal confidential data, indicating that the current cloud computing application environment contains many security vulnerabilities. Greamo and Gosh [12] cited the impact of malware on the cloud computing environment, and Hegazy et al. [13] use agent-based technology to describe the framework of intrusion detection in cloud computing. HishamA.Kholidy et al. [1] presents a computational framework for an intrusion detection system (IDS) deployed in all nodes including databases in the cloud, which may cause central server overload, communication, and excessive computation cost of each node.

III. KEY TECHNOLOGY OF KVM AND IDS

A. KVM

Virtualization is the abstraction of computing resources, such as servers, networks, memory, and storage, to enhance functionalities [14]. According to the definition of virtualization, a variety of computer resources are virtual objects, such as software, hardware, and the network. The functions available in the non-virtual environments can nearly all be realized in a virtual environment. Also, the virtual computer resources are merely logical resources for users. KVM is a fully virtualized technology based on the Linux environment and is responsible for the completion of the Linux kernel virtualization features running on x86 and x64 architectures. It is a kernel module in the Linux kernel, which is a virtual machine monitor in the Linux environment.

B. IDS

Two types of intrusion detection methods are common: misuse detection and anomaly detection [15]. Misuse detection first analyzes various possible intrusion behaviors and means, then summarizes the special collection of rules. In the process of detection, the matching rule is used to process the behavior of the detected object and then match the feature set and rule base. If the match is successful, the behavior is considered an intrusion. Anomaly detection records the past normal behavior to establish a normal behavior model. When the behavior of the system is significantly different from the expected normal behavior, it is regarded as an intrusion behavior. Two common techniques used for anomaly detection are neural networks and swarm intelligence algorithms.

Intrusion detection processes usually involve the following: the system first obtains the required sample, then processes and analyzes the sample. Finally, the system proceeds according to the test results.

C. Back-propagation Neural Network

A BP neural network contains an input layer, a hidden layer, and an output layer [16]. The signal propagates forward by passing from the input layer to the output layer. The error is considered back propagating as it modifies the weights and thresholds according to the gradient descent algorithm [16]. The specific process is described as follows.

1) Signal forward propagation

1. Obtain BP network training samples.
2. According to the structure and weight of the network, the signal passes from the input layer to the output layer.
   a. Calculate the output of the hidden layer as
      $$A_h = f(W_1^T X_h - b_1),\ h \in [1, hSize] \qquad (1)$$
   b. Calculate the output layer as
      $$A_o = f(W_2^T X_o - b_2),\ o \in [1, oSize] \qquad (2)$$

In equations (1) and (2), *hSize* is the number of nodes in the hidden layer, *oSize* is the number of nodes in the output layer, W1 and b1 represent the weights and thresholds of the input and hidden layers, respectively, W2 and b2 represent the weights and thresholds of the hidden and output layers, respectively, Ah is the output of the hidden layer, and Ao is the output of the output layer [17].

2) Error back propagation

1. Calculate the mean square error (MSE) value based on the real output and the desired output of the network [17].
2. Adjust the weights and thresholds based on the method of minimizing errors.

The error signal passes from the output layer to the input layer. The weights are adjusted according to the error feedback method, which gradually brings the actual output of the network close to the desired output [17].

$$\Delta w^{(k+1)} = -\eta\, (\partial E^k)/(\partial \omega^k) + \alpha \Delta w^k \qquad (3)$$

In equation (3), $\Delta w^{(k+1)}$ is the modified vector of the first k+1 modification, ηis the learning rate, and E^k is the error function of the neural network [17].

D. PSO algorithm

A BP network is very sensitive to the initial weights and thresholds. If the values and parameters are not set properly, it may cause shock effect and slow the convergence speed [18]. In this paper, PSO is used to search the optimal initial weights and thresholds of a BP neural network. PSO offers simple calculation and good robustness as well as good performance in multi-dimension continuous space, neural network training, combinatorial optimization, and other optimization problems. The primary particle swarm optimization algorithm for position and velocity update [19] is expressed as: $v_i(t+1) = \omega(t)v_i(t)+c_1r_1(pBest_i(t) - x_i(t))+ c_2r_2(gBest(t) - x_i(t))$

In this section, *v* represents speed, *x* represents the location, *i* is the current particle number, *t* is the current number of iterations, c_1 and c_2 are learning factors, r_1 and r_2 are random values in [0,1], PBest is the individual extremum for a single particle, and GBest is the global extremum found for the whole particle swarm [19].

Because of an increasing number of users in cloud computing and the cloud expansion of application environments, a malicious intrusion or attack on a cloud environment can interfere with the availability, confidentiality, and integrity of the resources and services of cloud computing service providers. An

IDS, as a kind of active monitoring technology and protection mechanism, can prevent the destruction of critical IT infrastructure. Intruders can destroy sensitive data and critical applications through network attacks.

IDS can be divided into the two categories of misuse detection and anomaly detection [15]. Previously, IDS could protect the cloud system from various types of attacks, but could not identify suspicious activities in a cloud environment. IDS can also be classified based on the source of the data. Host-based IDS intrusion detection sensors are focused on a single host, while IDS based on the network will put all the sensors on a sensor network segment. The anomaly detection mechanism can improve the ability of the system to detect unknown intrusion attacks, which is especially important in the virtual network model.

IV. Methodology

Based on the analysis of the KVM network structure, this paper proposes an intrusion detection model based on an improved BP neural network and a KVM NAT bridge structure. First, this model analyzes the KVM network model and uses the NAT bridge pattern to design the data capture module to retrieve the virtual machine's communication information. Next, it analyzes and extracts the data packets in turn, then sends the data to the neural network detection engine.

Several modules are included in this model. The data capture module sets the data capture mode based on the network mode of the virtual environment and submits the captured data to the data packet analysis module. The packet-parsing module analyzes the captured data packets based on the network protocol and submits the analytical results to the feature extraction module. Based on the characteristics of the intrusion detection system, the feature extraction module extracts the specific features from the data [20]. Based on the improved BP neural network algorithm, the engine determines whether the data is the invasion of the data, and the results are submitted to the intrusion response module.

Because the performance of the BP algorithm is largely dependent on the initial weights and thresholds, this research to improve the BP network is important to improve the convergence speed of the network. In this report, the PSO algorithm is introduced to optimize the initial weights and thresholds of the BP. The algorithm is based on momentum factor, adaptive learning rate, and PSO.

The design idea of the algorithm is as follows: The system combines the global search ability of the PSO algorithm and the gradient descent local search of the BP algorithm. The PSO algorithm is introduced into the optimization of the initial weights and thresholds of the BP. By using the momentum factor and adaptive learning rate methods, the convergence speed of the BP neural network is accelerated, and the local minimum is avoided. Finally, the algorithm is used to build an intrusion detection system in a cloud environment.

The specific process of the algorithm includes the following steps:

Step 1: initialize the parameters of BP neural network, set the number of nodes in each layer.

Step 2: initialize the parameters of the particle swarm and calculate the particle dimension D. The system initializes the cluster and generates parameters, such as the initial position and the velocity of the particle.

Step 3: calculate the fitness value of each particle compared with the current best fitness value, PBest. If the value is better, update PBest. Then, compare the PBest with the optimal global value of GBest. If the value is better, then use PBest to update GBest.

Step 4: update the inertia weight and adjust the position and speed of the particle.

Step 5: if current iteration achieves the maximum or error is in the scope, the initial weights and thresholds should be the current global extreme GBest, otherwise, return to Step 3.

Step 6: use the system to train the BP neural network, and establish the intrusion detection model with the initial weights and threshold optimization based on the value from Step 5.

A flowchart of the algorithm is shown in Fig. 1.

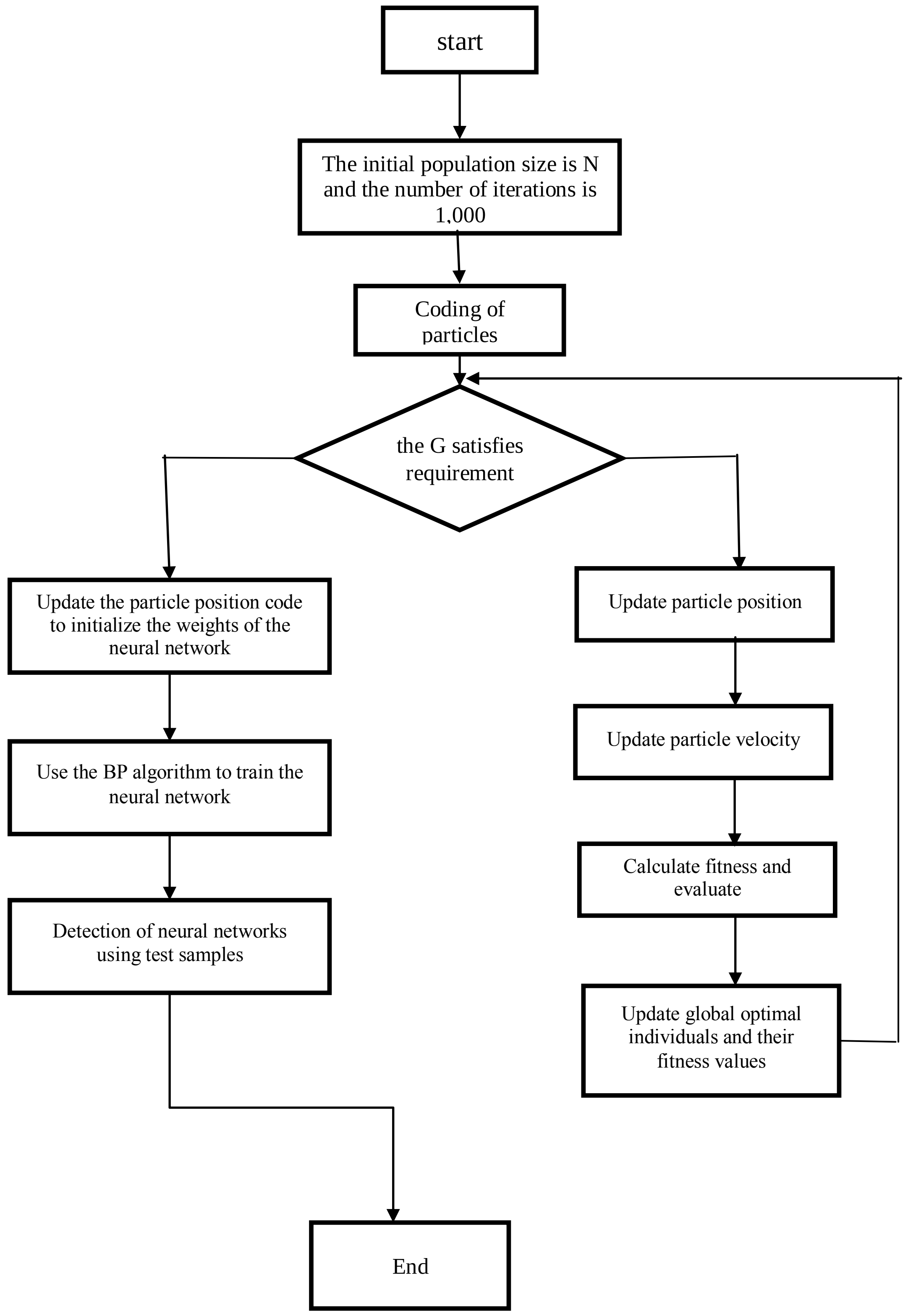

start
The initial population size is N and the number of iterations is 1,000
Coding of particles
the G satisfies requirement
Update the particle position code to initialize the weights of the neural network
Use the BP algorithm to train the neural network
Detection of neural networks using test samples
Update particle position
Update particle velocity
Calculate fitness and evaluate
Update global optimal individuals and their fitness values
End

Figure 1. Structure of algorithm

To illustrate the feasibility and effectiveness of this algorithm, its performance is analyzed with experiments using the intrusion detection dataset KDD Cup 99 [21]. This is a commonly-used intrusion detection algorithm training and testing data set, which includes the majority of the types of attacks faced by cloud computing virtualization environments.

Each sample (connection record) in the KDD dataset contains 42 attributes, and their details and serial numbers are presented in Table I.

TABLE I.PROPERTIES OF RECORDS [21].

SID	property	SID	property	SID	property
1	Duration	15	Su_attempted	29	Sam_srv_rate
2	Protocol_type	16	Num_root	30	Diff_srv_rate
3	Service	17	Num_file_creations	31	Srv_diff_host_rate
4	Flag	18	Num_shells	32	Dst_host_count
5	Src_bytes	19	Num_access_file	33	Dst_host_srv_count
6	Dst_bytes	20	Num_outbound_cmds	34	Dst_host_same_srv_rate
7	Land	21	Is_hot_login	35	Dst_host_diff_srv_rate
8	Wrong_fragment	22	Is_guest_login	36	Dst_host_same_src_prot_rate
9	Urgent	23	Count	37	Dst_host_srv_diff_host_rate
10	Hot	24	Srv_cound	38	Dst_host_ serror_rate
11	Num_failed_logins	25	Serror_rate	39	Dst_host_srv_serror_rate
12	Logged_in	26	Srv_serror_rate	40	Dst_host_rerror_rate
13	Num_compromised	27	Rerror_rate	41	Dst_host_srv_rerror_rate
14	Root_shell	28	Srv_rerror_rate	42	Normal_or_attack

There is a significant difference between the values and data types of each attribute for each original sample in the KDD dataset. In addition, the range of the BP neural network training data and test data should be consistent with the range of the activation function of the neurons in each layer. Also, the input attributes should be numeric values. Therefore, to apply the KDD data to the BP neural network detection model, we need to preprocess the original KDD data samples.

The neural network is trained and tested using the test data, and the experimental results are subsequently compared with the ordinary BP algorithm. The metrics include accuracy, precision, and recall. Accuracy is defined as the ratio of the number of samples correctly classified to the total sample size for a given test set. Precision shows the percentage of positive samples among all reported. The recall rate is for the original sample indicating how many of the positive samples in the sample are correctly predicted.

In this experiment, three layers of the BP neural network are selected, and the number of neuron in each layer is all in the order of 1. The input layer and the hidden layer activation function use a tangent S-type function. Since the number of samples is 42, the number of neurons in the input layer is also 42. If the output results can be judged as intrusion behavior, the number of neurons in the output layer is 1, thus obtaining o = 1. The number of neurons in the hidden layer settings has no normative theoretical guidance. The usual method is based on many experiments to determine the appropriate number of neurons. After the experiment, the hidden layer node is set to 22.

The particle size of the PSO algorithm is determined according to the particle coding and obtain the dimension D = 969. In this paper, the population size N = 30, the maximum velocity of the particle is set to 3, the minimum speed is -3, and the search space of the particle is set to [-1,1]. The velocity and trajectory of the particles depend on the set of learning factors G, O and the inertia weight W, which have great influence on the global search ability and convergence speed of the particle swarm. To ensure better global search ability and local search performance, we set $C_1 = C_2 = 2$ *and* $w = 0.7$.

V. Results

First, the performance of the PSO algorithm is analyzed by looking at the change tendency of the fitness function of the PSO algorithm for the training data. We used the training data to carry out 50 experiments on the PSO, and the average values were calculated. The results show that the MSE value reaches the minimum value of 0.015 when the PSO algorithm runs in about 150 iterations, which is the optimal particle, and it can be decoded as the initial weight and threshold value of the artificial neural network.

To analyze the performance of the system, ten experiments were carried out. The accuracy results of these two algorithms are shown in Fig. 2, precision results are shown in Fig. 3, and recall results are shown in Fig. 4.

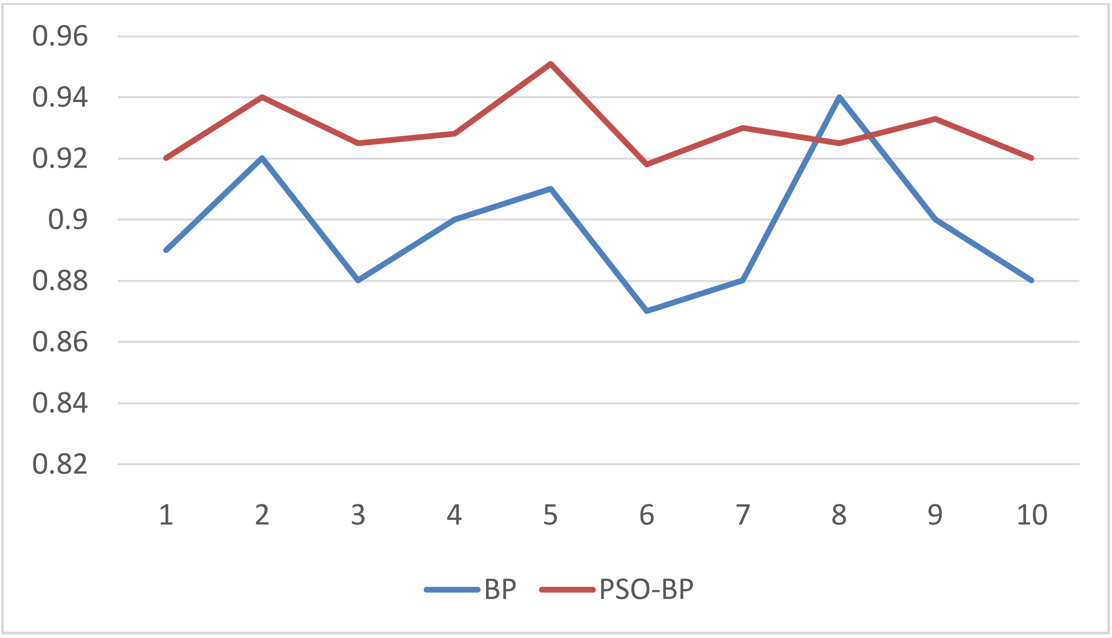
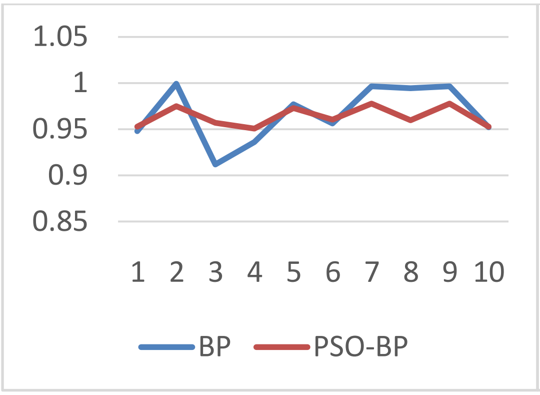


Figure 2. The accuracy of the two algorithms.



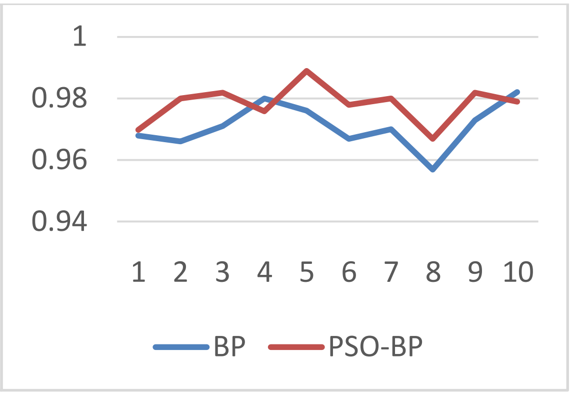


Figure 3. The precision of the two algorithms.

Figure 4. The recall of two algorithms.

As seen from the figures, the accuracy of the PSO-BP algorithm is slightly higher than the pure BP algorithm. First, the momentum factor and the adaptive rate algorithm introduced by the PSO-BP algorithm accelerated the convergence speed of the BP algorithm and avoided falling into a local minimum. Second, the PSO algorithm has a significant advantage in global optimization to enable a more stable precision compared to the traditional BP algorithm. Also, as seen in Figure 4, the recall of the PSO-BP algorithm is slightly better than the traditional BP algorithm.

In summary, the overall detection performance of the proposed PSO-BP detection algorithm is superior to the traditional BP detection algorithm.

VI. CONCLUSION

This paper presents a virtual intrusion detection model based on different types of virtual network structures in KVM. The model is implemented in the KVM network model of a NAT bridge, including the data capture, packet parsing, feature extraction, and neural network detection modules. It is compatible with different network modes in KVM virtualization environments and can capture, process, and analyze the virtual machine communication data flow under different network modes. It provides intrusion detection services for a cloud computing virtualization environment and responds to the attacks of malicious virtual machines.

Aiming at the problem that a BP algorithm is easy to fall into a local minimum [22], this paper proposes an improved BP detection algorithm based on the PSO algorithm, which combines the global search ability of the PSO algorithm and the gradient descent local search of the BP algorithm. The PSO algorithm is introduced to optimize the initial weights and threshold values of the BP algorithm [23]. By using momentum factor and adaptive learning rate method [24], the convergence speed of BP neural network is accelerated and prevented from falling into a local minimum, which improves the detection performance of the algorithm. The experimental results show that the average detection rate of the improved algorithm is higher. It has better detection performance, and can effectively and reliably provide intrusion detection services for cloud computing environments.

REFERENCES

[1] H. A. Kholidy and F. Baiardi, "CIDS: A framework for intrusion detection in cloud systems," *2012 Ninth International Conference on Information Technology: New Generations (ITNG)*, IEEE Press, Las Vegas, Nevada, USA, April 16-18, 2012, pp. 379-385.

[2] S. de Chaves, C. Westphall, and F. Lamin. "SLA Perspective in Security Management for Cloud Computing," *2010 Sixth International Conference on Networking and Services*, 2010.

[3] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42-57, 2013.

[4] M. Ficco, L. Tasquier, and R. Aversa, "Intrusion Detection in Cloud Computing," *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2013.

[5] A. Patel, M. Taghavi, K. Bakhtiyari, and J. Celestino Júnior, "An intrusion detection and prevention system in cloud computing: A systematic review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 25-41, 2013.

[6] Chien-Chung Su, Ko-Ming Chang, Yau-Hwang Kuo, and Mong-Fong Horng, "The new intrusion prevention and detection approaches for clustering-based sensor networks," *IEEE Wireless Communications and Networking Conference*, 2005.

[7] Sheng-Wei Lee and Fang Yu, "Securing KVM-Based Cloud Systems via Virtualization Introspection," *2014 47th Hawaii International Conference on System Sciences*, 2014.

[8] H. Wang, H. Zhou, and C. Wang, "Virtual Machine-based Intrusion Detection System Framework in Cloud Computing Environment," *Journal of Computers*, vol. 7, no. 10, 2012.

[9] A. Nantes, R. Brown, and F. Maire, "Neural network-based detection of virtual environment anomalies," *Neural Computing and Applications*, vol. 23, no. 6, pp. 1711-1728, 2012.

[10] S. Dhage and B. Meshram, "Intrusion detection system in cloud computing environment," *International Journal of Cloud Computing*, vol. 1, no. 23, p. 261, 2012.

[11] F. Rocha and M. Correia, "Lucy in the sky without diamonds: Stealing confidential data in the cloud," *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2011.

[12] C. Greamo and A. Ghosh, "Sandboxing and Virtualization: Modern Tools for Combating Malware," *IEEE Security & Privacy Magazine*, vol. 9, no. 2, pp. 79-82, 2011.

[13] I. Hegazy, T. Al-Arif, Z. Fayed, and H. Faheem, "A multi-agent based system for intrusion detection," *IEEE Potentials*, vol. 22, no. 4, pp. 28-31, 2003.

[14] M. Laureano, C. Maziero, and E. Jamhour, "Intrusion detection in virtual machine environments," *Proceedings of the 30th Euromicro Conference*, 2004.

[15] X. Zhang, Q. Li, S. Qing, and H. Zhang, "VNIDA: Building an IDS Architecture Using VMM-Based Non-Intrusive Approach," *First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008),* 2008.

[16] J. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications*, vol. 40, no. 14, pp. 5501-5506, 2013.

[17] L. Wang, Y. Zeng, J. Zhang, W. Huang, and Y. Bao, "The Criticality of Spare Parts Evaluating Model Using Artificial Neural Network Approach," *Computational Science – ICCS 2006*, pp. 728-735, 2006.

[18] A. Aslanargun, M. Mammadov, B. Yazici, and S. Yolacan, "Comparison of ARIMA, neural networks and hybrid models in time series: tourist arrival forecasting," *Journal of Statistical Computation and Simulation*, vol. 77, no. 1, pp. 29-53, 2007.

[19] J. Zhang, J. Zhang, T. Lok, and M. Lyu, "A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1026-1037, 2007.

[20] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," *Proceedings of the 2004 Congress on Evolutionary Computation* (IEEE Cat. No.04TH8753).

[21] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995.

[22] S. Bharadwaja, W. Sun, M. Niamat, and F. Shen, "Collabra: A Xen Hypervisor Based Collaborative Intrusion Detection System," *2011 Eighth International Conference on Information Technology: New Generations*, 2011.

[23] P. Angin, B. Bhargava, R. Ranchal, N. Singh, M. Linderman, L. Othmane, and L. Lilien, "An Entity-Centric Approach for Privacy and Identity Management in Cloud Computing," *29th IEEE Symposium on Reliable Distributed Systems*, 2010.

[24] J. Arshad, P. Townend, and J. Xu, "An automatic intrusion diagnosis approach for clouds," *International Journal of Automation and Computing*, vol. 8, no. 3, pp. 286-296, 2011.