Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 792 – 802

doi:10.3906/elk-1706-163

Insider threat detection of adaptive optimization DBN for behavior logs

Jiange ZHANG*, Yue CHEN, Ankang JU
State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, P.R. China

Received: 13.06.2017 • **Accepted/Published Online:** 15.11.2017 • **Final Version:** 30.03.2018

Abstract: For the problems of insider threats such as great harm due to damage and resultant loss, difficulty in extracting abnormal behavior features of insiders because of transparency and concealment, and low detection rate, an insider threat detection model using adaptive optimization DBN for behavior logs is put forward. The model carries out deep learning based on the integrated and normalized behavior logs to fully learn normal and abnormal behavior features of insiders to form optimal representations of the behavior features of insiders. The experimental results show that the multiple-hidden-layer deep learning model can fully learn the behavior features of insiders, improving the detection rate of insider threat. Particularly, the adaptive optimization method of the golden section is better than that using the dichotomy method, which can increase the threat detection rate of the DBN model to 97.872%, with more significant advantages.

Key words: Behavior logs, adaptive optimization DBN, insider threat detection, golden section

1. Introduction

Insider threats pose great harm and severe situations, and their damage and resultant losses are as serious as those of outsider security incidents. Unfortunately, it seems that every organization and company must address potential insider threats. Although the number of malicious insiders is assumed to be quite small, the security risks and financial damage they can cause to their employers, and at times to society, is substantial [1]. Surveys such as the e-crime watch survey reveal that current or former employees and contractors are the second greatest security threat, exceeded only by hackers, and that the number of security incidents has increased geometrically in recent years [2]. Hence, the problem of insider threat has gained increasing attention in the computer science community as well as government and industry [3].

Insider threats come from internal users, which feature transparency, concealment, high risk, etc., making it difficult to detect and prevent them. Especially with the rapid development of cloud computing, big data, and other technologies, a large amount of data is stored in the cloud or data centers, which potentially leads to more leakage of user data, thereby further increasing the harmfulness of the insider threats. Therefore, in order to reduce the loss caused by insider threats, research and detection of insider threats have become two of the most important research hotspots in the field of information security.

Fortunately, the development of deep learning provides an opportunity for insider threat detection. Through building up a hierarchical model structure similar to the human brain, deep leaning extracts the features of the input data step by step from the bottom to the top level, thus establishing a mapping relation from the bottom signal to the top-level semantics. Deep learning has been applied in many fields, such as image

*Correspondence: jiangezh@126.com

recognition, speech recognition, and video recognition. Similarly, deep learning has also been applied in the field of information security [4,5], which improves the detection rate of security incidents and reduces economic losses. Moreover, it has also been applied in the field of insider threat detection [6], but its application there is relatively limited compared with other fields. Furthermore, optimization theory, like dichotomy (dividing an interval into two equal subintervals and gradually reducing the number of hidden nodes) and golden section (dividing an interval into two unequal subintervals where the ratio of the greater part to the whole part is equal to that of the smaller part to the greater part, which is 0.618), provides a good theoretical basis for the optimization of network structure construction [7]. In view of the advantages of deep leaning with sufficient learning features and a relatively high detection rate, this paper adopts a deep network structure to learn the behaviors of insiders, which can learn both the essence of normal behaviors and abnormal behaviors, fully characterizing the rich internal information of the data, to form an adaptive optimization DBN (deep belief net) model for insider threat detection, adaptively selecting the optimal network structure for threat detection and providing the rate of the insider threat detection as much as possible.

The main contribution of the paper is to apply a deep learning model to insider threat detection, which can optimize the network structure of the DBN adaptively, thus improving the threat detection rate. First it studies the behavior logs of insiders, extracts the characteristics of the behaviors, and conducts integration and normalization; second, it studies the deep learning network model DBN; third, it analyzes a method using the deep learning network model DBN for threat detection; and finally it carries out experiments on the Computer Emergency Response Team-Insider Threat (CERT-IT) dataset and discusses the selection of batch size, restricted Boltzmann machine (RBM) iterations, the number of units in hidden layers, and the depth of hidden layer, using the dichotomy method and golden section method with regard to three aspects, i.e. the true negative rate, the true positive rate, and the accuracy rate, and makes a comparison between the dichotomy method and the golden section method.

This paper is organized as follows: the second part studies the technologies related to insider threat detection, the third part analyzes the network model DBN for deep learning, the fourth part designs the insider threat detection model for adaptive optimization DBN, the fifth part carries out experiments on the adaptive optimization DBN model and analyzes the experimental results, and the sixth part presents the conclusion and next steps for research work.

2. Deep learning models

There are various types of deep learning models [8], such as the convolutional neural network (CNN) [9], DBN [10,11], and stacked autoencoder (SAE) [12]. The CNN is a supervised learning model, while DBN and SAE are unsupervised learning models. This paper focuses on the research of the unsupervised deep learning model DBN.

The DBN network model includes multiple RBM layers [13] and a backpropagation (BP) network layer, as shown in Figure 1. The basic concept of the DBN network model is that each network layer performs the feature learning by using the unsupervised method; based on the previous training, each layer is trained by using the unsupervised learning method and the training results are adopted as the input of the next layer; and, finally, the entire network is fine-tuned by using the supervised training, making the input and output of the model as similar as possible.

The essence of deep learning using the DBN network model is to fully learn the features by training the parameters of the network structure, and the core is the RBM structure, which is shown in Figure 2.

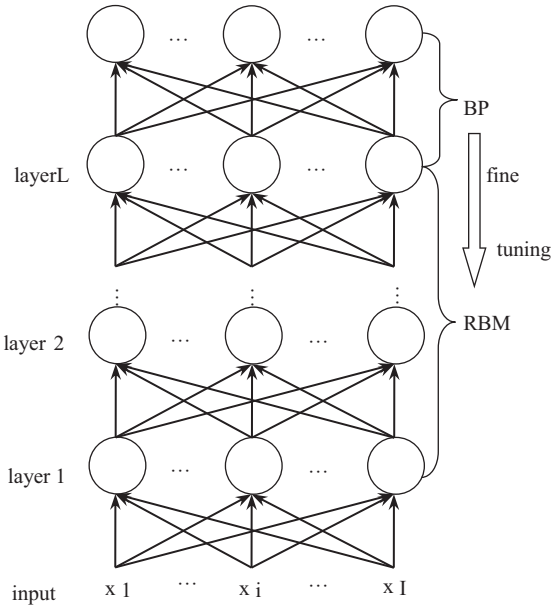

Figure 1. DBN model.

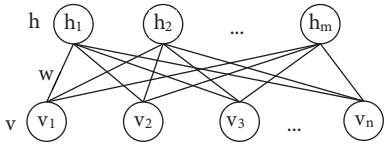

Figure 2. RBM structure.

The structure of the RBM consists of two parts: a visible layer (v) and a hidden layer (h). Nodes of the same layer are not connected, while the nodes between the two layers are connected with each other, and the joint probability distribution P(v,h) satisfies the Boltzmann distribution. Therefore, the visible layer and the hidden layer of the RBM structure can be represented by each other. Given the hidden layer, the probability of the visible layer P(v|h) is represented by Eqs. (1) and (2); given the visible layer, the probability of the hidden layer P(h|v) is represented by Eqs. (3) and (4).

$$P(v|h) = \prod_i P(v_i|h) \tag{1}$$

$$P(v_{\mathrm{i}} = 1|h) = \frac{1}{1 + \exp(-\sum_j w_{ij} h_j - c_i)} \tag{2}$$

$$P(\mathrm{h}|v) = \prod_j P(h_j|v) \tag{3}$$

$$P(\mathrm{h_j} = 1|v) = \frac{1}{1 + \exp(-\sum_i w_{ij} v_i - b_j)} \tag{4}$$

Here, b is the bias of the visible layer, c is the bias of the hidden layer, and the activation function is a sigma function.

The training process of the DBN model is to continuously update the weight parameter $\theta = \{w_1, w_2, ..., b_1, b_2, ..., c_1, c_2, ...\}$ to make the joint probability distribution P(v,h) of the visible layer vector v and the hidden layer vector h maximum [14], and then the slope $\frac{\partial \log P(v,h)}{\partial \theta}$ of the joint probability distribution can be calculated. The updating formula of the weight parameters θ is:

$$\theta_{\tau+1} = \theta_\tau + \eta \frac{\partial \log P(v,h)}{\partial \theta}|_{\theta_\tau} \tag{5}$$

Here, τ and η represent the number of iterations and the learning rate in the RBM training process, respectively.

According to the rules of contrastive divergence (CD) [15], the calculation formulas of each component of the parameter θ are as follows.

$$\mathrm{w}_{\tau+1} = w_\tau + \eta(< v_i h_j >_{data} - < v_i h_j >_{confabulation}) \tag{6}$$

$$\mathrm{b}_{\tau+1} = b_\tau + \eta(< v_i >_{data} - < v_i >_{confabulation}) \tag{7}$$

$$\mathrm{c}_{\tau+1} = c_\tau + \eta(< h_j >_{data} - < h_j >_{confabulation}) \tag{8}$$

Here, the subscript "data" refers to the original data and the subscript "confabulation" refers to the data calculated by using the network model.

3. Insider threat detection of adaptive optimization DBN

3.1. Insider threat detection process of adaptive optimization DBN

There are 4 stages of insider threat detection based on the adaptive optimization DBN, which are insider behavior log collection, insider behavior log preprocessing, deep learning of insider behavior features, and insider behavior log classification. The first stage is to collect the original logs of the insiders. Since these original logs exist in the form of information, they need to be preprocessed and transformed into a standard and processable numerical form. The preprocessed data are used as the input of the deep learning model. The deep learning model fully learns the input data features by adjusting the network weights and finally outputs the feature expression of the data as the classification evidence for the fourth stage.

(1) Insider threat detection preprocess of adaptive optimization DBN

The task in this stage is to transform the original logs in the form of information into a standardized and processable numerical form, thereby providing the raw material for deep learning in the next stage. This stage mainly includes log integration and log normalization.

1) Log integration: the behavior logs of insiders are recorded in different logs according to different types of the behaviors. However, a threat is often caused by a combination of various types of behaviors. Therefore, the behavior logs of insiders need to be integrated according to the type of the behavior classification. The behavior logs of the insiders are mostly recorded with a 4-tuple format, which includes the time of the occurrence of the behaviors, the subject of the behaviors, the host that produces the behaviors, and the specific behaviors. The first 3 items are common items to all types of logs, which are easy to integrate, while the specific behaviors of the fourth item will be different according to the type of behaviors, which are more difficult to integrate. Therefore, the original log data are integrated by adopting a method based on the type of behaviors, which is defined as follows: suppose that n types of logs b_1, b_2, ... b_n of the insider are recorded in an organization, and for any type of behavior b_i, the item will be 0 if it is not this type of behavior; otherwise, it is a specific value for that type of behavior. Thus, if n types of logs of the insiders are recorded in an organization, the behavior logs of the insiders in the organization can be represented by n+3 tuple after integration, and each record of each type of the original log can be mapped to the integrated $n+3$ tuple.

2) Log normalization: it is necessary to normalize the data when using the deep learning model to perform feature learning. In this paper, the 1/N code discretization method is used for normalization. For the 1/N code discretization method, if an attribute value is N, the attribute can be represented by a vector containing N binary values, wherein an exclusive element representing the type of the attribute value is set to 1 and the values of other elements are all 0. Normalization will bring convenience to the data processing in the next stage and it also guarantee the quick convergence of the deep learning model.

3) Adaptive optimization deep learning of the features of the insider behaviors

The normalized data are formed after the preprocessing of the behavior logs of the insiders, on which the deep network structure is adopted to train the behaviors of the insiders, so as to fully study the normal behaviors and abnormal behaviors of the insiders. The unsupervised learning model DBN is adopted for training. The DBN network model carries out the pretraining on a network structure with multiple RBM hidden layers by using the activation function of sigma and the CD rules, and then it carries out the reverse propagation by using the BP neural network so as to fine-tune the parameters of network structure to form an optimized DBN model.

(2) Classification of insider threat detection for adaptive optimization DBN

An optimized network model is formed after the deep learning of the behaviors of the insiders, which allows for the classification of the behaviors of the insiders. There are two kinds of classification for the deep learning, i.e. biclassification and multiclassification. Generally, biclassification uses the sigma function, while multiclassification uses the softmax function. Assuming that the number of classifications is k, the softmax function is as follows.

$$\mathrm{f}_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{9}$$

Here, f_j is the probability that the classification result is j (if one of the f_j is greater than the other f, the behavior is considered as the corresponding classification.), and the sum of the probability satisfying all the classification results is 1, i.e.:

$$\sum\nolimits_{\mathrm{k}} \mathrm{f}_k(z) = \frac{\sum_k e^{z_k}}{\sum_k e^{z_k}} = 1 \tag{10}$$

In this paper, the behaviors of insiders are classified into 6 categories. One is normal behavior, and the others are abnormal behaviors in 5 scenarios. Therefore, it belongs to the multiclassification category and uses the softmax function.

3.2. Algorithm flow of insider threat detection for adaptive optimization DBN

The insider threat detection for adaptive optimization DBN by using behavior logs can select a global optimal network structure adaptively according to the dataset. Assuming that the number of nodes in the input layer is *input*, the number of nodes in the output layer is *output*, the optimal result set is *results*, the DBN network structure is *layers*, and number of the hidden layer is c, then the algorithm process is as follows:

(1) initialization: *input* = the number of the initial feature, *output* = the classification number, *results* = [], *layers* = [], c= 1

(2) the number of the maximum node on the cth hidden layer is $i = input$

(3) judgment condition: $i >$ *output*

1) false: the global optimal network structure is *layers* and the algorithm ends

2) true: calculate number of the output nodes on the cth hidden layer *outputj*, i.e. the minimum number of nodes on the cth hidden layer, and go to step (4)

(4) the number of optional nodes on the cth hidden layer is j, and the initial value is $j = i$

(5) judgment condition: $j \quad <$ *outputj*

1) true: add j to the network structure to form a new temporary network structure *layerj*, on which we carry out deep learning by using the DBN model, add the result *resultj* to the result set on the cth hidden layer *result*, $j = j - 1$, and go to step (5) and continue the loop

2) false: select the optimal result *resOptimalj* on the cth hidden layer from the *result*; the corresponding number of nodes *resOptimal* is seen as the optimal number of nodes, and go to step (6)

(6) add the *resOptimalj* to the optimal result set *results* on the cth hidden layer

(7) add the optimal number of nodes *resOptimal* to the network structure on the cth hidden layer to form a new network structure *layers*

(8) calculate the maximum number of nodes of the next hidden layer $i = resOptimal$

(9) hidden layer plus 1, $c = c + 1$, and go to step (3)

(10) select the optimal result from the result set *results*, which includes all the hidden layers; the corresponding index is *max*, and *layers[1. . . . . max]* is the global optimized network structure

4. Experiments and result analysis

In the CERT-IT dataset, the log data of October 2010 were selected. Due to a large amount of data, 25,000 pieces of logs of 48 persons are selected as the training set and 10,000 pieces of logs of 41 persons are selected as the testing set, among which there are 1410 pieces of threat logs. After the integration and normalization of the selected datasets, the adaptive optimization DBN model is adopted to perform the feature learning on the training set and to perform the test on the testing set. Additionally, the testing results are compared and analyzed.

4.1. Dataset

The CERT-IT dataset is from the Insider Threat Center of Carnegie Mellon University. The dataset simulated threat data and large amounts of normal data for 5 scenarios implemented by malicious insiders, which involve user behavior data from multiple dimensions, such as file access (creating, deleting, modifying the file name and type, etc.), sending or receiving email, use of devices (mobile storage devices, printers, etc.), HTTP access, the system login, and other behaviors, and it also includes information such as users' positions and departments.

Because some operations during nonworking hours (e.g., login outside work hours, using USB disk after work), some operations by leaving users (e.g., the leaving users use a USB disk frequently to steal data before leaving, or send a large number of emails to create fear and panic), frequent logging into machines of other users to search for desired files and then sending them to their own mailboxes by email, and deviations in the number of copied files are all considered as abnormal behaviors, the extraction features include time, the employment status of the user, name of the host, and operation behaviors.

4.2. Result analysis

For threat detection, the model is typically evaluated from five aspects: false negative rate (FNR), false positive rate (FPR), true negative rate (TNR), true positive rate (TPR), and accuracy rate (AR). TP represents that legal behavior is correctly judged as legal behavior. TN represents that the potential threat is correctly judged as a threat behavior. FN represents that the potential threat behavior is misjudged as legal behavior. FP represents that the legal behavior is misjudged as a threat behavior. If the number of legal behaviors is represented by normalNO and the number of the threat behaviors is represented by threatNO, then the calculation formulas of FNR, FPR, TNR, TPR, and AR are as follows:

$$FNR = \frac{FN}{\text{threat}NO} \tag{11}$$

$$FPR = \frac{FP}{\text{normalNO}} \tag{12}$$

$$TNR = \frac{TN}{\text{threatNO}} \tag{13}$$

$$TPR = \frac{TP}{normalNO} \tag{14}$$

$$AR = \frac{TP + TN}{TP + FN + TN + FP} \tag{15}$$

According to the definitions of Eqs. (11) and (13), the lower the FNR, the higher the TNR, and the higher the FNR, the lower the TNR. Therefore, we only study TNR. Similarly, for Eqs. (12) and (14), we only study TPR. Therefore, this paper discusses the selections of batch size, RBM iterations, depth of hidden layer, and number of units in hidden layer by using the dichotomy method and golden section method in adaptive optimization DBN with regard to 3 aspects, i.e. TNR, TPR, and AR, and compares the dichotomy method with the golden section method of the adaptive optimization DBN network structure.

(1) Selection of batch size

Figure 3 shows that TPR is relatively stable. When batch size is 100, TNR reaches the maximum value at 95.745%, AR is also relatively high, and the adaptive optimization DBN model can achieve good performance.

(2) Selection of RBM iterations

Figure 4 shows that the model can achieve good performance when RBM is iterated once only. At this point, TNR reaches the maximum value at 95.745%.

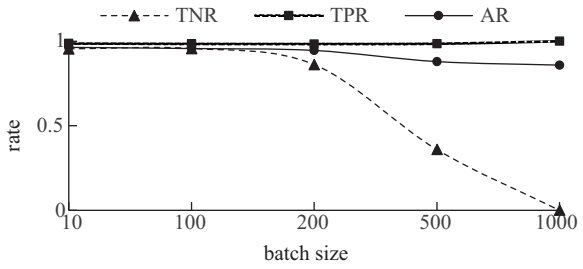

Figure 3. Selection of batch size.

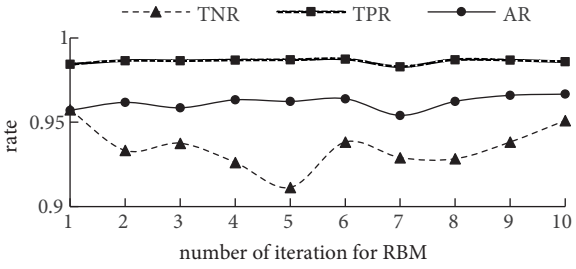

Figure 4. Selection of RBM iterations.

(3) Selection of number of units in hidden layer and hidden layer depth using the golden section method

Figure 5 shows that, when using the golden section method, in each hidden layer, there is a hidden unit number with good performance, and the hidden unit numbers are 152, 148, 103, 88, 58, 55, 40, 22, and 20, respectively. At those points, TNR is higher on their own hidden layers, which are respectively 96.596%, 97.376%, 95.745%, 97.092%, 93.83%, 92.057%, 89.929%, 97.872%, and 83.759%. Furthermore, TPR is relatively stable and AR is not low.

By analyzing the network structure of 9 hidden layers, we can see that when the hidden layer depth is 8, the maximum value of TNR is 97.872%, which is shown in Figure 6. Therefore, when the golden section method is used, the network structure [152 148 103 88 58 55 40 22] with the hidden layer depth of 8 is the optimal structure.

(4) Selection of hidden unit number and hidden layer depth using the dichotomy method

Similarly, when using the dichotomy method, in each hidden layer, there is a hidden unit number with good performance, and the hidden unit numbers are 152, 148, 103, 88, 47, 40, 38, 30, and 23, respectively. At those points, TNR is higher on their own hidden layers, which are respectively 96.596%, 97.376%, 95.745%, 97.092%, 96.454%, 96.241%, 91.418%, 89.362%, and 86.241%. Furthermore, TPR is relatively stable and AR is not low.

By analyzing the network structure of 9 hidden layers, we can see that when the hidden layer depth is 2, the maximum value of TNR is 97.376%, which is shown in Figure 7. Therefore, when the dichotomy method is used, the network structure [152 148] with the hidden layer depth of 2 is the optimal structure.

(5) Comparison of the dichotomy method with the golden section method for selecting the network structure using adaptive optimization DBN model

Figure 8 shows that the TNR of the dichotomy method is similar to that of the golden section method when the training testing is performed in the network model containing 1, 2, 3, and 4 hidden layers; the TNR of the dichotomy method is higher than that of the golden section method when the training testing is performed in the network model containing 5, 6, 7, and 9 hidden layers, while the TNR of the golden section method is higher than that of the dichotomy method when the training testing is performed in the network model containing 8 hidden layers, which is also the maximum value, reaching 97.872%. Moreover, the time performance is tested. The training time of the dichotomy method is 1.83 s, and that of the golden section method is 3.26 s. For the testing time, the former is 0.14 s and the latter is 0.28 s.

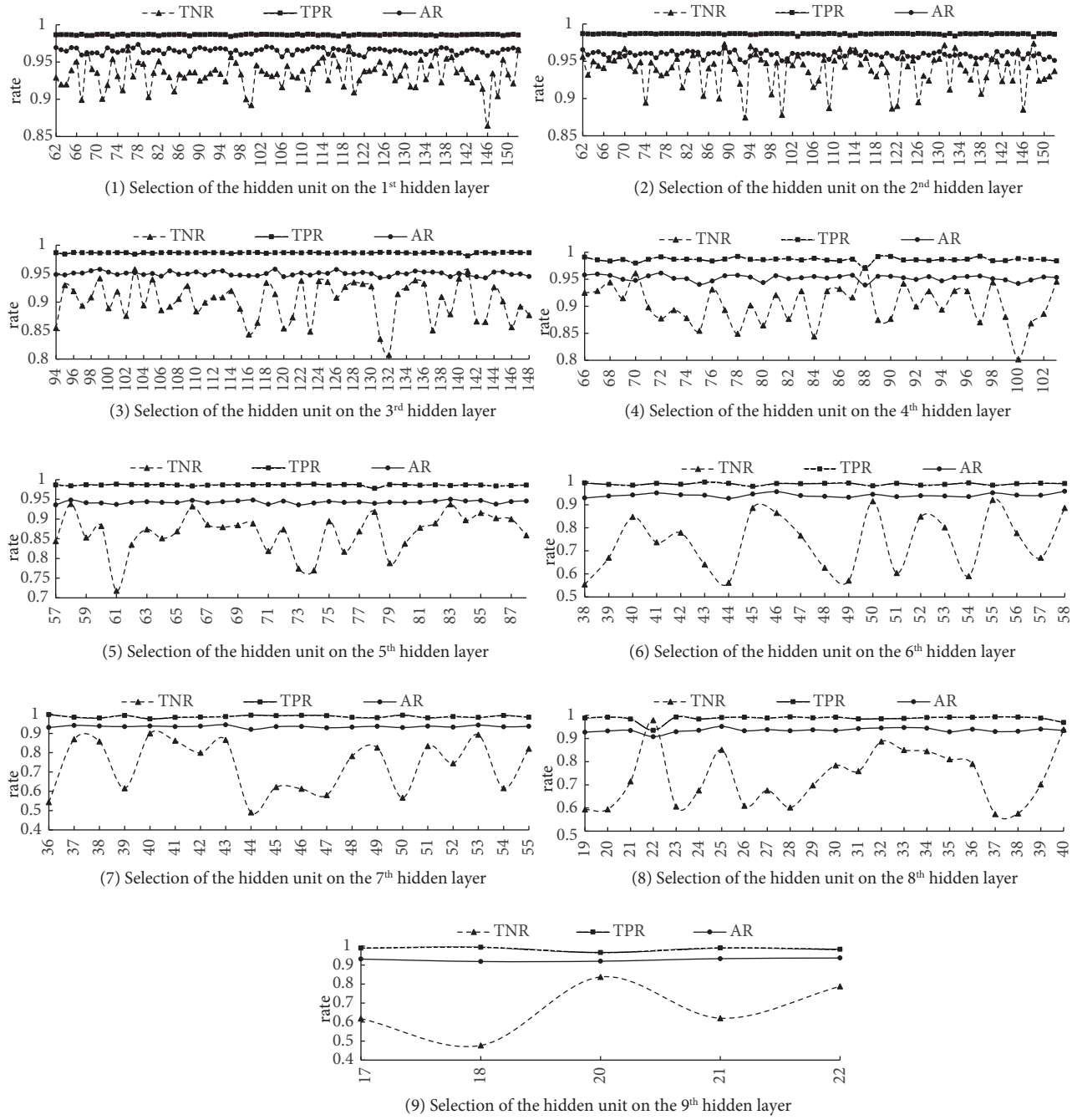

Figure 5. Selection of hidden unit number on every hidden layer using the golden section method.

5. Conclusion and future work

Most insider behaviors are normal and few have potential threats. However, the loss caused by insider threats is huge. Deep learning can extract the features of the input data from the bottom to the top, since these features can fully represent the normal behaviors and the abnormal behaviors of insiders, thereby further improving the detection rate of insider threat detection. Therefore, this paper integrates the behavior logs of insiders; normalizes the behavior logs by using the 1/N code discretization method; carries out the experiments on the

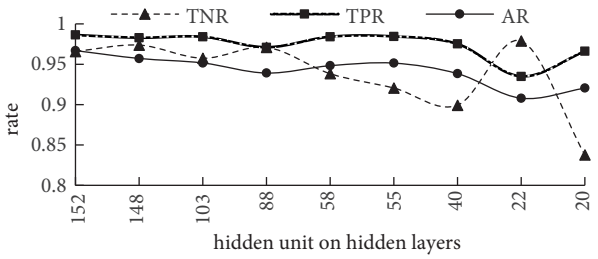

Figure 6. Selection of hidden layer depth using the golden section method.

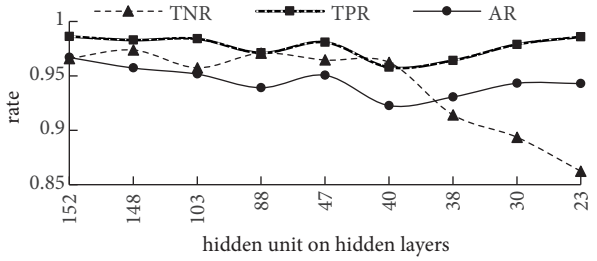

Figure 7. Selection of hidden layer depth using the dichotomy method.

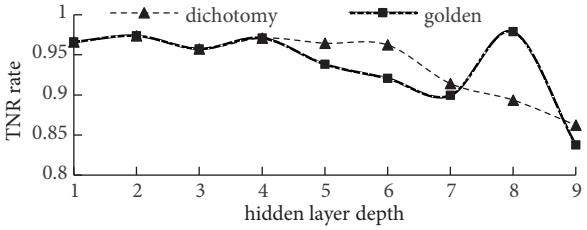

Figure 8. Comparison of methods.

CERT-IT dataset; discusses the selection of batch size, RBM iterations, number of units in the hidden layer and depth of the hidden layer by using the dichotomy method and golden section method with regard to 3 aspects (true negative rate, true positive rate, and accuracy rate); and compares the dichotomy method with the golden section method for selecting the network structure using the adaptive optimization DBN model. Experiments show that batch size of 100 is the optimal batch size; the model can achieve good performance when RBM is iterated only once; when the dichotomy method is used and the depth of hidden layer is 2, i.e. the network structure is [152 148], the TNR can achieve the maximum value at 97.376%; and when the golden section method is used and the depth of hidden layer is 8, i.e. the network structure is [152 148 103 88 58 55 40 22], the TNR can achieve the maximum value at 97.872%. In particular, in the adaptive optimization DBN algorithm, the golden section method is better than the dichotomy method, which can increase the TNR by 0.50% in the best case. The presented results illustrate that the detection rate is a little higher than those found in the literature [3], whose highest detection rate is 97.78%. Moreover, the method presented in this manuscript also has good adaptation and versatility. Therefore, the adaptive optimization DBN model can select an optimization network structure adaptively to improve the detection rate of insider threats. Moreover, the optimization method is very applicable, which can also be applied to other areas or other deep learning models.

The next step is to further improve the classification method, increase the detection rate, and train massive data in combination with big data technology to realize a better deep learning network model.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61201220, 61309018) and the National Basic Research Program of China (No. 2012CB315901).

References

[1] Coden A, Lin WS, Houck K, Tanenblatt M, Boston J, MacNaught JE, Soroker D, Weisz JD, Pan S, Lai JH et al. Uncovering insider threats from the digital footprints of individuals. IBM J Res Dev 2016; 60: 1-11.

[2] Greitzer FL, Moore AP, Cappelli DM, Andrews DH, Carroll LA, Hull TD. Combating the insider cyber threat. IEEE Secur Priv 2008; 6: 61-64.

[3] Azaria A, Richardson A, Kraus S, Subrahmanian VS. Behavioral analysis of insider threat: A survey and bootstrapped prediction in imbalanced data. IEEE T Comput Soc Syst 2015; 1: 135-155.

[4] Cybenko G. Deep learning of behaviors for security. In: ACM International Workshop on International Workshop on Security and Privacy Analytics; 2–4 March 2015. New York, NY, USA: ACM. pp. 1-1.

[5] Jaccard N, Rogers TW, Morton EJ, Griffin LD. Automated detection of smuggled high-risk security threats using deep learning. In: 7th International Conference on Imaging for Crime Detection and Prevention; 23–25 November 2016; Madrid, Spain. pp. 4-4.

[6] Ho SM, Warkentin M. Leader's dilemma game: an experimental design for cyber insider threat research. Inform Syst Front 2017; 19: 377-396.

[7] Stakhov AP. The generalized principle of the golden section and its applications in mathematics, science, and engineering. Chaos Soliton Fract 2005; 26: 1157-1182.

[8] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006; 313: 504.

[9] Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: European Conference on Computer Vision; 6–12 September 2014; Zurich, Switzerland. pp. 818-833.

[10] Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Comput 2006; 18: 1527-1554.

[11] Bengio Y. Learning Deep Architectures for AI. Foundations and Trends in Machine Learning. Delft, the Netherlands: Now Publishers, 2009.

[12] Cao LL, Huang WB, Sun FC. Building feature space of extreme learning machine with sparse denoising stacked-autoencoder. Neurocomputing 2016; 174: 60-71.

[13] Hinton GE. A practical guide to training restricted Boltzmann machines. In: Montavon G, editor. Neural Networks: Tricks of the Trade 2012. 2nd ed. Berlin, Germany: Springer. pp. 599-619.

[14] Salakhutdinov R, Hiton G. An efficient learning procedure for deep boltzman machines. Neural Comput 2012; 24: 1967-2006.

[15] Hiton GE. Training products of experts by minimizing contrastive divergence. Neural Comput 2002; 14: 1771-1800.