Identification and Processing of Network Abnormal Events Based on Network Intrusion Detection Algorithm

Yunbin He
(Corresponding author: Yunbin He)

College of Physics and Information Engineering, Zhaotong University
Room 101, Unit 1, Building 3, Zuanshi Renjia, Zhuquan Road, Zhaoyang district, Zhaotong, Yunnan, China
(Email: hybinztu@yeah.net)

(Received May 31, 2018; revised and accepted Sept. 7, 2018)

Abstract

With the popularity of the Internet, people's lives are becoming more and more convenient, but the network security problems are also becoming increasingly serious. In order to better prevent internal or external malicious attacks and protect the network security of users, this study chose deep neural network (DNN) learning algorithm and convolutional neural network (CNN) learning algorithm as network intrusion detection algorithms and tested two algorithms under different parameters and activation functions with KDD99 data set on the MATLAB simulation platform. Moreover, the performance of the algorithms was compared with those of other clinic algorithms and deep learning algorithms. The results suggested that the recognition performance of DNN and CNN learning algorithms was different under different network parameters and activation functions. When ReLU function was used as the activation function, the recognition performance was the best. The network parameters of DNN and CNN were 122-250-520-250-5 and was 10(18)-14(22)-16 (18), respectively. The recognition performance of DNN and CNN learning algorithms were better than those of the classical algorithms, self-organizing map (SOM) and support vector machine (SVM) algorithms, but was worse than that of dynamic Bayesian network (DBN) algorithm. DNN was superior to DBN in the aspect of false alarm rate; overall, DNN algorithm was superior to DBM algorithm.

Keywords: Convolutional Neural Network; Deep Neural Network; Detection Algorithm; Network Security

1 Introduction

With the development of the Internet and the popularity of computers, information sharing and communication between people are becoming more frequent. The flow of data in the network is also growing, and a large part of the growing data are individual information and confidential information of enterprises which need to be kept secret, but these data are very easy to induce malicious network attacks because of their business values [4].

In the Internet age, software used for network attacks is easy to obtain, making the unlawfully malicious attacks easily made without professional knowledge. These malicious network attacks have seriously affected the use of network and computers. Many studies have studied this problem. Hong *et al.* [1] put forward a multistage distributed vulnerability detection, measurement and game selection mechanism based on attack graph analysis model and reconfigurable virtual network, and built the monitor and control plane on distributed programmable virtual switches using OpenFLUE Application Program Interface (API) to significantly improve attack detection ability and mitigate consequences of attacks. The system and security assessment suggested that the proposed solution was effective and efficient.

To improve the security of in-vehicle network, Kang *et al.* [2] proposed a deep neural network (DNN) based intrusion detection system and the technology to initialize parameters using the non-supervised pre-training of deep belief network to improve detection preciseness. The experimental results demonstrated that the technology could produce real-time response to attacks on the bus of controller area network and improve the detection rate significantly. Hodo [3] proposed dealing with malicious attacks on the Internet with artificial neural network (ANN) and focused on the classification of normal mode and threat mode on the network. The experimental results demonstrated that the method had a preciseness of 99.4% and could detect all kinds of distributed denial of service (DDoS) attacks successfully.

In this study, DNN learning algorithm and convolutional neural network (CNN) learning algorithm were selected as network intrusion detection algorithms, and the two algorithms are tested with KDD99 data set under different parameters and activation functions on the MATLAB simulation platform. Finally, it was compared with the performance of other classical algorithms and depth learning algorithms.

2 Network Intrusion Detection

2.1 Intrusion Detection Model

Figure 1 shows a simple model of network intrusion detection [5]. It could be seen from Figure 1 that the intrusion detection model had four layers, data input layer, neural network layer, data classification layer and classification result layer. The neural network layer is used for feature extraction of data and combined with the classification layer to form a deep learning network. In this study, DNN learning algorithm and CNN learning algorithm were taken as network intrusion detection algorithms.

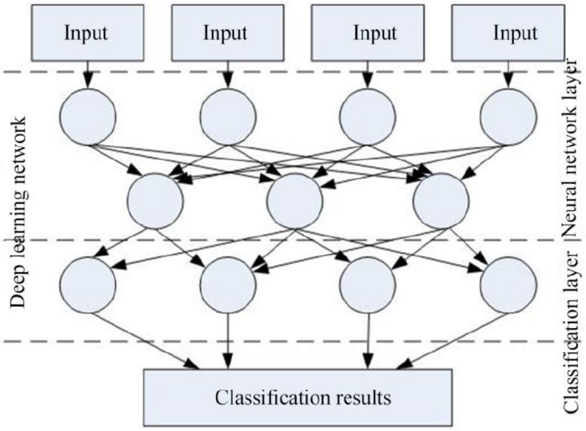

Figure 1: The structure of the intrusion detection model

2.2 DNN Model Algorithm

DNN is essentially is a multilayer perceptron containing multiple hidden layers in forward neural network structure, and it includes three layers, input layer, hidden layer and output layer. If the input feature is $g^0 = N$, then the activate value of nodes on the hidden layer of DNN [6] is expressed as:

$$\begin{aligned} b^m &= W^m g^{m-1} + a^m \quad (1 \leq m \leq M+1) \\ g^m &= f(b^m) \text{ with } g_j^m = \frac{1}{1+e^{-b_j^m}} \quad (1 \leq m \leq M) \end{aligned}$$

where N stands for the number of hidden layers of DNN, W^m and a^m are the weight and offset vector of the m-th hidden layer respectively, and $f(\cdot)$ is the non-linear activation function sigmoid of nodes on the hidden layer.

The output layer of DNN often uses softmax function [8] to model the posterior probability distribution of input features, and its expression is:

$$y_s = g_s^{M+1} = Pr(s|N) = softmax_s(b^{M+1})$$

where y_s is the s-th element in output vector y.

The result is obtained after extraction feature is input, and such a process is known as forward propagation process. Finally, the result of the output layer needs to be compared with the guidance signal, and the corresponding optimization algorithm is needed in the comparison. At present, the common optimization algorithm is the stochastic gradient descent based error back propagation algorithm [9].

2.3 CNN Model Algorithm

CNN is an algorithm model inspired by receptive field mechanism in biology. It is essentially a mathematical model with supervised learning module [13]. In CNN, multiple convolutional layers alternate to extract features of the input layer and then performed integration and transformation on the extracted features through the fully connected layer, *i.e.*, the largest pooling layer. CNN can effectively obtain generalized features from a large amount of learning data. Convolutional layer is the core part of the whole network, and its output is called feature map; the convolution is like a linear weighting operation, and its expression [7] is:

$$R(i,j) = (O*H)(i,j) = \sum_c \sum_d O(i+c, j+d)H(c,d).$$

The expression for the generation of feature map [14] is

$$\alpha_j^m = f(\gamma^m) = f(\sum_{i \in N_i} \alpha_i^m * H_j^m + \beta_j^m)$$

where α_j^m is the output feature map of the j-th convolution kernel on the m-th layer, N_j is the set of output feature map of the m-1-th layer, H_j^m is the j-th convolution kernel of the m-th layer, β_j^m is the bias term of the feature map of the corresponding convolution kernel, and $*$ is convolution operation.

Pooling layer, also called down sampling layer, is mainly used for compressing feature map obtained from the convolutional layer. Max pooling and even pooling are common in practical application.

3 Simulation Experiment

3.1 Data Preparation

KDD99 data set was used in the experiment [10]. Each data in the data set was 42-dimensional. The first 41 dimensions were feature attributes of data, and the last one was a decision attribute which indicated whether the data was abnormal. The data set included data of the known network intrusion categories and normal data, which could simulate real network environment. 20% of the data set were taken as training samples, and the remaining 80% were taken as test samples.

Table 1: The network parameters and activation functions of DNN algorithm

No. of model	Network parameter	Activation functions of the hidden layer and input layer	Activation function of the output layer
DNN1	122-90-40-10-5	relu	softmax
DNN2	122-90-40-10-5	tanh	
DNN3	122-90-40-10-5	sigmoid	
DNN4	122-250-520-250-5	relu	
DNN5	122-250-520-250-5	tanh	
DNN6	122-250-520-250-5	sigmoid	

3.2 Data Preprocessing

Among the 42 dimensions of features of data in KDD99 set, 38-dimensional features were numbers, and 3-dimensional features were characters which could not be directly identified by CNN. Therefore, character features should be firstly converted to numerical features. The 41 dimensions of features became 122 dimensions of numerical features. Then the numerical features were normalized, and its expression [15] is:

$$x' = \frac{x - N_{min}}{N_{max} - N_{min}}$$

where x is the numerical value which needs to be normalized, N_{min} is the minimum value in some dimension, and N_{max} is the maximum value in some dimension.

3.3 Evaluation Standard

Usually the performance of intrusion detection algorithm is represented by three data, accuracy rate B_C, false alarm rate E_A and missing report rate. Intrusion detection algorithms with higher accuracy rate and lower false alarm and missing report rates were better. The expressions of them [11] were:

$$\begin{aligned} B_C &= \frac{C_P + C_N}{C_P + C_N + M_P + M_N} \\ E_A &= \frac{M_N}{C_N + M_N} \\ N_A &= \frac{M_P}{C_P + M_P} \end{aligned}$$

where C_P stands for attack data which are accurately classified, C_N stands for normal data which are accurately classified, M_N stands for normal data which are wrongly classified, and M_P stands for attack data which are wrongly classified.

3.4 Experimental Environment

Algorithm model was edited using Matlab. The experiment was carried out on a server which was installed with Windows 7, i7 processor and 16 G memory in a laboratory.

3.5 Setting of Algorithm

1) DNN Algorithm Model
DNN included one input layer, one output layer and three hidden layers. The network parameters were represented by the corresponding dimensions of data in each layer. Cross entropy was used as the loss function in the training process. The random gradient descent method was selected to avoid the local optimal solution. In addition to the output layer which applied softmax as the activate function, the other layers applied sigmoid, relu and tanh as activation functions [12]. The performance test was performed using the testing set after training. The specific choices of network parameters and activation functions of DNN algorithm model are shown in Table 1.

2) CNN Algorithm Model
CNN included one input layer, one output layer, hidden layers including three convolution layers and three down sampling layers. In the convolution layer, the data features obtained from the upper layer was processed by activation function and convolution kernel and then output to the down sampling layer, the next convolution layer and output layer. The parameter of the convolution layer was expressed as $x(y)$, where x stands for the number of convolution kernel and y stands for the length of convolution kernel. Except the output layer which applied softmax, the other layers took sigmoid, relu and tanh as activation functions. The performance was tested using testing set after training. The specific choices of network parameters and activation functions of CNN algorithm are shown in Table 2.

3.6 Experimental Results

3.6.1 The Performance of DNN Algorithm

The recognition performance of the DNN based intrusion detection algorithm under different network parameters and activation functions is shown in Table 3. It was known from Tables 1 and 3 where the control variable method was used. The activation functions of DNN1,

Table 2: The network parameters and activation function of CNN algorithm

No. of model	Convolution Layer 1	Convolution Layer 2	Convolution Layer 3	Activation function of convolution layer	Activation function of output layer
CNN1	2(4)	4(5)	8(6)	relu	softmax
CNN2	2(4)	4(5)	8(6)	tanh	
CNN3	2(4)	4(5)	8(6)	sigmoid	
CNN4	10(18)	14(22)	16(18)	relu	
CNN5	10(18)	14(22)	16(18)	tanh	
CNN6	10(18)	14(22)	16(18)	sigmoid	

Table 3: The recognition performance of DNN algorithm under different network parameters and activation functions

No. of model	Accuracy BC/%	False alarm rate EA/%	Missing report rateNA/%
DNN1	92.32	1.90	9.11
DNN2	92.38	1.61	9.12
DNN3	91.99	1.51	9.71
DNN4	92.88	0.45	9.01
DNN5	92.45	1.45	9.21
DNN6	91.89	1.66	9.74

DNN2 and DNN3 were different from the activation functions of DNN4, DNN5, and DNN6. The network parameters were different between DNN1 and DNN4, DNN2 and DNN5, and DNN3 and DNN6. The final experimental result demonstrated that DNN4 network parameter, 122-250-520-250-5, and activation function, relu, had the strongest recognition performance, 92.88% accuracy, 0.45% false alarm rate and 9.01% missing report rate. The comparison of the recognition performance of different DNNs suggested that network parameter had little influence on the recognition rate, but activation function had an influence on the recognition rate, and the efficacy of relu and tanh was better than that of sigmoid.

As the proportion of attack data was far larger than that of normal data in the data set and the situation is opposite in the reality, the actual missing report rate should be significantly lower than the false alarm rate rather than the false report rate was lower than the missing report rate in the experimental result.

3.6.2 The Performance of CNN Algorithm

The recognition performance of the CNN based intrusion detection algorithm under different network parameters and activation functions is shown in Table 3. It was known from Tables 2 and 4 that the control variable method was used. It was found that the recognition accuracy of the CNN based intrusion detection algorithm was about 92%, nearly not affected by the number and length of convolution kernel; CNN4 had the highest recognition accuracy, 92.47%; activation function had an obvious influence on the recognition performance of the algorithms; relu and tanh had favorable effects; the over fitting of sigmoid led to the failure of experiment because it determined all data as attack data. CNN4 had the best recognition performance overall though not all indexes of CNN4 were the best. Similar to CNN, as the proportion of attack data was far larger than that of normal data in the data set and the situation is opposite in the reality, the actual missing report rate should be significantly lower than the false alarm rate rather than the false report rate was lower than the missing report rate in the experimental result.

3.6.3 Comparison between Different Algorithms

To verify the recognition abilities of the two algorithms, the recognition performances of DNN4 and CNN4 which had the best recognition performance was compared with those of self-organizing map (SOM) algorithm and support vector machine (SVM) algorithm which were mentioned in literature, as shown in Table 5.

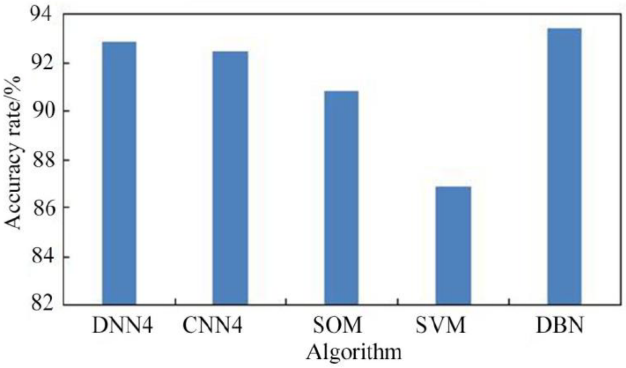

Figure 2: Comparison of the recognition accuracy between different algorithms

Table 4: The recognition performance of CNN algorithm under different network parameters and activation functions

No. of model	Accuracy BC/%	False alarm rate EA/%	Missing report rate NA/%
CNN1	91.99	1.56	9.52
CNN2	92.39	1.57	9.49
CNN3	80.43	100	0
CNN4	92.47	1.57	8.89
CNN5	92.14	1.58	9.31
CNN6	80.54	100	0

Table 5: The recognition performance of different algorithms

No. of model	Accuracy BC/%	False alarm rate EA/%	Missing report rate NA/%
DNN4	92.88	0.45	9.01
CNN4	92.47	1.57	8.89
SOM	90.85	1.14	10.45
SVM	86.92	1.92	13.45
DBN	93.39	0.75	7.65

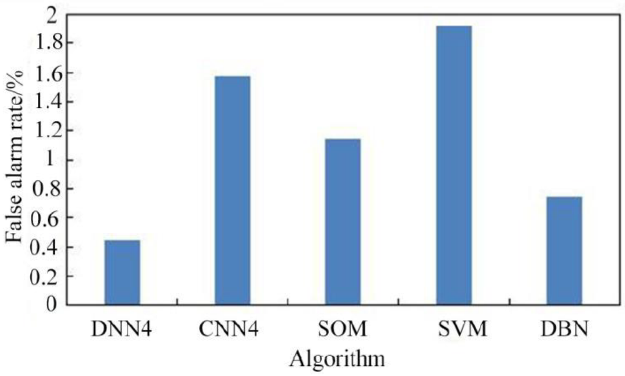

Figure 3: Comparison of the false alarm rate between different algorithms

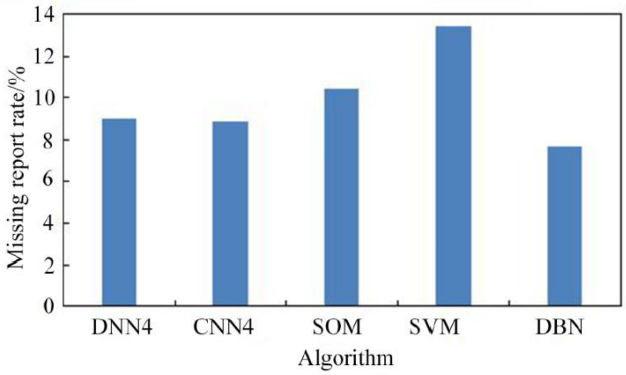

Figure 4: Comparison of the missing report rate between different algorithms

In Table 5, SOM and SVM algorithms are classical algorithms, and DBN algorithm is a deep learning model algorithm. Figure 2 exhibits that the recognition accuracy of the DNN4 and CNN4 algorithms was higher than those of the SOM and SVM algorithms; they are 2.05%, 5.96%, 1.62% and 5.55% higher, respectively; the recognition accuracy of the DBN was 0.51% and 0.92% higher than those of the DNN4 and CNN4 algorithms, respectively.

Figure 3 shows that the false alarm rate of the DNN4 algorithm was far lower than those of the SOM, SVM and DBN algorithms; they are 0.69%, 1.47% and 0.3%, respectively; the false alarm rate of the CNN4 algorithm was 0.43% and 0.82% higher than those of SOM and DBN algorithms, respectively, but 0.35% lower than that of the SVM algorithm.

Figure 4 shows that the missing report rates of the DNN4 and CNN4 algorithms (9.01% and 8.89%) were lower than those of the SOM and SVM algorithms (10.45% and 13.45%), but higher than that of the DBN algorithm (7.65%).

To sum up, the DNN algorithm and CNN algorithm were better than classical algorithms, SOM and SVM, in recognizing network abnormal events in the aspects of accuracy, false alarm rate and missing report rate, especially in the accuracy; though the comprehensive performance of the DNN4 algorithm was slightly poorer compared with the DBN algorithm, it was superior to the DBN algorithm in the false alarm rate.

4 Conclusion

DNN and CNN algorithms were used in this study as the network intrusion detection algorithms, and the recognition performance of the algorithms under different network parameters and activation functions was tested on the MATLAB simulation platform. Finally, the algorithm with better performance was selected and compared with SOM and SVM algorithms and DBN algorithm.

When the network parameter and activation function of the DNN algorithm were 122-250-520-250-5 and relu, respectively, the recognition performance was the best; the accuracy, false alarm rate and missing report rate at that time were 92.88%, 0.45% and 9.01%, respectively. Network parameter had little influence on the performance, while activation function had a large influence.

When the parameter of convolution kernel and activation function of the CNN algorithm was 10(18)-14(22)-16(18) and relu, respectively, the recognition performance was the best; the accuracy, false alarm rate and missing report rate at that time were 92.47%, 1.57% and 8.89%, respectively. Moreover the number and length of convolution kernel had little influence on the performance, while activation function had an obvious influence. The over fitting of sigmoid led to the failure of the experiment.

In recognizing network abnormal events, the DNN and CNN algorithms are better than the classical algorithms, SOM and SVM algorithms, especially in the accuracy, the false alarm rate and the failure rate; however, compared to the DBN algorithm, the overall recognition performance of the DNN and CNN algorithms was poorer, but the false alarm rate of the DNN algorithm was superior to the DBN algorithm.

References

[1] J. B. Hong, C. J. Chung, D. Huang, *et al.*, "Scalable network intrusion detection and countermeasure selection in virtual network systems," in *International Conference on Algorithms and Architectures for Parallel Processing*, pp. 582–592, 2015.

[2] M. J. Kang, J. W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *Plos One*, vol. 11, no. 6, 2016.

[3] E. Hodo, X. Bellekens, A. Hamilton, *et al.*, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in *IEEE International Symposium on Networks, Computers and Communications*, pp. 6865–6867, 2016.

[4] R. Singh, H. Kumar, R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8609–8624, 2015.

[5] S. Choudhury, A. Bhowal, "Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection," in *IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials*, pp. 89–95, 2015.

[6] A. Schwarz, C. Huemmer, R. Maas, *et al.*, "Spatial diffuseness features for DNN-based speech recognition in noisy and reverberant environments," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4380–4384, 2015.

[7] T. Yoshioka, S. Karita, T. Nakatani, "Far-field speech recognition using CNN-DNN-HMM with convolution in time," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4360–4364, 2015.

[8] M. Behnam, H. Pourghassem, "Power complexity feature-based seizure prediction using DNN and firefly-BPNN optimization algorithm," in *22nd Iranian Conference on Biomedical Engineering (ICBME'15)*, pp. 10–15, 2015.

[9] G. Li, S. K. S. Hari, M. Sullivan, *et al.*, "Understanding error propagation in deep learning neural network (DNN) accelerators and applications," in *International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–12, 2017.

[10] T. Yoshioka, S. Karita, T. Nakatani, "Far-field speech recognition using CNN-DNN-HMM with convolution in time," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4360–4364, 2015.

[11] S. Rastegari, P. Hingston, C. P. Lam, "Evolving statistical rulesets for network intrusion detection," *Applied Soft Computing*, vol. 33(C), pp. 348–359, 2015.

[12] A. J. Malik, W. Shahzad, F. A. Khan, "Network intrusion detection using hybrid binary PSO and random forests algorithm," *Security & Communication Networks*, vol. 8, no. 16, pp. 2646–2660, 2015.

[13] T. Szabo, P. Barsi, P. Szolgay, "Application of analogic CNN algorithms in telemedical neuroradiology," in *IEEE International Workshop on Cellular Neural Networks and Their Applications*, pp. 579–586, 2016.

[14] X. Ren, K. Chen, J. Sun, "A CNN based scene chinese text recognition algorithm with synthetic data engine," *CoRR*, vol. abs/1604.01891, 2016.

[15] N. Khamphakdee, N. Benjamas, S. Saiyod, "Improving intrusion detection system based on snort rules for network probe attacks detection with association rules technique of data mining," *Journal of ICT Research & Applications*, vol. 8, no. 3, pp. 234–250, 2015.

Biography

Yunbin He is a member of Communist Party of China and the associate professor of Zhaotong University, Yunnan, China. She is engaging in teaching and scientific research of computer. She has published more than 20 academic papers on journals which are at the provincial level or above such as Application Research of Computers, Electronic Technology & Software Engineering, Computer Programming Skills & Maintenance and Journal of Zhaotong University and participated in the writing of one textbook and one teaching auxiliary book.