IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Network intrusion detection algorithm based on deep neural network

ISSN 1751-8709
Received on 25th December 2017
Revised 27th May 2018
Accepted on 27th June 2018
E-First on 5th September 2018
doi: 10.1049/iet-ifs.2018.5258
www.ietdl.org

Yang Jia[1] ✉, *Meng Wang*[1], *Yagang Wang*[1]

[1]*School of Computer Science, Xi'an University of Posts and Telecommunications, Chang'an West St. Chang'an District, Xi'an, People's Republic of China*

✉ *E-mail: jiayang@xupt.edu.cn*

Abstract: With the rapid development of network technology, active defending of the network intrusion is more important than before. In order to improve the intelligence and accuracy of network intrusion detection and reduce false alarms, a new deep neural network (NDNN) model based intrusion detection method is designed. A NDNN with four hidden layers is modelled to capture and classify the intrusion features of the KDD99 and NSL-KDD training data. Experiments on KDD99 and NSL-KDD dataset shows that the NDNN-based method improves the performance of the intrusion detection system (IDS) and the accuracy rate can be obtained as high as 99.9%, which is higher when compared with other dozens of intrusion detection methods. This NDNN model can be applied in IDS to make the system more secure.

1 Introduction

Intrusion detection is to collect and analyse information about the key nodes in a network to find if there are violated security behaviours or signs of being attacked. Intrusion detection system (IDS) is an independent system providing local network services to ensure the security of the network system. With the rapid development of network technology, active defending of the network intrusion is more important than before. So, a more intelligent and accurate IDS is needed to address these challenges.

There are a lot of researches applying machine learning methods in IDS to detect network intrusions in both academia and industry [1–9]. Traditional IDS first extracts and analyses features of intrusion and attack mode. Then an intrusion pattern library and a series of discrimination rules are built to detect intrusion [10]. Many different approaches have been tried in researches. Sujendran and Arunachalam [11] propose an automatic fuzzy rule generation combined with a Wiener filter to identify attacks. Yaseen *et al.* [2] propose a multi-level hybrid intrusion detection model that uses support vector machine (SVM) and extreme learning machine to attacks. A modified *K*-means algorithm is also used to build new small training datasets representing the entire original training dataset to reduce the training time and improve the performance of IDS. Kaur *et al.* [3] propose hybridisation of *K*-means and firefly algorithm for anomaly detection. The algorithm uses clustering to build the training model and uses classification to evaluate the test set. Their experiment results show that *K*-means + firefly and *K*-means + bat outperform other methods by a huge margin. Tahir *et al.* [4] study an attack system using a hybrid method combining *K*-means with SVM. In [7], intrusion was detected using the deep confidence network, and KDD Cup 99 data was used to test the algorithm. Compared with the existing SVM model and the ANN model, the precision increased more than 6%. Lin *et al.* [8] propose a cluster centre and the nearest neighbour (CANN) approach and experimental results based on the KDD99 show that the CANN classifier not only performs better than or similar to *k*-NN and SVM trained method. In [5], a tree-weighted naive Bayesian algorithm is proposed to reduce the dimension of categorical data and improve the classification accuracy of the algorithm. In [6], an IDS based on information gain criterion is used to select features from network traffic records and a new version of support vector domain description is used to classify the extracted features and to detect new intrusions. It can achieve good performance in intrusion detection. In [12], injection types of attacks in wireless networks are detected by fusing multi-metrics using the Dempster–Shafer belief theory. The automatic and self-adaptive process of basic probability assignment is considered when combining beliefs. They do not require any prior training or calibration. They said multi-layer techniques perform more efficiently than other conventional methods.

All of these methods, *K*-means [3], SVM [2], *k*-NN [8], clustering [13] and regression [14] are traditional machine learning methods, mainly rule-based methods. Compared with traditional classification methods, such as naive Bayesian, and random forest, the deep neural network (DNN) obtains a higher accuracy rate and detection rate with a low false positive rate [15]. Due to the deep network structure, more abstract features of intrusions can be extracted [16] and great detection potential is predictable.

2 Deep neural networks

Recently, deep learning has been used for intrusion detection. Continuous layers in a hierarchical way are used to do feature learning and pattern classification. Its excellent feature descriptive ability has attracted people's attention rapidly. People also use deep learning in intrusion detection. Some related methods have been concluded in Table 1. Fiore *et al.* [17] suggest the adoption of machine learning techniques to implement semisupervised anomaly detection systems where the classifier is trained with 'normal' data only, so that knowledge about anomalous behaviours can be constructed and evolve in a dynamic way. They use the discriminative restricted Boltzmann machine (RBM) to combine the expressive power of generative models with excellent classification capabilities to infer part of its knowledge from incomplete training data. The accuracies are around 94%. In [18], MIXMAD is proposed for MIXed data Multilevel Anomaly Detection. They construct an ensemble of deep belief nets (DBNs) with varying depths. The results demonstrate that MIXMAD is superior to popular unsupervised detection methods for both homogeneous and mixed data. Alrawashdeh *et al.* [19] also use RBM, together with DBNs, and they achieve a detection rate of 97.9% on the total 10% KDDCUP'99 test dataset. In [20], an intrusion detection method using DBNs and probabilistic neural network (PNN) is proposed. The method includes dimension reduction with DBN, particle swarm optimisation and PNN based classification. Experiment result shows that the method performs better than the traditional PNN, PCA-PNN and unoptimised DBN-PNN. Javaid *et al.* [16] use self-taught learning (STL) on NSL-

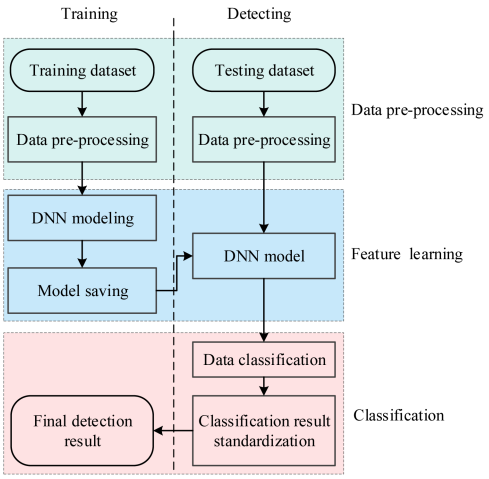

Fig. 1 *Proposed IDS structure*

KDD dataset for network intrusion and achieve a classification accuracy rate higher than 98% for all types of classification. In [21], an accelerated DNN architecture is developed to identify abnormalities in network data. NSL-KDD dataset is used to compute the training time and analyse the effectiveness of the detection mechanism. Roy *et al.* [22] check the potential capability of DNN as a classifier for different types of intrusion attacks. Their comparative study shows that DNN works better than SVM in intrusion detection. Yin *et al.* [15] propose an approach using recurrent neural networks (RNN-IDS). They compare it with those of J48, an artificial neural network, random forest, SVM and other machine learning methods proposed by previous researchers on the benchmark dataset. The experimental results show that RNN-IDS is suitable for modelling a classification model with high accuracy and that its performance is superior to that of traditional machine learning classification methods.

With the development of technology and size of internet, more and more risks and possibilities appeared on internet intrusions. To improve the accuracy of intrusion detection further, a new DNN (NDNN) model is built and tested in this paper. The IDS can be divided into three modules: data pre-processing, feature learning module and intrusion type classification module. The IDS model is presented in Fig. 1. The left side shows the training process and the right side shows the detection process. The trained DNN model is saved and used in detection. Details of the three modules will be explained below.

3 Data pre-processing

In this paper, KDD99 and NSL-KDD datasets are used as our training and testing datasets. Details about the KDD99 and NSL-KDD datasets can be found in [2]. There are four types of intrusion in the dataset [23], as shown in Table 2.

One record of the input intrusion is a feature vector with 41 elements. Description of the feature vector is shown in Table 3. Details can also be found in http://www.kdd.org/kdd-cup/view/kdd-cup-1999/Task.

An example of the original intrusion data record is

$$\begin{aligned} x = \{&0, icmp, ecr\_i, SF, 1032,\\ &0,0,0,0,0,\\ &0,0,0,0,0,\\ &0,0,0,0,0,\\ &0,0,511,511,0.00,\\ &0.00,0.00,0.00,1.00,0.00,\\ &0.00,255,255,1.00,0.00,\\ &1.00,0.00,0.00,0.00,0.00,\\ &0.00, smurf\} \end{aligned} \tag{1}$$

Table 1 Related intrusion detection algorithms based on deep neural networks

References	Methods	Performance
Fiore *et al.* [17]	RBM	accuracy: around 94%
Do *et al.* [18]	an ensemble of DBNs	detection *F*-score on mixed data is around 72%
Khaled *et al.* [19]	RBM together with DBNs	detection rate is 97.9%
Zhao *et al.* [20]	DBNs with PNN	detection accuracy is about 99%. Detection rate is about 90%
Javaid *et al.* [16]	STL	accuracy rate is more than 98%, a little lower than 99%. *F*-measure can achieve 98.84%
Potluri and Diedrich [21]	accelerated DNN	highest detection accuracy is 97.7%
Roy *et al.* [22]	DNN	better than SVM in intrusion detection
Yin *et al.* [15]	RNN-IDS	superior to traditional machine learning classification methods

Table 2 Description of five labels

Intrusion type	Description	Label
Probe	scanning and detection	1
Dos	denial of service attack	2
U2R	illegal access to local super users	3
R2L	unauthorised remote access	4
Normal	normal record	5

There are 42 attributes in x, 41 features and a label. *Smurf* is the label of the record x.

First step of data processing is data filtering (In real captured data, lots of intrusion records are same, so de-duplication would be used to reduce the amount of data entering the IDS to eliminate the information redundancy. KDD99 dataset has already been de-duplicated and filtering is not necessary in this paper.). Some of the features are numerical and others are symbolic. Then standardisation is used to transform all data captured from different IDS sources into numerical format to simplify the data processing. Symbolic features are converted to numerical labels. Rules are as follows:

(i) Protocal type: 1 icmp; 2 tcp; 3 udp; 4 others.
(ii) Service: 1 domain-u; 2 ecr_i; 3 eco_i; 4 finger; 5 ftp_data; 6 ftp; 7 http; 8 hostnames; 9 imap; 10 login; 11 mtp; 12 netstat; 13 other; 14 private; 15 smtp; 16 systat; 17 telnet; 18 time; 19 uucp; 20 others.
(iii) Flag: 1 REJ; 2 RSTO; 3 RSTR; 4 SO; 5 S3; 6 SF; 7 SH; 8 OTHERS.

Features including land, logged_in, is_host_login, is_guest_login are symbols with value 0 or 1. Different features have different metrics and the range is wide. To avoid the disappearance of small numerical attributes and reduce the amount of calculation iterations, numerical data is normalised to [0,1] with a linear function given in the following equation:

$$y = (x - x_{\min}/(x_{\max} - x_{\min}) \tag{2}$$

$x_{\min}$ and $x_{\max}$ are the minimum and maximum values of all the elements. One-hot encoding is used to present the 42th attribute of x. It is a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0) [24]. The one-hot code of 2 is 01000, as shown in (3). The advantage of using this code is all of the five codes have the same weight and it is unbiased for the classifier.

For instance, the intrusion data x is a 41-dimensional feature vector. After standardisation and normalisation, x' is

$$\begin{aligned} x' = \{&0.0, 0.0, 0.052, 0.714, 1.488 \times 10^{-6}, \\ &0.0, 0.0, 0.0, 0.0, 0.0, \\ &0.0, 0.0, 0.0, 0.0, 0.0, \\ &0.0, 0.0, 0.0, 0.0, 0.0, \\ &0.0, 0.0, 1.0, 1.0, 0.0, \\ &0.0, 0.0, 0.0, 1.0, 0.0, \\ &0.0, 1.0, 1.0, 1.0, 0.0, \\ &1.0, 0.0, 0.0, 0.0, 0.0, \\ &0.0, 01000\} \end{aligned} \tag{3}$$

4 NDNN model building

4.1 NDNN structure building

DNN is a basic structure of deep learning network with at least one hidden layer. DNN can provide modelling for complex non-linear systems. Extra layers provide a much higher level of abstract features to enhance the capability of the model.

NDNN has a simple network structure. It is composed of three parts: input layers, hidden layers and output layers. The structure is shown in, including an input layer with 41 neurons, 4 hidden layers and 100 neurons per layer, 1 fully connected (FC) layer with 5 neurons, a softmax layer and an output layer with 5 neurons. The training result of each layer is treated as the input of the next layer (Fig. 2).

4.1.1 Input and hidden layers: The complex mapping relationship between input vector $\boldsymbol{X}$ and output vector $\boldsymbol{Y}$ is constructed in the network model. The input vector $\boldsymbol{X}$ is a 41-dimensional feature vector of one intrusion record. The output vector $\boldsymbol{Y}$ is a 5-dimensional probability vector with values between 0 and 1. Each value shows the probability which kind of intrusion (four types of attack and one normal data) the input belongs to. Sum of the five outputs ($y_1, y_2,\ldots,y_5$) is 1. Maximum of the output is regarded as the classification result of the current data.

As an example of a single hidden layer network with 100 neurons, there are N samples (x_i, y_i) ($i = 1,2,\ldots,N$). Output of the hidden layer neurons is expressed in the following equation [25]:

$$h = g(wx + b) \tag{4}$$

Output of all connected layer neurons is expressed in the following equation:

$$z_i^T = h(x_i)\beta + d \quad (i = 1, 2, \ldots, N) \tag{5}$$

$g(\cdot)$ is the activation function of NDNN hidden layer. w is the weight between input layer and hidden layer. b is the bias of hidden layer neurons. β is the weight between hidden layer and FC layer. d is the bias of FC layer neurons. Equation (5) can also be expressed as follows:

$$\boldsymbol{H\beta} + \boldsymbol{d} = \boldsymbol{Z} \tag{6}$$

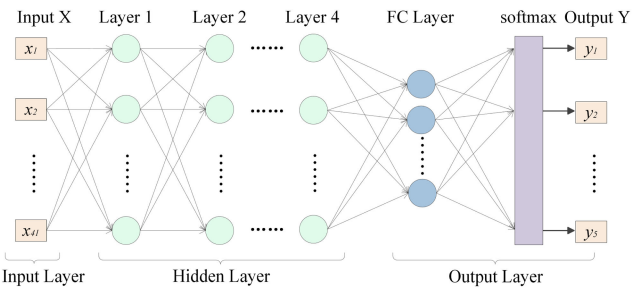

Fig. 2 *Designed NDNN model for IDS*

$\boldsymbol{H}$ is the output matrix of hidden layer. $\boldsymbol{\beta}$ is used to represent the connection weight matrix between hidden layer and FC layer. $\boldsymbol{Z}$ is the output matrix of FC layer.

$\boldsymbol{H}$, $\boldsymbol{\beta}$, $\boldsymbol{Z}$ are shown in the following equations [25]:

$$\boldsymbol{H} = \begin{bmatrix} g(w_1, b_1, x_1) & g(w_2, b_2, x_1) & \cdots & g(w_{100}, b_{100}, x_1) \\ g(w_1, b_1, x_2) & g(w_2, b_2, x_2) & \cdots & g(w_{100}, b_{100}, x_2) \\ \vdots & \vdots & \ddots & \vdots \\ g(w_1, b_1, x_n) & g(w_2, b_2, x_n) & \cdots & g(w_{100}, b_{100}, x_n) \end{bmatrix}_{n \times 100} \tag{7}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_{100}^T \end{bmatrix}_{100 \times 5} \quad \boldsymbol{Z} = \begin{bmatrix} z_1^T \\ z_2^T \\ \vdots \\ z_n^T \end{bmatrix}_{n \times 5} \quad \boldsymbol{d} = \begin{bmatrix} d_1^T \\ d_2^T \\ \vdots \\ d_n^T \end{bmatrix}_{n \times 5} \tag{8}$$

In NDNN, rectified linear unit (ReLU) is used as the activation function of the hidden layer neurons. ReLU is a non-linear

Table 3 Forty one features of an original intrusion data record

Description	Feature	Data attributes
basic features of individual TCP connections	duration	continuous
	protocol_type	symbolic
	service	symbolic
	flag	symbolic
	src_bytes	continuous
	dst_bytes	continuous
	land	symbolic
	wrong_fragment	continuous
	urgent	continuous
content features within a connection suggested by domain knowledge	hot	continuous
	num_failed_logins	continuous
	logged_in	symbolic
	num_compromised	continuous
	root_shell	continuous
	su_attempted	continuous
	num_root	continuous
	num_file_creations	continuous
	num_shells	continuous
	num_access_files	continuous
	num_outbound_cmds	continuous
	is_host_login	symbolic
	is_guest_login	symbolic
traffic features computed using a 2 s time window	count	continuous
	srv_count	continuous
	serror_rate	continuous
	srv_serror_rate	continuous
	rerror_rate	continuous
	srv_rerror_rate	continuous
	same_srv_rate	continuous
	diff_srv_rate	continuous
	srv_diff_host_rate	continuous
traffic features computed in and out a host	dst_host_count	continuous
	dst_host_srv_count	continuous
	dst_host_same_srv_rate	continuous
	dst_host_diff_srv_rate	continuous
	dst_host_same_src_port_rate	continuous
	dst_host_srv_diff_host_rate	continuous
	dst_host_serror_rate	continuous
	dst_host_srv_serror_rate	continuous
	dst_host_rerror_rate	continuous
	dst_host_srv_rerror_rate	continuous

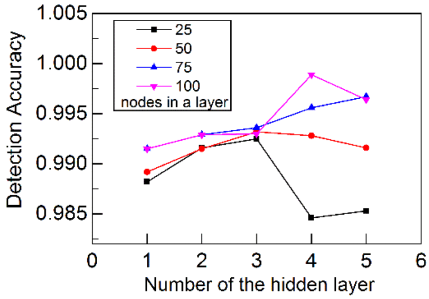

Fig. 3 *Network layer number, network node number and detection accuracy test of the proposed NDNN on KDD99 dataset*

Table 4 Details of the KDD 99 dataset

Intrusion category	Number of training data	Number of testing data
Probe	3723	384
DoS	356,691	34,767
U2R	41	11
R2L	1024	102
Normal	88,515	8763

activation function expressing the complex classification boundary better than linear activation function [26]. Due to ReLU, the gradient vanishing caused by sigmoid function can be prevented easily, and the derivation is easy. The ReLU function is shown in the following equation:

$$g(x) = \max(0, x) \tag{9}$$

In [15], three different learning rate (LR) values have been tested and finally they choose 0.1. Here a randomised optimisation method, Adam (adaptive moment estimation) [27], is used as the stochastic gradient descent optimiser in NDNN. The advantage of Adam is after bias correction, each iterative LR has a definite range, which makes the parameters more stable. The initial value of LR is 0.001. The value is dynamically changed by the Adam optimiser. Calculating different adaptive LRs for different parameters requires less memory. Compared with other adaptive learning algorithms, the convergence rate is increased and more effective, and the problems existed in other optimisation techniques, such as the disappearance of LR, the slow convergence or the loss function fluctuating caused by large variance parameter update, can be corrected.

4.1.2 FC layer – classification: NDNN network model is used to identify and classify unknown intrusion data. Outputs of neurons of hidden layers represent high-level features in the data and these data need to be classified into five types of intrusions. A FC layer is designed to map the features into five classes. There are five nodes in the FC layer. The 100-dimensional features from the fourth layer are mapped into five classes with this layer.

The last layer of the NDNN model is a softmax classifier. Softmax activation function is generally used for a network with more than one output neuron. softmax classifier is a kind of multiple-output competitive classification algorithm. Each output represents a classification category probability. Training softmax is a supervised process and labelled data is needed. To reduce the false positive, known attacks need to be classified into different categories in training. Normal data and unknown attacks are also added into the training dataset to improve the generalisation performance of the algorithm.

The softmax function is shown in the following equation:

$$y(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \quad j = 1, \ldots, K \tag{10}$$

z is the output of the FC layer. Each category has a z, and the softmax function is used to calculate the probability of the category. K is the number of intrusion categories, and here K is 5. Softmax layer gives five probability values. Finally, the largest probability max (y) is selected as the intrusion detection result showing which type the intrusion belongs to. If it is classified as an attack type, the response is triggered and the alert is called.

4.2 Network parameter selection

A number of feature learning experiments have been done to find the best network structure which has the best feature learning performance [15]. Fig. 3 shows several test accuracies of network structure with different network layer number, network node number.

Experiments show that no matter how many nodes are set in a layer, as the layer number increases from 1 to 3, the accuracy of testing is increasing (model structure is recorded as layer number-nodes number in a layer). While when the layer number increases to 4, model with 25 nodes a layer (model 4-25) and model with 50 nodes a layer (model 4-50) are overfitting and the accuracy starts to drop. The accuracy of the model 4-75 and model 4-100 is still increasing. When the layer number is added to 5, accuracy of the model 5-100 begins to drop. It indicates that 4-layer may be the best choice for the model with 100 nodes a layer. It can be treated as a local optimisation value. As the accuracy is already 0.998, we think it is high enough in test experiments and the model 4-100 is selected. The accuracy of the model with 75 nodes a layer is still increasing, while the accuracy is always not as high as a model with 100 nodes in a layer. Due to the satisfied model 4-100, no more experiments are done to explore the accuracy with more layers. Several parallel experiments also show that when there are 4 hidden layers and 100 nodes in each layer, the NDNN model has its best performance. So finally model 4-100 is chosen as the NDNN structure.

In the training procedure, NDNN network model is built to extract features from a large number of network intrusion data. Parameters of each network layer are optimised to make the later intrusion recognition performance better. Then the trained NDNN model with the best parameters and network structure is saved to classify the unknown network intrusion. In detection procedure, testing data is input the trained NDNN to do intrusion classification.

5 Experiment

KDD Cup 1999 dataset [23] and NSL-KDD dataset [28] are used in this research. The designed network is run in tensorflow 1.0 on a PC with an Intel Core i7–5820k (R) central processing unit (3.70 GHz), 16 GB memory and a GeForce GTX1060 GPU.

5.1 Dataset

The KDD99 dataset [23] and NSL-KDD dataset are usually used as a dataset to perform various performance tests on the designed IDS. There are four types of intrusion in the dataset: Probe, Dos, U2R and R2L. In KDD99 dataset, only 10% of the total dataset are selected, the dataset from 'kddcup.data_10_percent' contains 494021 connection records. About 90% of them are randomly selected as the training set, and the remaining 10% are used as the test set. The test set contains not only type of attack that has occurred in the training set, but also types of attack never occurred before. Besides, the performance of NDNN has also been tested on NSL-KDD. Similar operations of data processing have been done on this dataset. Traffic records distribution in the training and test data for normal and attack dataset is given in Tables 4 and 5.

5.2 Analysis of the experimental results

For the evaluation of IDS performance, five evaluation indexes have been used: accuracy (Acc), recall (R) [or detection rate (DR), computing methods of R and DR are the same], F-measure, false detection rate (FDR, which can also be treated as the false alarm) and missing alarm rate (MAR).

Table 5 Details of the NSL-KDD dataset

Intrusion category	Number of training data	Number of testing data
Probe	10,422	1235
DoS	41,407	4520
U2R	41	11
R2L	896	98
Normal	61,110	6233

Table 6 Types of detected intrusion

		Predicted	
		Attack	Normal
actual	attack	TP	FN
	normal	FP	TN

Table 7 Detection results of one single attack of KDD99 (DR, FDR and MAR)

Intrusion category	DR	FDR	MAR
Probe	0.9896	0.0001	0.0104
DoS	0.9997	0	0.0003
U2R	0.9091	0.0001	0.0909
R2L	0.9804	0.0001	0.0196
overall	0.9995	0.0003	0.0005

Table 8 Comparison results with other approaches based on the KDD99 dataset

Algorithm	DR	FDR	MAR
adaboost [29]	0.8340	0.1740	0.1660
auto-encoder Network [30]	0.9890	0.0110	0.0110
LSSVM-IDS + FMIFS [34]	0.9946	0.0013	0.0054
LSSVM-IDS + MIFS (β = 0.3) [34]	0.9938	0.0023	0.0062
LSSVM-IDS + FLCFS [34]	0.9847	0.0061	0.0153
LSSVM-IDS + All features [34]	0.9916	0.0097	0.0084
unoptimised DBN-PNN [20]	0.9931	—	0.0069
optimised DBN-PNN [20]	0.9914	—	0.0086
PCA-PNN [20]	0.9828	—	0.0172
PNN [20]	0.9904	—	0.0096
proposed algorithm	0.9995	0.0003	0.0005

The specific definitions of the five metrics are shown in equations (11)–(15). Explanation of True Positive (TP), False Negative(FN), False Positive (FP), True Negative (TN) used in these equations are described in Table 6.

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{11}$$

$$\text{DR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

$$F\text{-measure} = \frac{2 \times P \times R}{P + R} \tag{13}$$

$$\text{FDR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{14}$$

$$\text{MAR} = 1 - \text{DR} \tag{15}$$

First, labelled training set is used as the input of IDS to do feature learning. The best parameters are selected and saved after many experiments, and the trained IDS model is preserved. Then, the remaining 10% data is used as test dataset. Test data is input into the network to test the detection performance of single type attacks.

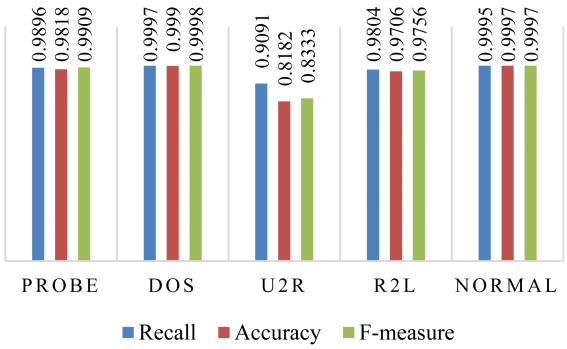

Fig. 4 *Detection results of one single attack of KDD99*

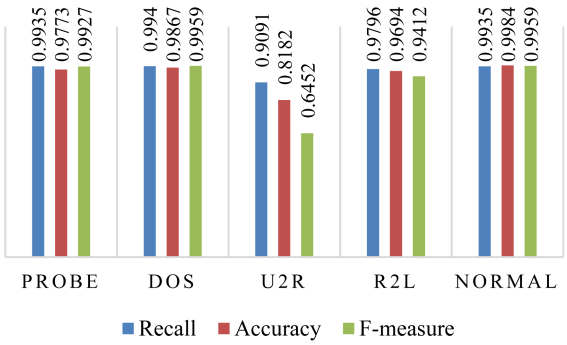

Fig. 5 *Detection results of one single attack of NSL-KDD*

Results of KDD99 and NSL-KDD are shown in Figs. 4 and 5 (Table 7).

Except U2R, accuracy of other types of intrusion is >0.98. As there are only 41 records of U2R intrusion, the detection result of U2R is not pleasant. This is obvious in Figs. 4 and 5. Besides, it shows that with larger data scale, the detection results are better. For example, comparing Table 4 and Fig. 4, number of the training data: DoS>Normal>Probe>R2L>R2L, and the accuracy has the similar rank. It can also be found in Table 5 and Fig. 5. So it can be suspected that if more U2R intrusion records are collected, the performance of the IDS will get better.

5.3 Comparative results of different methods

In order to verify the overall detection performance of the IDS model to intrusion data, results of several approaches are compared. As in these papers, KDD 99 dataset is used as the benchmark, DR, FDR and MAR of the NDNN proposed in this paper are also calculated. The results are shown in Table 7. With RBM in [17], the accuracy of intrusion detection is about 0.94, while the test dataset is not published. Adaboost [29], Auto-encoder Network [30], Bayesian Network [31], Flexible Neural Tree [32], Radial SVM [33] have been used in other literatures, and the experiment results show that the accuracy is not as well as LSSVM-IDS proposed in [34]. The proposed IDS model using NDNN is also compared with other ten methods. Detection rate, FDR and MAR are all shown in Table 8. The table shows generally deep learning based methods have better performance than traditional methods, such as Adaboost and LSSVM. The proposed method achieves 0.9995 DR and 0.0003 FDR in experiment on KDD99 dataset.

In [16], STL is used on NSL-KDD. The accuracy rate is higher than 98%, a little lower than 99%. *F*-measure can achieve 98.84%. Compared with the data in Fig. 5, we can see that with some data higher than 99%, the performance of the NDNN is a little better.

The proposed NDNN-based IDS can obtain a higher detection rate on the basis of low FDR and missing alarming rate. The overall performance of the system is better than that of other machine learning algorithms. It suggests that NDNN is a feasible IDS model.

6 Conclusion

In this paper, a DNN-based IDS is built. Experiment results show that capability and performance of the NDNN-based IDS are better than methods based on traditional machine learning method. Using NDNN network model to do intrusion detection is feasible, especially in this multi-feature dataset. However, there are still some problems need to be solved in the future. First, if there are too many parameters and difficult to tune, such as the number of nodes and layers, the increase of the network structure will lead to the exponential progression of computing time and other issues. In this paper, some of the parameters are decided by experiments. This is also used in some related papers [15]. While the risk is that maybe it is not the global best solution. So related optimisation algorithms and automatic parameter tuning methods will be studied to enhance the performance. Second, the IDS should be tested in a real network environment. High detection rate on dataset does not mean the similar performance in actual detection. So, more network simulation experiments need to be done in the future. Third, it is found that with larger training data scale, the detection performance is better. Adding more U2R and R2L intrusion records in the dataset is required.

7 Acknowledgments

The project is supported by the National Natural Science Foundation of China under grant 61136002, the Foundation of Shaanxi Educational Committee under grant 14JK1674 and the Graduate Innovation Fund in Xi'an University of Posts and Communications under grant 103-602080010.

8 References

[1] Dong, H.S., An, K.K., Choi, S.C., *et al.*: 'Malicious traffic detection using *K*-means', *J. Korean Inst. Commun. Inf. Sci.*, 2016, **41**, (2), pp. 277–284

[2] Al-Yaseen, W.L., Othman, Z.A., Nazri, M.Z.A.: 'Multi-level hybrid support vector machine and extreme learning machine based on modified *K*-means for intrusion detection system', *Expert Syst. Appl.*, 2017, **67**, pp. 296–303

[3] Kaur, A., Pal, S.K., Singh, A.P.: 'Hybridization of *K*-means and firefly algorithm for intrusion detection system', *Int. J. Syst. Assur. Eng. Manage.*, 2017, **9**, (4), pp. 1–10

[4] Tahir, H.M., Hasan, W., Said, A.M., *et al.*: 'Hybrid machine learning technique for intrusion detection system'. 'Computer science for improving the quality of life'. Int. Conf. Computing and Informatics, Icoci, Istanbul, Turkey, 2015

[5] Yang, Q., Lou, J., Liu, S., *et al.*: 'Naïve Bayes decision tree hybrid approach for intrusion detection system', *Bul. Tek. Elektro Dan Inform.*, 2013, **2**, (3), pp. 2477–2487

[6] Boujnouni, M.E., Jedra, M.: 'New intrusion detection system based on support vector domain description with information gain metric', *Int. J. Netw. Secur.*, 2018, **20**, (1), pp. 25–34

[7] Gao, N., Gao, L., Gao, Q., *et al.*: 'An intrusion detection model based on deep belief networks'. Second Int. Conf. Advanced Cloud and Big Data, Huangshan, People's Republic of China, 2014, pp. 247–252

[8] Lin, W.-C., Ke, S.-W., Tsai, C.-F.: 'CANN: an intrusion detection system based on combining cluster centers and nearest neighbors', *Knowl.-Based Syst.*, 2015, **78**, pp. 13–21

[9] Zhu, W., Deng, M., Zhou, Q.: 'An intrusion detection algorithm for wireless networks based on ASDL', *IEEE/CAA J. Autom. Sinica*, 2018, **5**, (1), pp. 92–107

[10] Wang, W., He, Y., Liu, J., *et al.*: 'Constructing important features from massive network traffic for lightweight intrusion detection', *IET Inf. Sec.*, 2015, **9**, (6), pp. 374–379

[11] Sujendran, R., Arunachalam, M.: 'Hybrid fuzzy adaptive wiener filtering with optimization for intrusion detection', *ETRI J.*, 2015, **37**, (3), pp. 502–511

[12] Kyriakopoulos, K.G., Aparicio-Navarro, F.J., Parish, D.J.: 'Manual and automatic assigned thresholds in multi-layer data fusion intrusion detection system for 802.11 attacks', *IET Inf. Sec.*, 2014, **8**, (1), pp. 42–50

[13] Costa, K.A.P., Pereira, L.A.M., Nakamura, R.Y.M., *et al.*: 'A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks', *Inf. Sci.*, 2015, **294**, pp. 95–108

[14] Bamakan, S.M.H., Wang, H., Shi, Y.: 'Ramp loss *K*-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem', *Knowl.-Based Syst.*, 2017, **126**, pp. 113–126

[15] Yin, C.L., Zhu, Y.F., Fei, J.L., *et al.*: 'A deep learning approach for intrusion detection using recurrent neural networks', *IEEE. Access.*, 2017, **5**, pp. 21954–21961

[16] Javaid, A., Niyaz, Q., Sun, W., *et al.*: 'A deep learning approach for network intrusion detection system'. Eai Int. Conf. Bio-Inspired Information and Communications Technologies, New York City, NY, USA, 2016, pp. 21–26

[17] Fiore, U., Palmieri, F., Castiglione, A., *et al.*: 'Network anomaly detection with the restricted boltzmann machine', *Neurocomputing*, 2013, **122**, pp. 13–23

[18] Do, K., Tran, T., Venkatesh, S.: 'Multilevel anomaly detection for mixed data', 2016, arXiv:1610.06249

[19] Alrawashdeh, K., Purdy, C.: 'Toward an online anomaly intrusion detection system based on deep learning'. IEEE Int. Conf. Machine Learning and Applications, Anaheim, CA, USA, 2017, pp. 195–200

[20] Zhao, G., Zhang, C., Zheng, L.: 'Intrusion detection using deep belief network and probabilistic neural network'. IEEE Int. Conf. Computational Science and Engineering, Guangzhou, People's Republic of China, 2017

[21] Potluri, S., Diedrich, C.: 'Accelerated deep neural networks for enhanced intrusion detection system'. IEEE Int. Conf. Emerging Technologies and Factory Automation, Berlin, Germany, 2016

[22] Roy, S.S., Mallik, A., Gulati, R., *et al.*: 'A deep learning based artificial neural network approach for intrusion detection', 2017, pp. 44–53

[23] Stolfo, S.J., Stolfo, S.J.: 'KDD cup 1999 dataset', 1999, available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[24] Harris, D.M., Harris, S.L.: 'Digital design and computer architecture', *Microelectron. Reliab.*, 2012, **44**, pp. 1279–1280

[25] Ding, S., Guo, L., Hou, Y.: 'Extreme learning machine with kernel model based on deep learning', *Neural Comput. Appl.*, 2017, **28**, (8), pp. 1975–1984

[26] Dahl, G.E., Sainath, T.N., Hinton, G.E.: 'Improving deep neural networks for LVCSR using rectified linear units and dropout'. IEEE Int. Conf. Acoustics, Speech and Signal Processing, 2013, pp. 8609–8613

[27] Kingma, D.P., Ba, J.: 'Adam: a method for stochastic optimization', Computer Science, 2014

[28] Chae, H.S., Jo, B.O., Choi, S.H., *et al.*: 'Feature selection for intrusion detection using NSL-KDD', Proceedings of the 6th WSEAS World Congress: Applied Computing Conference (ACC'13), Nanjing, China, 2013, pp. 184–187

[29] Dong, C., Zhou, G., Liu, Y. J., *et al.*: 'The detection of network intrusion based on improved adaboost algorithm', Journal of Sichuan University, 2015

[30] Chun-Lin, L.I., Huang, Y.J., Wang, H., *et al.*: 'Detection of network intrusion based on deep learning', in Ding, S. (Ed.): '*Information security & communications privacy*' (Beijing, China, 2014), pp. 68–72

[31] Chebrolu, S., Abraham, A., Thomas, J.P.: 'Feature deduction and ensemble design of intrusion detection systems', *Comput. Secur.*, 2005, **24**, (4), pp. 295–307

[32] Chen, Y., Abraham, A., Yang, B.: 'Feature selection and classification using flexible neural tree', *Neurocomputing*, 2006, **70**, (1-3), pp. 305–313

[33] Chandrasekhar, A.M., Raghuveer, K.: '*An effective technique for intrusion detection using neuro-fuzzy and radial SVM classifier*' (Springer New York, 2013)

[34] Ambusaidi, M.A., He, X., Nanda, P., *et al.*: 'Building an intrusion detection system using a filter-based feature selection algorithm', *IEEE Trans. Comput.*, 2016, **65**, (10), pp. 2986–2998