Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks

Yalei Ding
Southeast University, China
220163521@seu.edu.cn

Yuqing Zhai
Southeast University, China
yqzhai@seu.edu.cn

ABSTRACT

With the increment of cyber traffic, there is a growing demand for cyber security. How to accurately detect cyber intrusions is the hotspot of recent research. Traditional Intrusion Detection Systems (IDS), based on traditional machine learning methods, lacks reliability and accuracy. In this paper, we build an IDS model with deep learning methodology. Instead of the traditional machine learning used in previous researches, we think deep learning has the potential to perform better in extracting features of massive data considering the massive cyber traffic in real life. Therefore, we propose to train an IDS model based on Convolution Neural Networks (CNN), a typical deep learning method, using entire NSL-KDD dataset. We study the performance of the model using multi class classification to compare with the performance of traditional machine learning methods including Random Forest (RF) and Support Vector Machine (SVM), and deep learning methods including Deep Belief Network (DBN) and Long Short Term Memory (LSTM). The experimental results show that the performance of our IDS model is superior to the performance of models based on traditional machine learning methods and novel deep learning methods in multi-class classification. Our model improves the accuracy of the intrusion detection and provides a new research direction for intrusion detection.

CCS Concepts

• Security and privacy → Intrusion/anomaly detection and malware mitigation → Intrusion detection systems • C Computing methodologies → Machine learning → Machine learning approaches→Neural networks.

Keywords

Intrusion Detection System, deep learning, NSL-KDD, Convolution Neural Networks.

1. INTRODUCTION

Due to the rapid growth of internet construction, it is critical to identify intrusions that threat network security. In 1980, Intrusion Detection System (IDS) was proposed to provide solid protection for the equipment against malicious software attacks, such as denial of service (DoS), by analyzing patterns of captured data [1]. IDS is able to detect attacks, for example, it denies or prevents illegitimate traffic when traffic acts like Denial of service (DoS). Generally, intrusion detection can be seen as the process of solving a classification problem. One of the problems in some existing IDS is that their detection accuracy is low. Another problem is that they rely on signatures of known attacks, which means they are incapable of detecting unknown attacks.

CSAI '18, December 8–10, 2018, Shenzhen, China

ACM ISBN 978-1-4503-6606-9/18/12…$15.00
DOI: https://doi.org/10.1145/3297156.3297230

In order to solve these disadvantages, traditional machine learning methodologies have been widely used to distinguish several categories of attacks [2]. However, most of the traditional machine learning methodologies are shallow learning methods which usually emphasize feature engineering and selection. Moreover, they usually cannot provide an efficient solution for the massive intrusion data classification problem that caused by tremendous amount of network application traffic [3]. Under the circumstance that massive data often needs high-dimensional learning, shallow learning is unsuited to requirements of analysis and forecast. In contrast, deep learning has the potential to extract better representations for creating better models [4]. As a result, researchers in this field are trying to develop IDS that are based on deep learning.

Since Professor Hinton proposed the theory of deep learning in 2006, deep learning theory and technology undergone a swift rise and caught lots of interest in the field of machine learning [5]. A new era of artificial intelligence has been opened and offering a completely new way to develop intelligent IDS because deep learning theory and technology has been going through a rapid development in recent years. Due to growing computational resources, convolutional neural networks (CNN), created decades ago, has recently generated a significant development in the domain of deep learning. The special architecture of CNN enhances the representations from data. CNN is mainly widely used in the field of image recognization and modeling sentence, while it has not been used in the field of intrusion detection. Consequently, we propose a deep learning approach for an intrusion detection system using CNN.

The main contributions of this paper are summarized as following:

1) We present the architecture and implementation of the detection system based on CNN. Moreover, we study the performance of the model using multiclass classification.
2) We utilize the entire NSL-KDD dataset instead of extracting part of this dataset which many former researchers did.
3) By contrast, we compare the performance of our model to the performance of traditional machine learning methods (RF and SVM) and deep learning methods (DBN and LSTM) in multi class classification using the entire NSL-KDD dataset. The experimental results illustrate that CNN based IDS is more suitable than traditional machine learning methods and novel proposed deep learning methods in the area of intrusion detection.

The remainder of this paper is organized as follows. In Section II, we review the related research in the field of intrusion detection. We introduce the input of CNN-based model and describe the architecture of CNN-based IDS with multi-stage features in

Section III. Next, we introduce the NSL-KDD dataset. Meanwhile, we select the performance evaluation measures, take the experiments in the dataset and compare the results with a few previous studies in Section IV. Finally, the conclusion is discussed in Section V.

2. RELATEDWORK

The notion of intrusion detection (ID) is firstly introduced by Anderson in 1980. He built an anomaly-based intruder detection methodology utilizing system monitoring to detect abnormal activities [6]. After that, a number of researches have been completed based on the traditional machine learning, making lots of progress for IDS. A novel support vector machine (SVM) model combining kernel principal component analysis (KPCA) with genetic algorithm (GA) was proposed by Kuang et al. in 2014\cite{b7}. In 2014, Li et al. proposed a new intrusion detection system based on K-nearest neighbor (KNN) classification algorithm in wireless sensor network [8]. Ingre and Yadav evaluated the performance of NSL-KDD dataset using ANN. The result obtained a higher detection rate in both binary class and five class classification [9]. In 2016, Farnaaz and Jabbar presented a model for intrusion detection system using Random Forest (RF) classifier which is an ensemble classifier [10]. Other traditional machine learning methodologies are also presented in this paper [11].

Recently, deep learning has been applied for intrusion detection. Qu et al. presented an intrusion detection model based on Deep Belief Network (DBN) and analyzed the NSL-KDD dataset [12]. They just extracted 10,000 records as training data from 125,973-records train set and 2,000 test data from 22,544-records test set. Kim et al. applied long short term memory (LSTM) architecture to RNN and trained the IDS using KDD Cup'99 dataset. By comparing with other IDS classifiers, LSTM-RNN IDS achieved great accuracy with slightly higher FAR [13]. Although their experiment result was amazing, they only used extracted, even fewer, 1,630 records from NSL-KDD which has 125,973 records. Meanwhile, they used train set as test set, which means their experiment dataset may be unrepresentative. Abolhasanzadeh propose a method based on dimensionality reduction and auto-encoder bottleneck feature extraction. He conducted several experiments on NSL-KDD dataset to investigate the effectiveness of the proposed approach [14]. Fiore et al. explored a new detection approach based on Discriminative Restricted Boltzmann Machine (DBM) to combine the expressive power of generative models with well-classify capabilities [15]. However, there is a lack of study of the performance compared with other methodologies.

With the continuous development of big data and computing power, deep learning methods have been widely utilized in various fields and achieved amazing results. Deep learning is gradually applied to the field of intrusion detection. To the best of our knowledge, CNN has been proven to be a good classifier in many other fields [16, 17] while it has not been exploited in the field of intrusion detection. Therefore, following this line of thinking, we propose an intrusion detection technique using CNN in this paper. Besides, we utilize the entire NSL-KDD dataset with a separate training and two testing sets rather than just part of this dataset to evaluate their performances in multi class intrusion classification, and we compared it with RF, SVM, DBN and LSTM proposed by previous researchers.

3. CNN-BASED IDS WITH MULTI-STAGE FEATURES

In previous chapter, we discover that deep learning is gradually applied to the field of intrusion detection and CNN has not been applied in this field. In this chapter, we will try to build CNN based IDS for NSL-KDD dataset.

3.1 CNN-based Model Input

Traditional CNN architecture requires the input to be 3-dimensional (m rows, n columns, p depths) - a picture which has p channels with width m and height n. However, in our paper, we consider the 1-dimensional input data as 3-dimensional picture whose height is 1 and only has 1 channel. Namely, we transform NSL-KDD dataset into the input format of 1-dimensional convolution architecture. In addition, NSL-KDD dataset contains numeric and some non-numeric features. They need to be converted as numeric attribute because the training input and testing input fed to CNN should be numeric matrix. Moreover, we use one-hot encoder to translate category features in dataset since one-hot encoder could solve the problem that converting category to integer which may imply order importance of encoded value. After using one-hot encoder to convert dataset into numeric matrix, we use the following method which keeps values in the range of 0 to 1 to normalize input numerical matrix (equation 1).

$$x_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1)$$

In the equation, x_i is the value of each data point, x_{min} is the minima among all data points, x_{max} is the maxima among all data points. The data point is normalized between 0 and 1. After applying this step, the data is normalized and ready for being pushed into input.

3.2 CNN-based Model with Multi-stage Features

3.2.1 CNN architecture

A common CNN that is widely applied to analyze visual imagery consists of an input, and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. Convolutional layers apply a convolution operation to the former input, and pass the result to the next layer. A convolution operation is operated by corresponding convolutional neuron which processes data only for its receptive field. Meanwhile, weights are shared for each receptive field in the same filter, which can reduce memory footprint and improve the performance. There are mainly two types of pooling layers, including max pooling and average pooling.

3.2.2 Multi-stage features

Multi-Stage features (MS) are obtained by branching out outputs of all convolutional stages into the later layer. They provide richer representations than all layers in one line by adding complementary information such as local information and details lost by higher levels. MS features have consistently improved the performance for our work.

In our model (Figure 1), there are 3 stacked stages of feature extraction using convolution module which contains a convolution layer followed by a max pooling and no normalization layer. These stages create deep features of input. Then, 2 dense layers are followed. Next, softmax is used to extract 5-dimensional features which equals to the number of target

classes. Finally, we concatenate 3 stages deep features with 5-dimensional features, and feed them to dense layer with softmax classifier.

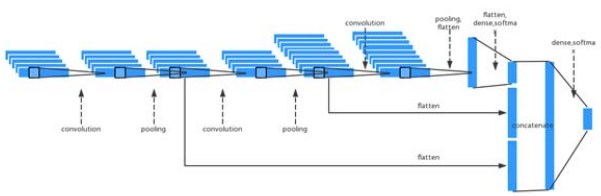

Figure 1. CNN-based IDS Architecture

Table 1. CNN-based Model

layer (type)	output Shape	param
input1 (InputLayer)	(-1, 1, 122)	0
conv1d1 (Conv1D)	(-1, 1, 62)	15190
activation1 (Activation)	(-1, 1, 62)	0
max pooling1d1 (MaxPooling1D)	(-1, 1, 62)	0
flatten1 (Flatten)	(-1, 1, 62)	0
dropout1 (Dropout)	(-1, 1, 62)	0
conv1d2 (Conv1D)	(-1, 1, 62)	15438
activation2 (Activation)	(-1, 1, 62)	0
max pooling1d2 (MaxPooling1D)	(-1, 1, 62)	0
flatten2 (Flatten)	(-1, 1, 62)	0
dropout2 (Dropout)	(-1, 1, 62)	0
conv1d3 (Conv1D)	(-1, 1, 124)	61628
activation3 (Activation)	(-1, 1, 124)	0
max pooling1d3 (MaxPooling1D)	(-1, 1, 124)	0
flatten3 (Flatten)	(-1, 1, 124)	0
dropout3 (Dropout)	(-1, 1, 124)	0
dense1 (Dense)	(-1, 256)	32000
dropout4 (Dropout)	(-1, 256)	0
dense2 (Dense)	(-1, 5)	1285
softmax (Softmax)	(-1, 5)	0
concatenate1 (Concatenate)	(-1, 253)	0
dense3 (Dense)	(-1, 5)	1285
softmax (Softmax)	(-1, 5)	0

Our CNN-based model is defined in Table 1. It has a 3-stage architecture where Multi-Stage features (MS) are fed to a 2-layer classifier. Features in the 1st stage, extracted by convolutional layer and max pooling layer are branched out and then concatenated to 2nd and 3rd stage features. All convolution layers are followed by a ReLU activation layer and a max pooling layer which has stride 1 to maintain the shape of previous convolution layer. The first convolution block consists of a layer with regular convolutions (kernel_size=2, strides=1), a ReLU activation layer and a max pooling layer (kernel_size=2, strides=1). The second convolution block contrasts a layer with regular convolutions (kernel_size=4, strides=1), a ReLU activation layer and a max pooling layer (kernel_size=4, strides=1). Next, a convolution block contrasts a layer with regular convolutions (kernel_size=8, strides=1), a ReLU activation layer and a max pooling layer (kernel_size=8, strides=1). After that, it is followed by a flatten layer which reduces the output dimension from 3 to 2. The fully connected layer - dense layer which is followed by a dropout layer, is added after the previous layer. A dense layer reduces the feature size to 5 before the softmax layer which creates 5-dimensional features. Next, we concatenate 3-stages deep features and 5-dimensional features. The final dense layer reduces the feature size to 5 before the final softmax layer.

Our IDS models are trained in Keras which is based on Tensorflow using Adam as the optimizer. However, contrary to training large models, we use less regularization and data augmentation techniques because small models have less trouble with over fitting. Additionally, we find that it is important to put very little or no weight decay (l2 regularization) on filters since there are little parameter in them.

4. EXPERIMENT AND RESULTS

In previous chapter, we elaborate the model structure. In this chapter, we will experiment the model in NSL-KDD dataset.

4.1 Dataset Description and Analysis

The methodology discussed in this paper is applied on the entire NSL-KDD dataset. The NSL-KDD dataset was proposed to deal with inherent problems of the KDD Cup 1999 dataset which contain too many redundant records [18]. Although it is quite old and not a perfect representation of existing real networks, it is continuously an index which is used to compare the NIDS models in common researches. In the latest literature [19, b20, b21], all the researchers use the NSL-KDD as the benchmark dataset.

NSL-KDD dataset includes three sub-files: $KDDTrain^{+}$, $KDDTest^{+}$ and $KDDTest^{-21}$. There are 125,973 network traffic samples in the $KDDTrain^{+}$ dataset, 22,554 network traffic samples in the $KDDTest^{+}$ dataset and 11850 network traffic samples in the $KDDTest^{-21}$ dataset. There are 41 features, 1 class label and 1 difficulty label for each traffic record. The features include basic features (No.1-No.10), content features (No.11 - No.22), and traffic features (No.23 - No.41) as shown in the following example.

0,udp,private,SF,44,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4,3,0.00,

0.00,0.00,0.00,0.75,0.50,0.00,255,254,1.00,0.01,0.01,0.00,0.00,

0.00,0.00,0.00,snmpguess,12

Since the imbalance of detailed category in NSL-KDD dataset, it is difficult to predict category using origin class label. In the following experiment, according to their characteristics, class label of records in the dataset are categorized into 5 main categories (Normal, Dos, Probe, U2R and R2L). There are some specific attack types that only exist in testing set, which allows it to provide a more realistic theoretical basis for CNN-based IDS. Below are describe of Dos, Probe, R2L and U2R (table 2):

- Denial of Service (DoS) - This attack occupies too much computing or memory resource so that the machine cannot handle legitimate requests and access.
- Probe - This attack gathers information about potential vulnerabilities of the target system that can be used to launch attacks lately.
- Remote to Local (R2L) - Attacker does not have access to the victim machine, hence tries to gain local access as a user of that machine.
- User to Root (U2R) - Using this attack, attackers access the system as a normal user and exploit some vulnerability to gain root access to the system.

There are 4 non-numeric attributes: *protocol_type*, *service, flag and class*. For example, the feature *protocol_type* has three types of attributes: tcp, udp, and icmp, and its one-hot encoder values are binary vectors [1,0,0], [0,1,0] and [0,0,1]. Similarly, the feature *service* has 70 types of attributes, the feature *flag* has 11

types of attributes, and *class* have 5 types of attributes. In this way, 41-dimensional features are mapped into 122-dimensional features after transformation, prediction target is mapped into 5 categories classification (table 3).

Table 2. Attacks in the NSL-KDD Dataset

Category	Training Set	Testing Set
DoS	back, land, Neptune, pod, smurf, teardrop	apache2, back, land, mailbomb, Neptune, pod, smurf, teardrop, worm processtable, udpstorm
Probe	ipsweep, nma, portsweep, satan	ipsweep, mscan, nmap, portsweep, saint, satan
R2L	spy, warezclient, ftpwrite, guesspasswd, imap, multihop, phf, warezmaster	ftpwrite, guesspasswd, httptunnel, imap, multihop, named, phf, sendmail, snmpgetattack, snmpguess, wxlock, warezmaster, xsnoop
U2R	bufferoverflow, ps, loadmodule, rootkit	bufferoverflow, ps, perl, loadmodule, sqlattack, xterm
normal	normal	normal

Table 3. Numbers of Attacks in Dataset

	Total	Normal	Dos	Probe	R2L	U2R
$KDDTrain^{+}$	125973	67343	45927	11656	995	52
$KDDTest^{+}$	22544	9711	7460	2421	2885	67
$KDDTest^{-21}$	11850	2152	4344	2402	2885	67

4.2 Evaluation Metric

Accuracy (AC) is most important performance indicator of intrusion detection that is used to measure the performance of the CNN-based IDS model. In addition to the accuracy, we considered the detection rate and false positive rate. In intrusion detection field, we have the following notation:

Accuracy: the percentage of the records number classified correctly over total the records in Equation (2).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2)$$

True Positive Rate (TPR): also known as Detection Rate (DR), is the percentage of the anomaly records number correctly flagged as anomaly over the total number of anomaly records in Equation (3).

$$DR = TPR = \frac{TP}{TP + FN} \quad (3)$$

False Positive Rate (FPR): the percentage of the normal records number wrongly flagged as anomaly is divided by the total number of normal records in Equation (4).

$$FPR = \frac{FP}{FP + TN} \quad (4)$$

True Positive (TP) represents the amount of anomaly records that are truly identified as anomaly, False Positive (FP) represents the amount of normal records that are identified as anomaly, False Negative (FN) represents the amount of anomaly records that are identified as normal, True Negative (TN) represents the number of normal records that are identified as normal.

Obviously, IDS is eager to achieve a higher accuracy and a balance between higher DR and lower FPR.

4.3 Experiment

In this research, we use deep learning frameworks - Keras. The experiment is performed on PyCharm using Intel Core i7-6700HQ CPU @ 2.60GHzb x 8 with 8GB RAM and GeForce GTX 960M. We design an experiment to study the training and testing performed on 122 prepossessing features using NSL-KDD dataset for five-class classification. In order to compare with other traditional machine learning methods and novel deep learning methods, contrast experiments are designed at the same time. We compare performances of RF, SVM, DBN and LSTM in the five-category classification.

4.4 Results Analysis

Using $KDDTrain^{+}$ as the training set and $KDDTest^{+}$ and $KDDTest^{-21}$ as the testing set, the experimental result in table 4 shows the accuracy of methods proposed by previous researchers and our model. The overall two testing accuracy of five class classification with CNN are 80.1321% and 62.3206% respectively. Obviously, in $KDDTest^{+}$ and $KDDTest^{-21}$, accuracy of our model is much better than traditional machine learning methods and novel deep learning methods.

Table 4. Accuracy for each model

	RF	SVM	DBN	LSTM	CNN
$KDDTest^{+}$	74.1882%	71.3094%	71.9127%	73.1827%	80.1321%
$KDDTest^{-}$	51.0126%	45.5443%	46.7341%	49.375%	62.3206%

Table 5. DR of $KDDTest^{+}$

	Normal	Dos	Probe	R2L	U2R
RF	92.5445%	82.9088%	57.9512%	5.1993%	0.0993%
SVM	92.9255%	74.5576%	61.5448%	2.1313%	0.0222%
DBN	93.6978%	75.2412%	61.9578%	2.4365%	0.0434%
LSTM	93.3862%	73.1827%	67.2456%	6.3423%	0.0819%
CNN	96.7356%	85.2681%	73.9776%	17.7816%	8.9552%

Table 6. FPR of $KDDTest^{+}$

	Normal	Dos	Probe	R2L	U2R
RF	36.7412%	5.4826%	1.3417%	0.0305%	0.0044%
SVM	42.5153%	2.6252%	3.0611%	0.1778%	0.0163%
DBN	41.5335%	5.2108%	1.0733%	0.0739%	0.0156%
LSTM	39.4480%	4.6759%	1.3223%	0.0610%	0.0082%
CNN	29.5098%	1.9291%	1.7194%	0.2746%	0.0044%

Table 7. DR of $KDDTest^{-21}$

	Normal	Dos	Probe	R2L	U2R
RF	66.4033%	70.5801%	57.6602%	5.7192%	0.0833%
SVM	68.7267%	56.3075%	61.2822%	2.3473%	0.0333%
DBN	72.2118%	57.5046%	61.8651%	2.5637%	0.0634%
LSTM	93.3862%	73.1827%	67.2456%	6.3423%	0.0819%
CNN	96.7356%	85.2681%	73.9776%	17.7816%	8.9552%

Table 8. FPR of $KDDTest^{-21}$

	Normal	Dos	Probe	R2L	U2R
RF	48.5976%	10.8579%	2.8577%	0.0391%	0.0008%
SVM	56.2590%	5.0892%	6.5093%	0.1313%	0.0163%
DBN	54.9185%	10.2584%	4.2861%	0.1796%	0.0144%
LSTM	51.6664%	9.2703%	4.9657%	0.1339%	0.0077%
CNN	38.9667%	3.8769%	3.6515%	0.5465%	0.0008%

Meanwhile, table 5 shows the DR of $KDDTest^{+}$; table 6 shows the FPR of $KDDTest^{+}$; table 7 shows the DR of $KDDTest^{-21}$; table 8 shows the FPR of $KDDTest^{-21}$. Results in table 5, 6, 7 and 8 state clearly that our model achieve high DR and low FPR. In addition, compare DR and FPR of $KDDTest^{+}$ and $KDDTest^{-21}$, we can find that results of R2L and U2R are relatively quite small for all methods, which might because of the insufficiency of their records in the dataset. However, our model still can detect some of them. In general, our model obtains high accuracy with high DR and low FPR, which is evidently superior to traditional machine learning classification methods and novel deep learning methods in 5-class classification.

5. CONCLUSION

Traditional machine learning methodologies cannot efficiently detect new intrusion and deep learning have the potential to extract better representations to create better models. Hence, in this paper, we apply CNN which is a famous deep learning method in computer vision field to intrusion detection and introduce CNN-based IDS. The CNN-based IDS model with multi-stage features has a strong modeling ability for intrusion detection compared with traditional machine learning methods (RF and SVM) and novel deep learning methods(DBN and LSTM). Our model obtains a higher Accuracy with high DR and low FPR. The model can effectively improve both the accuracy of intrusion detection and the ability to recognize the intrusion type. By comparing the results with those of other methods, we have shown the potential of using CNN which is already proved to be a good classifier in many other fields for intrusion detection. In the future research, we will pay attention to improve Accuracy and DR of U2R and R2L, meanwhile, we need to reduce FPR of Dos. In addition, avoiding exploding and vanishing gradients are also need to be concerned, if we continue to add more convolution layers to achieve better performance

6. REFERENCES

[1] SANS Institute.(2018, Oct.) The History and Evolution of Intrusion Detection. [Online].Available: https://www.sans.org/reading-room/whitepapers/detection/history-evolution-intrusion-detection-344.

[2] Sarkar T, Das N. Survey on Host and Network Based Intrusion Detection System[J]. 2014, 6(2):2266-2269.

[3] Yin C, Zhu Y, Fei J, et al. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks[J]. IEEE Access, 2017, 5(99):21954-21961.

[4] MLecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553):436.

[5] Schmidhuber J. Deep Learning in neural networks: An overview.[J]. Neural Netw, 2015, 61:85-117.

[6] Javaid A, Niyaz Q, Sun W, et al. A Deep Learning Approach for Network Intrusion Detection System[C]// Eai International Conference on Bio-Inspired Information and Communications Technologies. ICST, 2016:21-26.

[7] Kuang F, Xu W, Zhang S. A novel hybrid KPCA and SVM with GA model for intrusion detection[J]. Applied Soft Computing Journal, 2014, 18(C):178-184.

[8] Li W, Yi P, Wu Y, et al. A New Intrusion Detection System Based on KNN Classification Algorithm in Wireless Sensor Network[J]. Journal of Electrical and Computer Engineering, 2014, 2014(5):1-8.

[9] Ingre B, Yadav A. Performance analysis of NSL-KDD dataset using ANN[C]// International Conference on Signal Processing and Communication Engineering Systems. IEEE, 2015:92-96.

[10] Farnaaz N, Jabbar M A. Random Forest Modeling for Network Intrusion Detection System[J]. Procedia Computer Science, 2016, 89:213-217.

[11] Buczak A L, Guven E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection[J]. IEEE Communications Surveys and Tutorials, 2017, 18(2):1153-1176.

[12] Qu F, Zhang J, Shao Z, et al. An Intrusion Detection Model Based on Deep Belief Network[C]// Vi International Conference. 2017:97-101.

[13] Kim J, Kim J, Thu H L T, et al. Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection[C]// International Conference on Platform Technology and Service. IEEE, 2016:1-5.

[14] Abolhasanzadeh B. Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features[C]// Information and Knowledge Technology. IEEE, 2015:1-5.

[15] Fiore U, Palmieri F, Castiglione A, et al. Network anomaly detection with the restricted Boltzmann machine[J]. Neurocomputing, 2013, 122:13-23.

[16] Gibiansky A.(2018, Oct.) Convolutional Neural Networks[Online]. Available: http://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/.

[17] Y. LeCun.(2018, Oct.) Learning Invariant feature Hierarchies, 2012. [Online]. Available: http://yann.lecun.com/exdb/publis/pdf/lecun-eccv-12.pdf.

[18] (2018, Oct.) KDD Cup 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/.

[19] Paulauskas N, Auskalnis J. Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset[C]// Electrical, Electronic and Information Sciences. IEEE, 2017:1-5.

[20] Shone N, Ngoc T N, Phai V D, et al. A Deep Learning Approach to Network Intrusion Detection[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2018, 2(1):41-50.

[21] Naseer S, Saleem Y, Khalid S, et al. Enhanced Network Anomaly Detection Based on Deep Neural Networks[J]. IEEE Access, 2018, PP(99):1-1.