Intrusion Detection Using Convolutional Recurrent Neural Network

Tongtong Su
Tianjin Normal University
Tianjin, China
sutongwyf923@163.com

Huazhi Sun*
Tianjin Normal University
Tianjin, China
sunhuazhi@mail.tjnu.edu.cn

Sheng Wang
Tianjin Normal University
Tianjin, China
wangsheng404@163.com

ABSTRACT
There are two main problems in the current abnormal network traffic monitoring methods: feature dependence and low accuracy. To solve these problems, we propose a novel deep learning model OCL based on network traffic in this paper. Our OCL model consists of one-dimensional convolutional neural network and long short-term memory. Convolutional neural network is first used to extract features from the image representation of original network traffics. Secondly, LSTM is used to combine the extracted features over time from the obtained features via convolutional and pooling operation. Finally, the output of the LSTM model is fed to fully connection network for classification. End-to-end representation learning is adopted to automatically learn the key features of network traffics. This can describe the network traffic behavior better and improve the ability of anomaly detection effectively. To verify the effectiveness of OCL model, it is comprehensively evaluated on the NSL-KDD dataset and get the best results. Experimental results show that the OCL model achieves high accuracy, which has better performance than other detection methods.

CCS Concepts
•Security and privacy → Intrusion/anomaly detection and malware mitigatio → Intrusion detection systems → Artificial immune systems

Keywords
Intrusion detection; NSL-KDD dataset; Deep Learning; CNN; LSTM.

1. INTRODUCTION
Network traffic detection can detect unknown attacks and provide important support for network security, which has been an effective means of network security [1]. The key of intrusion detection is to distinguish all kinds of malicious attack traffics from normal network traffics. Hence, intrusion detection can be considered as a network traffic classification problem. ie., network flow can be divided into two categories (normal traffics and malicious traffics), and can also be divided into five categories: Normal, DoS (Denial of Service attacks), R2L (Root to Local attacks), U2R (User to Root attack) and Probe (Probing attacks).

The main motivation of intrusion detection is to improve the accuracy of classifiers in effectively identifying intrusive traffics. At present, most network traffic classification methods are based on traditional machine learning methods. In [2], the authors propose a novel intrusion detection approach by applying kernel principal component analysis (KPCA) for intrusion feature extraction and followed by support vector machine (SVM) for classification. The results show a significantly high accuracy and false rate, 97.2% and 0.42%, respectively. In [3], the authors combine k-mean clustering on the basis of KNN classifier. The experimental results on NSL-KDD dataset show that this method greatly improve the performance of KNN classifier. In [4], some robust decision tree are developed to produce effective decision rules from the attacked data. Experimental results on intrusion dataset KDD Cup 1999 show that robust decision tree evaluates less false positive and true negative alarm rates compare to existing algorithms. These methods mentioned above belong to shallow learning and often emphasize feature engineering and selection. They have difficulty in features selection and cannot effectively solve the massive intrusion data classification problem, which leads to low recognition accuracy and high false alarm rate.

In recent years, deep learning has grown very fast and achieve good results in many scenarios,such as image classification and speech recognition. Hence, intrusion detection methods based on network traffics have been proposed successively. In [5], the authors propose a mal-ware traffic classification method based on convolutional neural network with traffic data as image. This method does not need manual design features, and directly takes the original traffic as the input data to the classifier. In [6], the authors propose a system-call language-modeling approach for designing anomaly-based host intrusion detection systems. Experiments on public benchmark datasets show the proposed ensemble method gave a better AUC value (0.928) with a large margin than that of the averaging ensemble method (0.890) and the voting ensemble method (0.859). In [7] the authors introduce a hybrid scheme that combines the advantages of deep belief network and support vector machine. They conduct experiments on NSL-KDD dataset and show that the overall accuracy offered by the employed approach is high. All the above methods adopt a single network for feature learning. For example, CNN converts continuous network traffic into images for processing, which is equivalent to treating traffics as independent and ignore the internal relations of network traffics. LSTM is good at dealing with serialization tasks but poor in the ability to extract features. Hence, we should adopt model fusion ideas to enhance the accuracy and generalization ability of single model.

ICCPR '19, October 23–25, 2019, Beijing, China

ACM ISBN 978-1-4503-7657-0/19/10…$15.00
DOI: https://doi.org/10.1145/3373509.3373539

In this paper, we address the problem of malicious traffics identification from a computer vision perspective. A deep learning model OCL that combines CNN [8] and LSTM [9] is proposed. The OCL model first uses one-dimensional convolution operation as a initial learning layer to automatically learn the traffic characteristics directly from the traffic raw data. Secondly, LSTM is used to combine the extracted features over time from the obtained features via convolution operation. The whole process of feature learning does not use any feature engineering skills. Up to now, we have finished the key features extraction of network traffics. Finally, softmax function is performed for network traffics classification. By testing our model on a public benchmark dataset, NSL-KDD dataset, the accuracy of OCL model can reach 81.25% and 63.75%, which is about 1.12% and 1.43% higher than the existing CNN model on the KDDTest+ and KDDTest-21 dataset, respectively.

The following are some of the key contributions and findings of our work:

1) We propose an end-to-end deep learning model OCL that is composed of CNN and LSTM, which realizes the automatic learning of the nonlinear relationship from original input to expected output.

2) We compare the performance of OCL model with other machine learning methods and deep learning methods. Experimental results show that the performance of OCL model is better than the traditional methods.

3) We evaluate our proposed model with a real NSL-KDD dataset, which can well solve the problem of intrusion detection and provide a new research method for intrusion detection.

The rest of the paper is organized as follows: In Section 2, we give a brief overview of the related work. In Section 3, we present details of the proposed application. In Section 4, we explain the experimental setup and present our results. Section 5 draws the conclusions.

2. RELATED WORKS

The intrusion detection technology based on machine learning method can be divided into two major categories: traditional machine learning methods and deep learning methods.

Traditional machine learning methods The traffic anomaly detection methods based on machine learning has achieved a lot of success. In [10], a novel support vector machine (SVM) model combining kernel principal component analysis (KPCA) with genetic algorithm (GA) is proposed for intrusion detection. Experimental results show that the proposed model performs higher predictive accuracy, faster convergence speed and better generalization. In [11], the authors propose a new intrusion detection system based on K-nearest neighbor (KNN) classification algorithm in wireless sensor network. The test results show that the system has high detection accuracy and speed, in accordance with the requirement of wireless sensor network intrusion detection. In [12], the kernel-based fuzzy-rough feature selection method is used to select the feature subset for the intrusion detection. Experimental outcomes obtained by applying the kernelbased fuzzy-rough feature selection method on KDD dataset demonstrate that it performs better than in terms of reduction effect and accuracy. In [13], the authors propose a new method of feature selection and classification based on support vector machine (SVM). Experimental results on NSL-KDD cup 99 of intrusion detection dataset show that the classification accuracy of this method with all training features reached 99%. As described above, machine learning methods have been proposed and have achieved success for an intrusion detection system. However, these methods require large-scale preprocessing and complex feature engineering of traffic data. It is impossible to solve massive intrusion data classification problem using machine learning methods.

Deep learning methods With the superior performance of deep learning in image recognition [14-16] and speech recognition [17-18], traffic anomaly detection methods based on deep learning have been proposed. In [19], the authors use Self-taught Learning (STL) on NSL-KDD dataset for network intrusion. Testing results show that their 5-class classification achieved an average f-score of 75.76%. In [20], the authors apply recurrent neural network with hessian-free optimization which is one of the deep learning algorithm for intrusion detection. Experiment result on the DARPA dataset and KDD Cup 1999 dataset demonstrate that our proposed model is superior to the existing studies through comparing the performance. In [21], the authors build a Deep Neural Network (DNN) model for an intrusion detection system and train the model with the NSL-KDD Dataset. Experimental results confirm that the deep learning approach shows strong potential to be used for flow-based anomaly detection in SDN environments. In [22], the authors propose an intrusion detection method using deep belief network (DBN) [23-24] and probabilistic neural network (PNN). The experiment result on the KDD CUP 1999 dataset shows that the method performs better than the traditional PNN, PCA-PNN and unoptimized DBN-PNN. In [25], the authors propose to use a typical deep learning method Convolution Neural Networks (CNN) for detecting cyber intrusions. 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.

3. PROPOSED WORK

The OCL model can automatically learn the key features of traffic data via hierarchical deep learning models, which can achieve efficient and accurate malicious traffic detection. The architecture of OCL model is shown in figure 1. At the data preprocessing layer, OCL model converts each data sample into a numerical form and then performs normalization operations. We convert the numerical data into traffic images. Then, we adpot convolutional neural network as a feature extractor that takes a image representation of the network traffic. Convolutional operation and pooling operation are followed by the relu activation function. This feature extractor convolves the input image in several steps and produces a feature map. Each time step represents the extracted frequency features, and then can be used as input to the LSTM. The LSTM layer can model the temporal information of the activation of the feature maps. Finally, the fused features are fed into a classifier to get the final recognition results.

3.1 Data Preprocessing Layer

There are three symbolic datatypes in NSL-KDD data features: protocol type, flag and service. We use one-hot encoder mapping these features into binary vectors.

One-hot processing. NSL-KDD dataset is processed by one-hot method to transform symbolic features into numerical features. For example, the second feature of the NSL-KDD data sample is protocol type. The protocol type has three values: tcp, udp, and icmp. One-hot method is processed into a binary code that can be recognized by a computer, where tcp is [1, 0, 0], udp is [0, 1, 0], and icmp is [0, 0, 1].

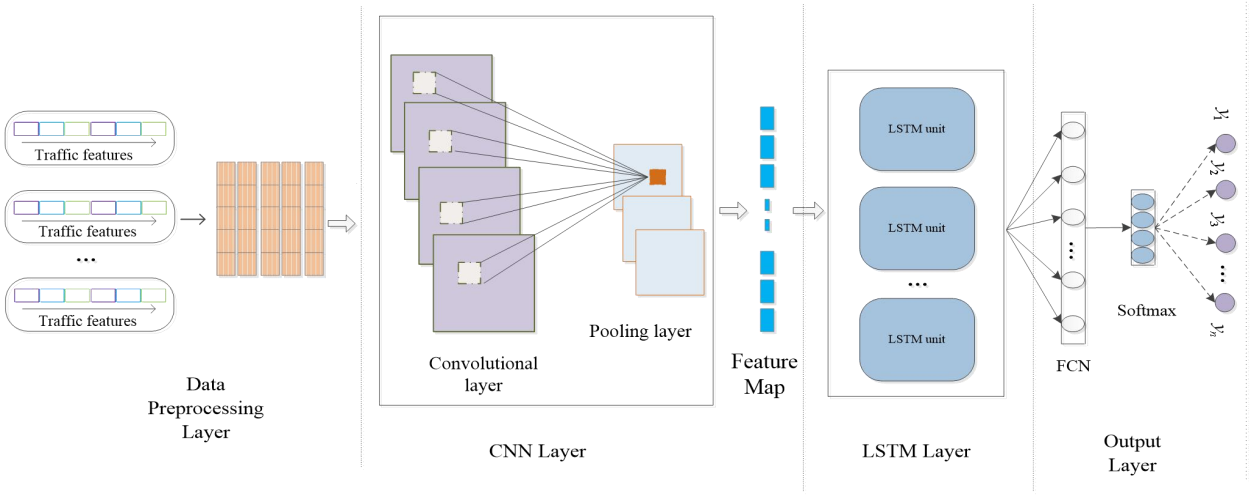

Figure 1. The architecture of OCL model. The whole architecture is divided into four parts. Convolutional neural network is used to extract features from the traffic images. LSTM can model the feature maps yielded by the upper layer. Output layer are made up of a fully connected layer and a softmax layer to classify various traffic.

Normalization processing. The value range of the original data may be too large, resulting in problems such as large numbers to eat decimals, data processing overflows and inconsistent weights so on. We use standard scaler to normalize continuous data into the range [0, 1]. Normalization processing eliminates the influence of the measurement unit on the model training, and makes the training result more dependent on the characteristics of the data itself. The formula is shown in equation (1).

$$r_{ij}^{''} = \frac{r_{ij}^{'} - r_{\min}}{r_{\max} - r_{\min}} \qquad r_{\max} = \max\left\{r_{ij}^{'}\right\} \tag{1}$$

where r stands for numeric feature value, r_{min} stands for the minimal value of the feature, r_{max} stands for the max value, r' stands for value after the normalization.

3.2 CNN Layer

To make our gathered data compatible with our intrusion detection system, we need to transfer the data into the image domain. Data samples are converted into an activity image with size of $41 \times 19 \times 1$. Hence, CNN model can be used to extract features from these images. Convolutional operation is the most critical of the CNN model. The formula is shown in equation (2).

$$x_{i,k}^{l,j} = \sigma\left(b_j + \sum_{a-1}^{m} w_a^j x_{l+a-1}^{l-1,j}\right), \tag{2}$$

Max pooling operation is selected to extract the most significant elements in each convolution and then they are turned into feature vectors. Hence, the max pooling operation is used for dimensionality reduction in this paper, and the calculation process is as shown in equation (3).

$$x_i^{l,j} = \max_{k=1}^{r}\left(x_{(i-1)\times s+k}^{l-1,j}\right), \tag{3}$$

where x is one of the i-th unit of j feature map of the k-th section in the l-th layer, and s is the range of the section. σ represents a nonlinear function, here the relu function is used.

3.3 LSTM Layer

After the network traffics are learned, the hidden layer can output feature maps yielded by the CNN layer. This feature map is sliced along the x-axis and each slice is used as a time step for the LSTM network. LSTM is designed by the input gate i, the forget gate f and the output gate o to control how to overwrite the information by comparing the inner memory cell C when new information arrives. When information enters a LSTM network, we can judge whether it is useful according to relevant rules. Only the information that meets algorithms authentication will be remained, and inconsistent information will be forgotten through forget gate. Given an input sequence $x = (x_0,...,x_t)$ at time t and the hidden states $h = (h_0,...,ht)$ of a LSTM layer can be derived as follows.

The forget gate will take the output of hidden layer h_{t-1} at the previous moment and the input x_t at the current moment as input to selectively forget in the cell state Ct, which can be expressed as:

$$f_t = sigmoid\ \left(W_{xf} x_t + W_{hf} h_{t-1} + b_f\right) \tag{4}$$

The input gate cooperates with a *tanh* function together to control the addition of new information. *tanh* generates a new candidate vector. The input gate generates a value for each item in Ct from 0 to 1 to control how much new information will be added, which can be expressed as:

$$C_t = sigmoid\ \left(f_i \cdot C_{t-1} + i_i \cdot \hat{C}_t\right) \tag{5}$$

$$i_t = sigmoid\ \left(W_{xi} x_t + W_{hi} h_{t-1} + b_i\right) \tag{6}$$

$$\hat{C}_t = \tanh\left(W_c x_t + W_c h_{t-1} + bc\right) \tag{7}$$

The output gate is used to control how much of the current unit state will be filtered out, which can be expressed as:

$$o_t = \sigma\left(W_{xo} x_t + W_{xo} h_{t-1} + b_o\right) \tag{8}$$

The output of the last LSTM unit is ht, which can be expressed as:

$$h_t = o_t \tanh(c_t) \tag{9}$$

3.4 Model Training

Training the proposed network contains a forward pass and a backward pass.

Forward Propagation The model is mainly composed of CNN model and LSTM model, each of which presents different structures and thus plays different role in the whole model. The objective function of our model is the cross-entropy based cost function. The goal of training the model is to minimize the cross entropy of the expected and actual outputs for all activities. *i* is the index of network traffic. *j* is traffic category, which can be given as:

$$C = -\sum_{i}\sum_{j} y_i^j \ln a_i^j + (1-y_i^j)\ln(1-a_i^j) \qquad (10)$$

Backward Propagation In this paper, we use the Back Propagation Through Time (BPTT) [26-27] algorithm to obtain the derivatives of the objective function with respect to all the weights, and minimize the objective function by stochastic gradient descent.

4. EVALUATION

In this section, we analyze the performance of the OCL model on the NSL-KDD dataset [28]. In order to verify the advancement and practicability of the OCL model, we compare the performance of this model with some state-of-the-art works.

4.1 Benchmark Datasets

The final result of network traffic anomaly detection is closely related to the dataset. NSL-KDD dataset is an enhanced version of KDD cup 1999 dataset, which is widely used in intrusion detection experiments. NSL-KDD dataset not only effectively solves the inherent redundant records problems of the KDD Cup 1999 dataset but also makes the number of records reasonable in the training set and testing set. The NSL-KDD dataset is mainly composed of KDDTrain+ training dataset, KDDTest+ and KDDTest-21 testing dataset. These data sets have different normal records and four different types of abnormal records, as shown in Table 1. The KDDTest-21 dataset is a subset of the KDDTest+ and is more difficult for classification.

Table 1. different classification in the NSL-KDD dataset

	Total	Normal	Dos	R2L	U2R	Probe
KDDTrain+	125973	67343	45927	995	52	11656
KDDTest+	22544	9711	7458	2421	2754	2421
KDDTest-21	11850	2152	4342	2754	200	2402

Network traffic is generally collected at fixed time intervals. Essentially, network traffic data is a kind of time series data. There are 41 features and 1 class label for every traffic record [29]. These features include basic features (1-10), content features (11-22) and traffic features (23-41). According to its characteristics, there are four types of attacks in this dataset: DoS (Denial of Service attacks), R2L (Root to Local attacks), U2R (User to Root attack), and Probe (Probing attacks).

4.2 Experimental Settings

In this paper, the NSL-KDD dataset is used to test the performance of OCL model. The data samples of the NSL-KDD dataset are divided into two parts: one is used to build a classifier, that is called the training dataset. The other is used to evaluate the classifier, that is called the testing dataset. There are 125,973 records in the training set and 22,543 records in the testing set. Table 2 shows the distribution of training and testing records for the (normal/attack) type of network traffic.

Table 2. Distribution of training and testing records

	Normal	DoS	Probe	U2R	R2L	Total
Train	67,343	45,927	11,656	52	995	125,973
Test	9,711	7,458	2,421	200	2,754	22,543

The operating environment of all experiments isKeras with tensorflow as the backend; Operating system is 64-bit CtOS7; Processor is E5-2620 v4; Main frequency is 2.10GHz; Memory is 32.0G; Python version is 3.6. The parameter list of OCL model is shown in Table 3.

Table 3. Super parameters of the OCL model

parameters	values
CONV KSIZE	40
CONV NUM	5
Pool KSIZE	3
Lstm hidden units	40
Dense 1	64
Dense 2	10
Loss function	categorical_crossentropy
optimizer	adam
batch_size	256
lr	0.001

4.3 Performance Analysis of the Model

In this paper, accuracy is used to evaluate the OCL model. Accuracy represents the proportion of correctly classified samples to the total number of samples. Figure 2 illustrates the accuracy of the OCL model on the KDDTrain+、KDDTest+ and KDDTest-21 dataset. After careful fine-tuning, the accuracy of the OCL model on both the training set and testing set shows an overall upward trend. Experiments on the KDDTest+ dataset show that when epoch = 100, the OCL model has a good accuracy (82.56%). At the same time, the accuracy of the OCL model on the KDDTest-21 dataset is 67.55% and the accuracy on the KDDTrain+ data set is 99.21%. In addition, as the number of iterations increases, the amplitude of the OCL model nearly remain steady when iteration reaches to 10 epoches on the KDDTrain+ and KDDTest+ dataset. In contrast, the amplitude of the OCL model on the KDDTest-21 dataset fluctuate lager than others and it needs more epoches to be steady.

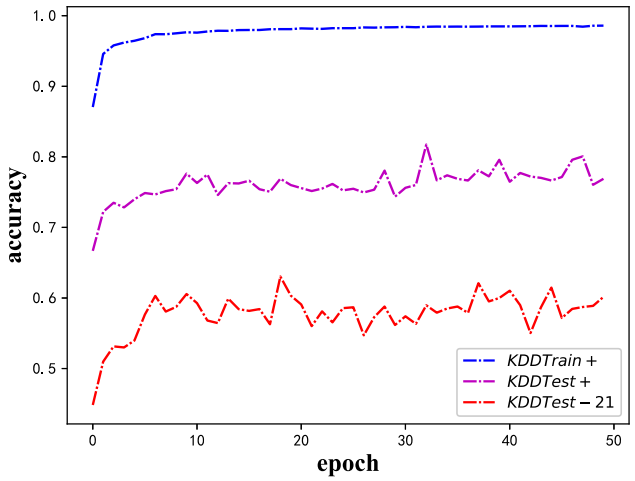

Figure 2. Accuracy of the OCL model on the NSL-KDD dataset.

The confusion matrix generated by the OCL model on the KDDTest+ dataset is shown in Figure 3. The experimental results show that most samples are concentrated on the diagonal of the confusion matrix, indicating that the overall classification performance is very high. However, it can be intuitively seen from the confusion matrix in Figure 3 show that the OCL model achieves good detection performance in distinguishing normal traffics. Dos attack traffics and Probe attack traffics, but there is still further improvement in distinguishing different attack traffics.

Figure 3. Confusion matrix yielded by the OCL model on the NSL-KDD dataset.

4.4 Comparison to the State of the Art

In order to objectively evaluate the accuracy and differentiation of the OCL model, we compare our network with some related works proposed by[30] [31] [32]. According to the results shown in Table 4, our proposed method has the relatively good accuracy compared to other methods. In [30], the authors have shown the results obtained by J48, Naive Bayesian, Random Forest, Multi-layer Perceptron, Support Vector Machine and the other classification algorithms on the NSL-KDD dataset. In [31], the authors build a Deep Neural Network (DNN) model for an intrusion detection system and train the model with the NSL-KDD Dataset. Experimental results confirm that the deep learning approach shows strong potential to be used for flow-based anomaly detection in SDN environments. In [32], the authors propose to use a typical deep learning method Convolution Neural Networks (CNN) for detecting cyber intrusions. The experimental results show that the performance of CNN model is superior to the performance of models based on traditional machine learning methods and novel deep learning methods in multi-class classification. According to the results shown in Table 4, our proposed method has the relatively good accuracy compared to other method.

Table 4. Comparison of different method

model	KDDTest+	KDDTest-21
J48[30]	81.05%	63.97%
Naive bayes[30]	74.40%	55.77%
NB Tree[30]	75.40%	55.40%
Random forest[30]	74.00%	50.80%
Random tree[30]	72.80%	49.70%
MLP[30]	78.1%	58.4%
SVM[30]	74.00%	50.70%
DNN[31]	75.75%	-
DBN[32]	71.91%	46.73%
LSTM[32]	73.18%	49.38%
CNN[32]	80.13%	62.32%
CNN	78.47%	60.03%
OCL	81.25%	63.75%

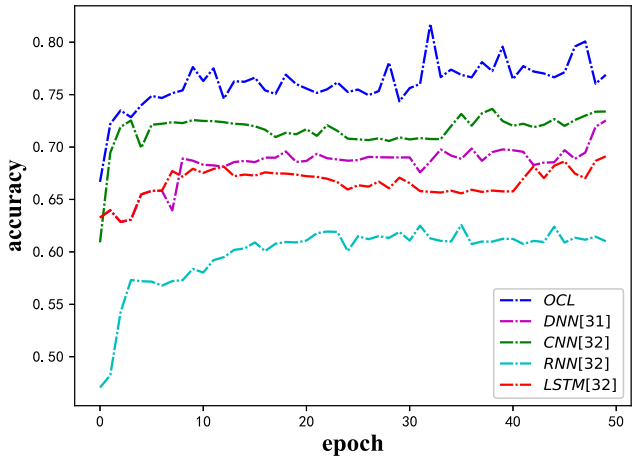

Figure 4. Comparison of Accuracy with different models.

As shown in Table 4, we can observe that the OCL model performs better than other models in terms of accuracy, which can reach 81.25%, 63.05% in the KDDTest+ and KDDTest-21 dataset. Compared with the model of [31], the OCL model directly takes the collected traffic as original input, which can automatically extract features by means of end-to-end learning. The OCL model

achieves better classification results than manual design methods. Meanwhile, we compare the recent work of using the deep learning model for abnormal traffic detection. As can be seen from Table 4, the OCL model achieved the best results on both the KDDTest+ and KDDTest-21 dataset. On the KDDTest+ set, the accuracy of the OCL model is 1.12% higher than CNN [32]. On the KDDTest-21 set, the accuracy of the OCL model is 1.43% higher than CNN [32]. The OCL model is more accurate than CNN because OCL combines LSTM to model feature maps over the time, which can obtain more contextual information. These results prove that the OCL model can offer a significant advantage across very different scenarios. In addition, it can be seen from Figure 4 that the OCL model has the best accuracy among all the deep learning models.

5. CONCLUSION

In this paper, we have evaluated the OCL model for intrusion detection using NSL-KDD dataset. The OCL model is combination of Convolutional neural network (CNN) and Long Short-Term Memory (LSTM). CNN is used to extract features from the traffic images. LSTM can model feature maps yielded by the CNN layer. This method does not need to manually extract and select features of network traffic, but directly takes the original traffic as the input data of deep neural network. The overall process of network traffic learning is completed by the deep neural network, which can save a lot of feature engineering workload and reduce complexity of this task. Experimental results on the NSL-KDD dataset indicate that the OCL model achieves pretty high accuracy. On the KDDTest+ and KDDTest-21 dataset, the accuracy of the OCL model is 1.12% and 1.43% higher than the standard CNN model, respectively. Hence, we believe that the proposed method can be used as a powerful tool for intrusion detection problems.

6. ACKNOWLEDGMENTS

This work is supported in part by the Natural Science Foundation of Tianjin under Grants (17JCYBJC16400, 18JCYBJC85900, 18JCQNJC70200).

We consider both adjectives and verbs that can express emotions and feelings as sentiment words in this research.

7. REFERENCES

[1] Rieck K, Laskov P . Language models for detection of unknown attacks in network traffic[J]. Journal in Computer Virology, 2007, 2(4):243-256.

[2] Gao H H, Yang H H , Wang X Y . Kernel PCA Based Network Intrusion Feature Extraction and Detection Using SVM[J]. Lecture Notes in Computer Science, 2005:89-94.

[3] H. Shapoorifard and P. Shamsinejad, “Intrusion detection using a novel hybrid method incorporating an improved knn,”International Journal of Computer Applications, vol. 173, no. 1, pp. 5-9, 2017.

[4] Kumar G S . Real Time and Offline Network Intrusion Detection using Improved Decision Tree Algorithm[J]. International Journal of Computer Applications, 2012, 48(25):1-6.

[5] Wang W , Zhu M , Zeng X , et al. Malware traffic classification using convolutional neural network for representation learning[C]// 2017 International Conference on Information Networking (ICOIN). IEEE, 2017.

[6] Kim G , Yi H , Lee J , et al. LSTM-Based System-Call Language Modeling and Robust Ensemble Method for Designing Host-Based Intrusion Detection Systems[J]. 2016.

[7] Salama M A , Eid H F , Ramadan R A , et al. Hybrid Intelligent Intrusion Detection Scheme[M]// Soft Computing in Industrial Applications. Springer Berlin Heidelberg, 2011.

[8] Poria S, Cambria, Erik, Gelbukh, Alexander. Aspect Extraction for Opinion Miningwith a Deep Convolutional Neural Network[J]. Knowledge-Based Systems, 2016, 108:42-49.

[9] Su Tongtong, Sun Huazhi, Ma Chunmei, et al. Research on Human Behavior Recognition Based on Recurrent Neural Network[J]. Journal of Tianjin Normal University (Natural Science Edition), 2018, 38(06):61-65+79.

[10] Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.

[11] Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[12] Zhang Q, Qu Y, Deng A. Network Intrusion Detection Using Kernel-based Fuzzy-rough Feature Selection[C]// 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE, 2018.

[13] Pervez M S , Farid D M . Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs[C]// International Conference on Software. IEEE, 2015.

[14] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2016.

[15] Dong C, Loy C C, He K, et al. Image Super-Resolution Using Deep Convolutional Networks[J]. IEEE Trans Pattern Anal Mach Intell, 2016, 38(2):295-307.

[16] Shin H C, Roth, Holger R, Gao, Mingchen, et al. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning[J]. IEEE Transactions on Medical Imaging, 2016, 35(5):1285-1298.

[17] Noda K, Yamaguchi Y, Nakadai K, et al. Audio-visual speech recognition using deep learning[J]. Applied Intelligence, 2015, 42(4):722-737.

[18] Hinton G, Deng, Li, Yu, Dong, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups[J]. IEEE Signal Processing Magazine, 2012, 29(6):82-97.

[19] Javaid, Ahmad Yazdan , et al. "A Deep Learning Approach for Network Intrusion Detection System." 9th EAI International Conference on Bio-inspired Information and Communications Technologies ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015.

[20] Kim J , Kim H . Applying Recurrent Neural Network to Intrusion Detection with Hessian Free Optimization[C]// International Workshop on Information Security Applications. Springer International Publishing, 2016.

[21] Tang T A , Mhamdi L , Mclernon D , et al. Deep Learning Approach for Network Intrusion Detection in Software

Defined Networking[C]// The International Conference on Wireless Networks and Mobile Communications (WINCOM'16). IEEE, 2016.

[22] Zhao G , Zhang C , Zheng L . Intrusion Detection Using Deep Belief Network and Probabilistic Neural Network[C]// 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC). IEEE, 2017.

[23] Gao N , Gao L , Gao Q , et al. An Intrusion Detection Model Based on Deep Belief Networks[C]// 2014 Second International Conference on Advanced Cloud and Big Data (CBD). IEEE Computer Society, 2014.

[24] Alom M Z , Bontupalli V R , Taha T M . Intrusion detection using deep belief networks[C]// NAECON 2015 - IEEE National Aerospace and Electronics Conference. IEEE, 2015.

[25] Y. Ding and Y. Zhai, "Intrusion detection system for nsl-kdd dataset using convolutional neural networks," in Proceedings of the 2018 2Nd International Conference on Computer Science and Artificial Intelligence, 2018, pp. 81-85.

[26] Graves A . Supervised Sequence Labelling with Recurrent Neural Networks[J]. Studies in Computational Intelligence, 2008, 385.

[27] Zeng M , Nguyen L T , Yu B , et al. Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors[C]// Sixth International Conference on Mobile Computing, Applications and Services (MobiCASE 2014). IEEE, 2014.

[28] Harbola A, Harbola J, Vaisla K S. Improved Intrusion Detection in DDoS Applying Feature Selection Using Rank & Score of Attributes in KDD-99 Data Set[C]// International Conference on Computational Intelligence & Communication Networks. 2014.

[29] L. Dhanabal and S. P. Shantharajah, "A study on nsl-kdd dataset for intrusion detection system based on classification algorithms," 2015.

[30] Tavallaee M , Bagheri E , Lu W , et al. A detailed analysis of the KDD CUP 99 data set[C]// IEEE International Conference on Computational Intelligence for Security & Defense Applications. IEEE, 2009.

[31] Tang T A , Mhamdi L , Mclernon D , et al. Deep Learning Approach for Network Intrusion Detection in Software Defined Networking[C]// The International Conference on Wireless Networks and Mobile Communications (WINCOM'16). IEEE, 2016.

[32] Y. Ding and Y. Zhai, "Intrusion detection system for nsl-kdd dataset using convolutional neural networks," in Proceedings of the 2018 2Nd International Conference on Computer Science and Artificial Intelligence, 2018, pp. 81-85.