2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing (HPSC), and IEEE Intl Conference on Intelligent Data and Security (IDS)

A Novel Intrusion Detector Based on Deep Learning Hybrid Methods

Wang Shizhao
Beijing Key Laboratory of Network Technology
Beihang University
Beijing, China
wangsz@buaa.edu.cn

Xia Chunhe
Beijing Key Laboratory of Network Technology
Beihang University
Beijing, China
xch@buaa.edu.cn

Wang Tianbo*
School of Cyber Science and Technology
Beihang University
Beijing, China
wangtb@buaa.edu.cn

***Abstract*—Intrusion detection system plays an important role in network security defense. It analyzes network traffic and connection characteristics to identify various types of network attacks. Deep learning based intrusion detectors perform better in predicting unknow attacks and detection accuracy. In this paper, we use long short-time memory (LSTM) in recurrent neural network (RNN) units, and propose an improved long short-time memory tree (LSTMTree) model with ability of secondary detection to solve the problem of high false negative rate in the RNN intrusion detector. And then we use NSL-KDD data set to verify the performance of our presented model. The experimental results show that our model can improve the detection performance better than previous models.**

Keywords—Intrusion detector, deep learning, Long short-time memory, Decision tree, Secondary Detection

I. INTRODUCTION

With the rapid development of internet technology, network attacks are changing constantly, and the cyber threats we face are getting serious in both traditional networks and new networks such as smart grid networks[1]. Therefore, the cyber defense measures are particularly important. Traditional static defenses lack the capabilities of analysis, and can't fully identify multiple attacks. For these reasons, the intrusion detection systems (IDS) with active defense come into being. IDS can detect whether an intrusion has occurred by analyzing the real-time network packets and generates an alarm. Intrusion detection also plays an important role in the Fifth Generation (5G) context [2]. Thus IDS protects the network from attacks, and is of great importance to network security.

Original intrusion detection systems require manual design rules to detect intrusion behaviors. Due to the rules design lagging behind the appearance of attacks, it is hard to recognize unknow attacks. The essence of intrusion detection is the data classification which is exactly a key application of machine learning. Machine learning technology uses numerous of data to train, so that the model can learn the details of the data and improve the accuracy of intrusion detection. At the same time, machine learning technology has a good generalization ability which can enable intrusion detection system to predict unknown attacks. It is a significant characteristic for intrusion detection, which can reduce the false negative rate and false positive rate.

In order to obtain a better detection performance, shallow classifiers based on machine learning methods need feature extraction and selection to reduce redundant features [3]. Compared with the methods above, deep learning technology has the ability to automatically extract features, and have a better performance in classification through big data analysis. In recent years, it has been widely used to intrusion detection field [4-6]. Recurrent neural network (RNN) can capture time series features of data and discover temporal correlation by memory past information [7]. It has excellent effects in the fields of natural language processing (NLP), speech recognition, etc. LSTM as a variant of RNN has better performance in handling long-term dependency issues [8]. Because some features such as traffic and login in intrusion data are related to timing, LSTM is suitable for intrusion detection.

We propose a LSTMTree model with ability of secondary detection. The model consists of two parts, LSTM and Decision Tree. Firstly, we use the LSTM to detect the data and separate the normal samples in test results. Secondly, we put these negative samples into a Decision Tree for a secondary detection. Ultimately, the final detection results can be obtained. In this way, the detection performance of intrusion detectors can be improved obviously. The experimental results show that our model has higher accuracy and lower false negative rate than the LSTM model. At the same time, compared with traditional machine learning methods (i.e. NB, SVM, RF), our model have a better detection accuracy.

The rest of this paper is organized as follows.

- Section 2 reviews the related works in deep learning based intrusion detection method.
- Section 3 introduces the basic theories of the methods we use and describes the details of our approach.
- Section 4 evaluates the performance of our approach by experiments and compares with other IDS methods.
- Section 5 concludes and proposes the future work.

II. RELATED WORK

Recently, numerous machine learning approaches have been used in intrusion detection. Such as Naive Bayes (NB) [9, 10] and Support Vector Machine (SVM) [11-13]. Decision Tree (DT) [14, 15] is also a typical method which can generates rules automatically and random forest is a combination of tree predictors which can prevent overfitting in intrusion detection [16].

Deep learning can handle big data efficiently, and has the ability to extract the representative characteristics from raw data. It provides a new direction for intrusion detection. With the great progress of deep learning, many prior studies have applied it into network security fields especially in intrusion detection. In[5], a deep neural network (DNN) with 3 hidden

Corresponding author: Wang Tianbo, E-mail: wangtb@buaa.edu.cn.

978-1-7281-0006-7/19/$31.00 ©2019 IEEE
DOI 10.1109/BigDataSecurity-HPSC-IDS.2019.00062

layers is used. The author compared the accuracy in NSL-KDD test set with learn rate in 0.1,0.01,0.001, 0.0001, and the best result is 75.75% with a 0.001 learn rate. Because the classification ability of simple DNN is limited, the model doesn't get a high accuracy. A Self-taught Learning (STL) method combining sparse auto-encoder and soft-max regression is proposed in [17], it can collect features representation from unlabeled data by Auto-encoder, and the accuracy on NSL-KDD test set is 79.10%. However, it does not conduct in-depth research on detection algorithms. In [18], authors use Text-CNN to extract effective information from original payloads and combine them, then execute classification by random forest, and it has a superior performance in detection rate and false alarm rate. But the false negative rate, an important indicator, is not discussed in the paper. Authors in [19] applies the recurrent neural network (RNN) into intrusion detection. The results of experiments show that the model with 1 layer and 80 hidden nodes gets a higher accuracy 81.29% of five categories classification on NSL-KDD test set. And the results also obtain better compared with other machine learning methods. But from the confusion matrix given in the paper, we find that the false normal prediction is very large which will lead to a high false negative rate. Based on the above analysis, we concentrate on the classification algorithm to improve the detection capabilities.

In this paper, we propose a LSTMTree model with the ability of secondary detection to improve the detection performance. We use LSTM for the first detection, and then do a secondary detection on normal predicted samples using decision tree to cut down the false negative rate. Finally, we compare our method with other typical approaches such as NB, SVM, RF, RNN and LSTM.

III. METHODOLOGY

A. Long Short-Time Memory

Recurrent neural network is a one-way propagation network with the ability of memory. It can consider both the current input and the hidden layer state. Thus, it is suitable for tasks related to chronological order as shown in Fig. 1.

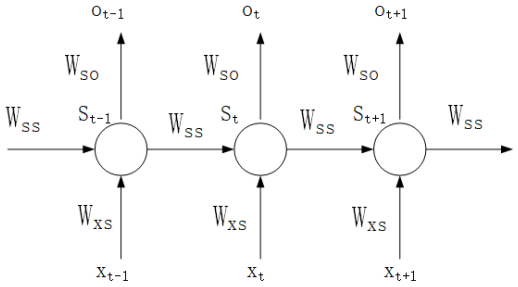

Fig. 1. expanded structure of RNN

Suppose that x is the current input, s is the current state, and o is the current output, t is the time, w is the weight. The calculation formula can be expressed as follows:

$$o_t = g(w_{so}s_t) \quad (1)$$

$$s_t = f(w_{sx} + w_{ss}s_{t-1}) \quad (2)$$

where $g()$and $f()$ are the activation function.

RNN has the problem of exploding and vanishing in backpropagation. Long short-time memory (LSTM) as a novel recurrent network can optimize RNN [20] by memory the long-term information as shown in Fig. 2.

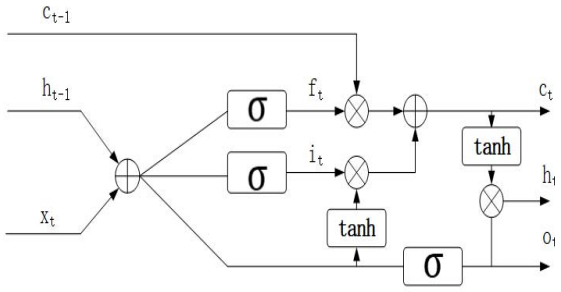

Fig. 2. structure of LSTM cell

We can describe the mechanism of LSTM by the following formulas:

$$o_t = \sigma(w_{xo}x_t + w_{ho}h_{t-1} + b_o) \quad (3)$$

$$h_t = o_t\tanh(c_t) \quad (4)$$

$$c_t = f_t c_t + i_t\tanh(w_{xs}x_t + w_{hc}h_{t-1} + b_c) \quad (5)$$

$$f_t = \sigma(w_{xf}x_t + w_{hf}h_{t-1} + b_f) \quad (6)$$

$$i_t = \sigma(w_{hi}h_{t-1} + w_{xi}x_t + b_i) \quad (7)$$

where $x_t$and o_t represent the input and output respectively, σ is the sigmoid function, different w are different connection weights. The equation i_t is the input gate which controls the input information ratio. The equation f_t is the forget gate which determines how much previous information will be abandoned. The equation $o_t$is the output gate which decides what will be outputted. These gates with information control ability can help to solve the exploding and vanishing problems in RNN [21].

B. Decision Tree

Decision tree is a classification algorithm with top-down decision rules [22]. A node with the largest information entropy value is selected in each time, then it will be split into child nodes. The evaluation algorithm of information entropy includes ID3, C4.5 and CART. Gradually a tree with the fastest decline in entropy can be constructed and the entropy of the leaf nodes is zero. It is shown in Fig. 3, where $[f_1, f_2, f_3]$ are different features, and $[c_1, c_2, c_3, c_4]$ are different categories. The finish tree can be used to classify, so it also suitable for intrusion detection.

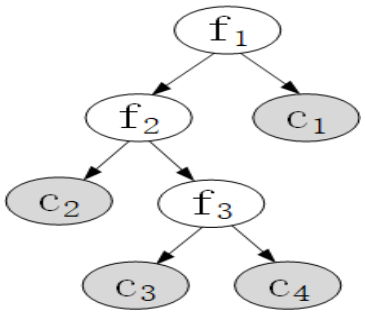

Fig. 3. Decision Tree

C. LSTMTree Model

In train stage, we first preprocess the NSL-KDD dataset by digitalization and normalization. It is shown in Fig. 4 The specific method will be introduced in the next section. Secondly, we use the preprocessed data to train the LSTM whose weight is initialized by Xavier method to make the neural networks learn better. We use softmax-cross-entropy as the loss function and Adam optimizer to update the network weights. Then for the decision tree, we choose CART as the evaluation algorithm to select nodes, and use the real classification of the samples detected as '*normal'* by LSTM to train it. In test stage, we use the LSTM for the first detection, and separate the output samples predicted as normal. Then we feed these '*normal'* samples into the trained decision tree for a secondary detection. Finally, we update the results of LSTM using the results of decision tree. The overall training process of the model is shown in Fig. 5.

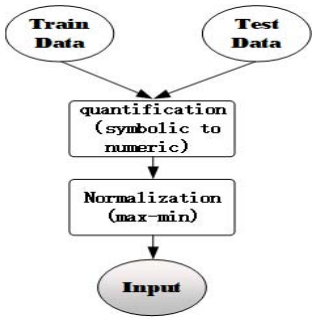

Fig. 4. data preprocessing

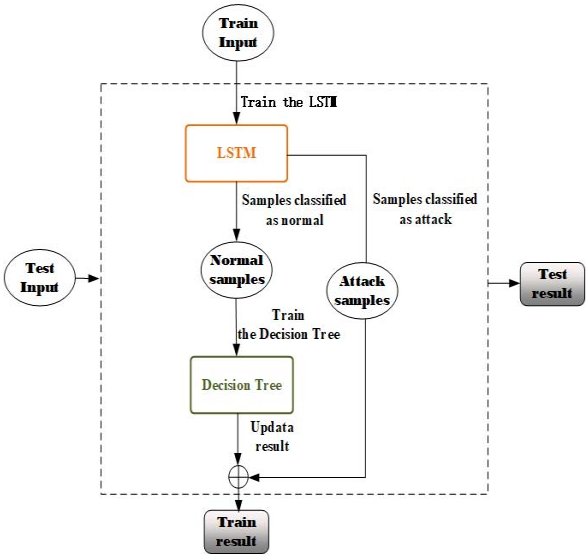

Fig. 5. overall training process of the mode

IV. EXPERIMENTS AND DISCUSSION

A. Dataset

1) Introduction

In this paper, we use the benchmark intrusion detection dataset NSL-KDD which solves the issues of redundancy in another intrusion detection dataset KDD CUP99[23]. There are 125973 records in train set and 22544 records in test set, and some records in the test set are not in the train set. 41 features are in each record which can be divided into 2 categories: continues features and discrete features [24]. The name and type is shown in TABLE I.

TABLE I. FEATURES OF NSL-KDD DATASET.

	Name	Type		Name	Type
1	duration	numerical	3	count	numerical
	protocol_type	discrete		srv_count	numerical
	service	discrete		serror_rate	numerical
	src_bytes	numerical		srv_serror_rate	numerical
	dst_bytes	numerical		rerror_rate	numerical
	flag	discrete		srv_rerror_rate	numerical
	land	discrete		same_srv_rate	numerical
	wrong_fragment	numerical		diff_srv_rate	numerical
	urgent	numerical		srv_diff_host_rate	numerical
2	hot	numerical	4	dst_host_count	numerical
	num_failed_logins	numerical		dst_host_srv_count	numerical
	logged_in	discrete		dst_host_same_srv_rate	numerical
	num_compromised	numerical		dst_host_diff_srv_rate	numerical
	root_shell	discrete		dst_host_same_src_port_rate	numerical
	su_attempted	discrete		dst_host_srv_diff_host_rate	numerical
	num_root	numerical		dst_host_serror_rate	numerical
	num_file_creations	numerical		dst_host_srv_serror_rate	numerical
	num_shells	numerical		dst_host_rerror_rate	numerical
	num_access_files	numerical		dst_host_srv_rerror_rate	numerical
	num_outbound_cmds	numerical			
	is_hot_login	discrete			
	is_guest_login	discrete			

There are 41 features divided in 4 categories. Category 1 are the basic characteristics of the TCP connection containing some basic properties of the connection. Category 2 are characteristics of TCP connections reflecting intrusion behavior. Category 3 are time-based network traffic statistics characteristics. Category 4 are host-based network traffic statistics characteristics.

There are one normal and 4 attacks in the dataset. The types of attacks include denial of service (DoS), probing (Probe), user to root (U2R), root to local (R2L). The number of different categories in the data set is shown in TABLE II

TABLE II. DIFFERENT CATEGORIES IN NSL-KDD

	Normal	DoS	Probe	U2R	R2L
NSL-KDD Train	67343	11656	45927	52	995
NSL-KDD Test	9711	7458	2421	200	2754

In these features, the protocol_type, service and flag are character, and others are numeric, so we should digitize the character ones to adapt the input of LSTM. The specific method will be introduced next

2) Data preprocessin

For the character features including protocol_type, service and flag in NSL-KDD dataset, we using the numbers to digitalize. The 3 types of protocol_type are replaced by integers from 0 to 2, the 70 types of service are replaced by integers from 0 to 69, and the 11 kinds of flags are changed to integers from 0 to 10. We use the 5 labels by integers from 0 to 4.

Because the range of some features is very large, we use max-min method to normalization the original features. The formula is shown as (8)

$$x' = \frac{x-min}{max-min} \tag{8}$$

where min represents the minimum value for feature x, max is the maximum value for feature x. After processing, the value of each feature will be in range [0,1].

B. *Evaluation Metrics*

In this paper, we mainly focus on three key evaluation indicators, namely accuracy (Acc), false negative rate (FNR) and false positive rate (FPR), where TP is the true positive, TN is the true negative, FP is the false positive and FN is the false negative. The accuracy is how many of the samples are correctly classified. The formula is shown as (9).

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

The false negative rate is the ratio of samples that are misidentified as negative. The formula is shown as (10).

$$FNR = \frac{FN}{TP+FN} \tag{10}$$

And the false positive rate is the ratio of samples that are misidentified as positive. The formula is shown as (11).

$$FPR = \frac{FP}{TN+FP} \tag{11}$$

The confusion matrix is shown in TABLE III.

TABLE III. CONFUSION MATRIX

Predicted class / Actual class	Attacks	Normal
Attacks	TP	FN
Normal	FP	TN

V. EXPERIMENTAL RESULTS

For the experiments, we use the Dell personal laptop with Intel Core i7-7500U CUP, 12GB memory and Window 10 operating system. And we choose tensorflow framework of python to build the LSTM model and use Scikit-learn library for the decision tree classifier.

Different parameters of LSTM have been set to explore the best performance in experiments. We give 30 epochs and set batch size to be 500, and make the value of learning rate (LR) vary in range {0.05,0.01,0.001}, number of layers (L) vary in range {1,2,3}, number of nodes (N) in each hidden layer vary in range {64,128,256}. We do the five-categories classification which are normal, DoS, Probe, U2R and R2L. The best test accuracy (acc) and corresponding train epoch with different parameters on NSL-KDD are shown in TABLE IV (We record the best test accuracy and corresponding epoch).

From the results shown in above table, it is obviously that the best accuracy obtains when learning rate is 0.01, number of layers is 2, nodes in hidden layer is 128, so we choose this set of parameters in our model. We also calculate the FNR, FPR for the best parameters, they are 0.267 and 0.0302 respectively.

TABLE IV. ACCURACY WITH DIFFERENT PARAMETERS IN LSTM

parameters			results		
LR	**L**	**N**	**train acc**	**test acc**	**epoch**
0.05	1	64	0.984	0.759	3
0.05	1	128	0.974	0.751	3
0.05	1	256	0.989	0.774	7
0.05	2	64	0.985	0.755	5
0.05	2	128	0.992	0.751	9
0.05	2	256	0.998	0.731	10
0.05	3	64	0.992	0.750	7
0.05	3	128	0.943	0.738	3
0.05	3	256	0.988	0.730	11
0.01	1	64	1.000	0.794	11
0.01	1	128	0.990	0.800	4
0.01	1	256	0.996	0.801	4
0.01	2	64	0.998	0.803	7
0.01	2	128	**0.994**	**0.820**	**5**
0.01	2	256	0.989	0.804	2
0.01	3	64	0.987	0.795	6
0.01	3	128	0.997	0.804	9
0.01	3	256	0.990	0.803	2
0.001	1	64	1.000	0.787	16
0.001	1	128	0.998	0.779	6
0.001	1	256	0.973	0.774	2
0.001	2	64	0.982	0.762	6
0.001	2	128	0.997	0.753	12
0.001	2	256	0.989	0.748	8
0.001	3	64	0.997	0.730	12
0.001	3	128	0.959	0.721	7
0.001	3	256	0.956	0.742	2

In decision tree, we use DecisionTreeClassifier in Scikit-learn with CART algorithm and choose min_samples_split=20, min_samples_leaf=5, max depth=200 by empiric. We change the class_weight to get a better result. The comparation of final accuracy (acc), FNR, FPR in different class_weight are shown in TABLE V.

TABLE V. RUSULTS OF LSTMTREE IN DIFFERENT CLASS_WEIGHT

parameters	Test acc	FNR	FPR
class_weight=[{0:1, 1:1},{0:1, 1:1},{0:1, 1:1},{0:1, 1:70},{0:1, 1:70}]	0.871	0.121	0.032
class_weight=[{0:1, 1:1},{0:1, 1:1},{0:1, 1:1}, {0:1,1:80},{0:1, 1:80}]	**0.873**	**0.120**	**0.031**
class_weight=[{0:1, 1:1},{0:1, 1:1},{0:1, 1:1},{0:1, 1:90},{0:1, 1:90}]	0.872	0.120	0.032

We can see that when the class_weight in decision tree is [{0:1, 1:1},{0:1, 1:1},{0:1, 1:1}, {0:1,1:80},{0:1, 1:80}], the results is best where accuracy rises from 0.820 to 0.873, FNR drops from 0.260 to 0.120, and FPR just slightly rises from 0.030 to 0.031 compared to LSTM. The false negative rated drops to half of the original, the false positives just rise by 0.001. However, it is more harmful that the attacks can't be detected.

We compare our model with LSTM and RNN in Fig. 6, and it has a better performance than other two.

From the figure, we know that both the original LSTM and the RNN model have a high false negative rate which affects the performance of the intrusion detector seriously. This problem can be solved well by retesting the normal samples in

the first detection and hardly affects the FPR. Thus our model is suitable for intrusion detection.

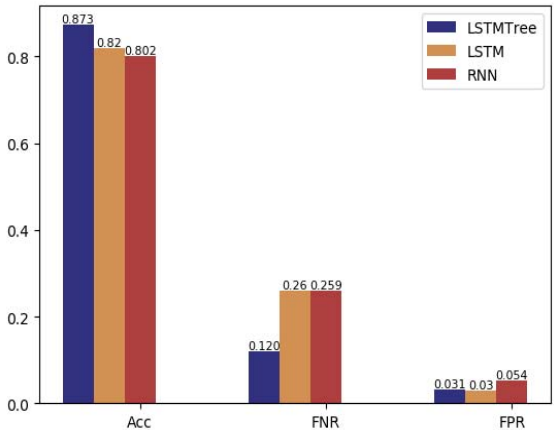

Fig. 6. Comparation of RNN, LSTM and LSTMTree

The confusion matrix for 5-classes and 2-classes of the best results are shown as Fig. 7 and Fig. 8.

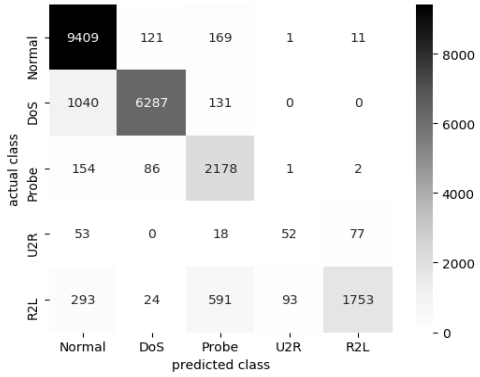

Fig. 7. Confusion matrix for 5-classes of our mode

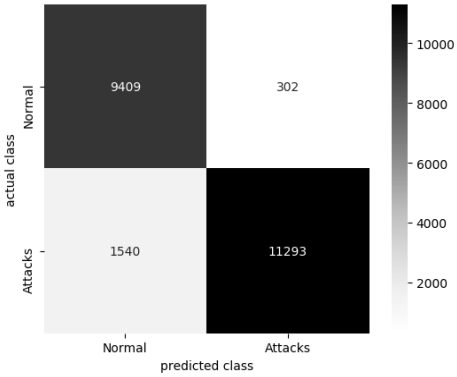

Fig. 8. Confusion matrix for 2-classes of our mode

We also compare the accuracy in 5-classification of our model on NSL-KDD test set with other methods including Naïve Bayes (NB), support vector machine (SVM), Random Forest (RF), RNN, LSTM as shown in Fig. 9. From the figure, we can see that the typical machine learning models such as NB, SVM, RF can't predict the attacks well, the accuracy of NB and SVM is just 0.569 which is almost close to the random guess. The accuracy of another two deep models RNN and LSTM is higher than machine learning models, which reaches above 0.8. But they are still lower than our method whose accuracy is 0.873.

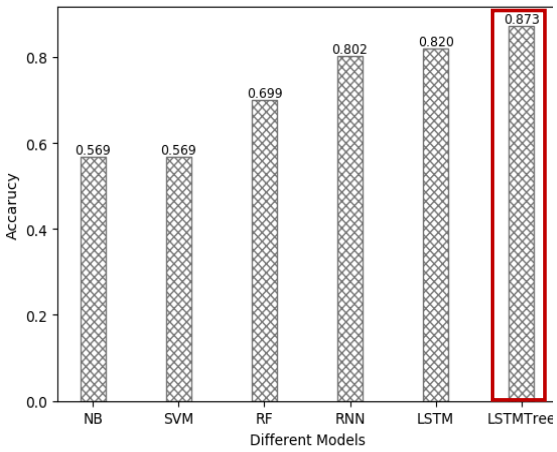

Fig. 9. Accuracy comparation of different model

VI. Conclusion and Future Works

In this paper, a LSTMTree model with secondary detection for normal predicted output of LSTM is proposed for IDS. We compare the accuracy, FNR, FPR with RNN and LSTM, and our method has better results, especially it can reduce the false negative rate of LSTM. But we find that the number of U2R samples is very small, and the detected samples of it is low, so in the future, we will concentrate on the unbalance intrusion data set, and get a better detection results by adjust proportion of each type in the dataset.

Acknowledgment

This work is supported by National Natural Science Foundation of China (U1636208, F020605).

References

[1] K. Gai, et al., Spoofing-jamming attack strategy using optimal power distributions in wireless smart grid networks, IEEE T SMART GRID, vol. 8, no. 5, 2017, pp. 2431-2439

[2] K. Gai, et al., Intrusion detection techniques for mobile cloud computing in heterogeneous 5G, SECUR COMMUN NETW, vol. 9, no. 16, 2016, pp. 3049-3058.

[3] Y. LeCun, et al., Deep learning, *NATURE*, vol. 521, no. 7553, 2015, pp. 436.

[4] S.R. C, Applying long short-term memory recurrent neural networks to intrusion detection, *South African Computer Journal*, vol. 1, no. 56, 2015, pp. 136-154.

[5] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi and M. Ghogho, "Deep learning approach for Network Intrusion Detection in Software Defined Networking," *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Fez, 2016, pp. 258-263.

[6] W. Lin, H. Lin, P. Wang, B. Wu and J. Tsai, "Using convolutional neural networks to network intrusion detection for cyber threats," *2018 IEEE International Conference on Applied System Invention (ICASI)*, Chiba, 2018, pp. 1107-1110.

[7] Pascanu, R., Mikolov, T., & Bengio, Y. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, 2013, pp. 1310-1318.

[8] H. S and S. J, Long short-term memory, *NEURAL COMPUT*, vol. 8, no. 9, 1997, pp. 1735-1780.

[9] Barbara, D , N. Wu , and S. Jajodia . "DETECTING NOVEL NETWORK INTRUSIONS USING BAYES ESTIMATORS." *Siam Conference on Data Mining* , 2001.

[10] M, P., A. A and R.P. M, Discriminative multinomial naive bayes for network intrusion detection. 2010. p. 5-10

[11] T. Ambwani, "Multi class support vector machine implementation to intrusion detection," *Proceedings of the International Joint Conference on Neural Networks*, Portland, OR, vol.3, *2003*, pp. 2300-2305.

[12] P. Nskh, M. N. Varma and R. R. Naik, "Principle component analysis based intrusion detection system using support vector machine," *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, 2016, pp. 1344-1350.

[13] Hasan, M. , Nasser, M. , Pal, B. and Ahmad, S. Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS). *Journal of Intelligent Learning Systems and Applications*, Vol. 6, No. 1, 2014, pp.45-52.

[14] L. P. Rajeswari and A. Kannan, "An Intrusion Detection System Based on Multiple Level Hybrid Classifier using Enhanced C4.5," *2008 International Conference on Signal Processing, Communications and Networking*, Chennai, 2008, pp. 75-79.

[15] Stein, Gary , et al. "Decision tree classifier for network intrusion detection with GA-based feature selection." *Southeast Regional Conference* ACM, 2005.

[16] S. Thaseen and C. A. Kumar, "An analysis of supervised tree based classifiers for intrusion detection system," *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, Salem, 2013, pp. 294-299.

[17] Javaid, A. Y. , Niyaz, Q. , Sun, W. , & Alam, M. "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.

[18] Erxue, Min , Jun, L. , Qiang, L. , Jianjing, C. , & Wei, C."TR-IDS: Anomaly-Based Intrusion Detection through Text-Convolutional Neural Network and Random Forest." *Security and Communication Networks*, 2018, pp. 1-9.

[19] C. Yin, Y. Zhu, J. Fei and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," in *IEEE Access*, vol. 5, 2017, pp. 21954-21961.

[20] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in *Neural Computation*, vol. 9, no. 8, 15 Nov. 1997, pp. 1735-1780.

[21] J. Kim, J. Kim, H. L. T. Thu and H. Kim, "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," *2016 International Conference on Platform Technology and Service (PlatCon)*, Jeju, 2016, pp. 1-5.

[22] Quinlan, J. R.. Induction of decision trees. *Machine Learning,* vol. 1, 1986, pp. 81-106.

[23] Revathi, S., and A. Malathi. "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection." *International Journal of Engineering Research and Technology. ESRSA Publications*, 2013.

[24] University Of California, I., DERIVED FEATURES. 1999. [EB/OL]. http://kdd.ics.uci.edu/databases/kddcup99/task.html.