International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 3, June 2020, pp. 3315~3322
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i3.pp3315-3322

LSTM deep learning method for network intrusion detection system

Alaeddine Boukhalfa, Abderrahim Abdellaoui, Nabil Hmina, Habiba Chaoui
System Engineering Laboratory, ADSI Team, National School of Applied Sciences, Ibn Tofail University, Morocco

Article Info

Article history:

Received Jun 15, 2019
Revised Dec 4, 2019
Accepted Dec 14, 2019

Keywords:

Deep Learning
LSTM
Machine Learning
NIDS
RNN

ABSTRACT

The security of the network has become a primary concern for organizations. Attackers use different means to disrupt services, these various attacks push to think of a new way to block them all in one manner. In addition, these intrusions can change and penetrate the devices of security. To solve these issues, we suggest, in this paper, a new idea for Network Intrusion Detection System (NIDS) based on Long Short-Term Memory (LSTM) to recognize menaces and to obtain a long-term memory on them, in order to stop the new attacks that are like the existing ones, and at the same time, to have a single mean to block intrusions. According to the results of the experiments of detections that we have realized, the Accuracy reaches up to 99.98 % and 99.93 % for respectively the classification of two classes and several classes, also the False Positive Rate (FPR) reaches up to only 0,068 % and 0,023 % for respectively the classification of two classes and several classes, which proves that the proposed model is effective, it has a great ability to memorize and differentiate between normal traffic and attacks, and its identification is more accurate than other Machine Learning classifiers.

Corresponding Author:

Alaeddine Boukhalfa,
System Engineering Laboratory, ADSI Team,
National School of Applied Sciences,
Ibn Tofail University, Kenitra, Morocco.
Email: alaeddine.boukhalfa@gmail.com

1. INTRODUCTION

Nowadays, the world is experiencing a great revolution in the field of information technology, everybody is exchanging continuously information across the network. This implies the establishment of new tools and mechanisms of prevention and detection, and the strengthening of those that exist, like Network Intrusion Detection System (NIDS), in order to enhance security and protect the network from intrusions. The function of a NIDS is to observe, evaluate and classify traffic transiting through the network, it is based, in advance, on established methods and techniques in order to differentiate between normal and suspicious traffic. Furthermore, attackers are attracted by information and knowledge passing through the network, and to exploit and profit from them, they are forced to overcome obstacles and barriers of security by creating new attacks, and evolving the existing ones. While the current NIDS are not evolutionary, their identification algorithms do not progress to identify automatically new menaces, which pushes us to think about advanced and intelligent methods of detection that can identify new attacks and accompany the progression of the existing ones.

Moreover, attacks can be of different types, like DoS (Denial-of-Service) and U2R (User to Root) etc…, this problem of diversity leads us to find a resolution to detect and stop them all in a unique way. Currently, Deep Learning is experiencing huge success in several domains, it is a set of techniques used to recognize objects, extract information hidden in the data, and make predictive analytics [1], one of these methods characterized by its long-term memory is the Long Short-Term Memory (LSTM) [2]. And, to solve the issues cited above, we propose in this paper a new approach for NIDS based on the Deep Learning

method LSTM, which will recognize attacks and keep a long-term memory of them, in order to block the other new attacks, and at the same time, will deal, with a single way, with all type of these attacks. To verify the effectiveness of our proposed method LSTM for NIDS we apply it on NSL KDD [3] dataset, and we give a comparison of its capacity to memorize and detect intrusion with the famous Machine Learning classifiers like Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Trees. The paper is structured as follows. In section 2, we give a summary of the related work. We define our experimental environment in section 3. Section 4 is reserved for the exposure of results and analysis. At the end, section 5 announces the conclusion and the future work.

2. RELATED WORK

The exploitation of Deep Learning in a NIDS was debated by the authors in [4], one of Deep Learning methods Self-Taught Learning (STL) was evaluated on NSL KDD [3], which is a basis that contains traffic records of the network. The study compared the level of recognition with an ancient method of classification Soft-Max Regression (SMR), the results of the evaluation confirmed that STL identifies attacks better. This proposal presents effectively a remarkable step to separate normal traffic from doubtful traffic, except that the solution was not really realized by a real NIDS.

Approximately, the same demarche was adopted in [5], the version previous of NSL KDD called KDD Cup 99 [6], was employed to compare the precision of traffic identification. The results of experimentations proved that Support Vector Machine method based on Restricted Boltzmann Machine (SVM-RBMs) [7] determines certainly the nature of the traffic better than old classification algorithms, also it consumes minimum time for treating a large mass of data. The authors did not justify the reasons for the choice of the used Deep Learning method.

Another different concept was planned to catch code injection attacks associated with the JavaScript code in the paper [8], a new combination of Deep Learning named Hybrid Deep Learning Network (HDLN) was created. Performances of this latter were judged according to two levels, at the first, relatively to the number of hidden layers, the number of filters and number of neurons, the results proved that accuracy increases as the number of filters increases, secondly, it was confronted with other traditional classifiers, the marked accuracy was clearly the greatest. In the end, they affirmed that the antecedent work was improved in terms of accuracy by this new model. The effort provided is considerable, except that, the solution was limited to injection attacks attached to JavaScript code, and it is not dedicated to another type of attack.

The authors expose, in the paper [9], a new idea of Deep Learning, which associates Auto-Encoder and Deep Belief Network (DBN). The Auto-Encoder was used for the purpose of decreasing the dimensionality of data and identifying the principal features of it, while the DBN had the mission of detecting the dubious code. The test of the new suggestion of model was done with the dataset KDD Cup 99 [6], the assessment of results was compared with only a single DBN. The attainment has announced that the new method is completely more accurate with less consumption of time. However, the authors did not specify why they preferred to combine DBN and Auto-Encoder to form this hybrid.

In the world of the Internet of Things (IoT), another dispersed model to reveal intrusions was opted in [10]. On each node of fog-to-things networks, the Deep Learning has been set up in order to get autonomy to disclose threats and transfer the parameters to neighbouring nodes, so that they can update themselves against menaces. The objective was to speed up the identification and get a quick and local update of nodes parameters. The authors validated the success of the modern structure via NSL KDD, and they convinced that this conception is more efficient than the other centralized. The design provides a fascinating and advanced vision in the field of attacks detection, but the proposers did not specify exactly the applied Deep Learning method.

A subject of security monitoring in a Big Data context was debated by the manuscript [11]. Firstly, the authors explained the necessities of monitoring of security, which are anticipation problems, the adaptation of devices and security tools to a large amount of data, the rapid detection of the abnormal alarm and the establishment of appropriate diagnosis of alert, the limitation of security correlation algorithms to merely a few components and not all network components. Secondly, they exhibit a system to monitor security, which leans on the aggregation of Big Data, integration and extracting knowledge for decision making. Lastly, they describe some correlation algorithms to inspect the data. The paper announces the design of a security monitoring system for a large mass of data, but, it did not take into consideration the progress of the correlation methods to follow the evolution of the attacks.

One more vision to monitor the security of the Internet of Things (IoT) was discussed in [12], a large quantity and a diversity of logs of security was collected from electronic apparatus of users, hereafter, they were gathered in a distributed way using the framework Hadoop. The data was of different formats, the operation of standardization was carried out to unify them, the process of aggregation and correlation was conducted by Complex Event Processing (CEP) method, the outcome was presented adopting developed

tools for displaying. The paper is mainly oriented towards visualizing logs without any intervention against menaces. Our work presents a new approach compared to the other solutions, it is oriented to memorize a long-term attacks in order to discover the new others, and to deal with all intrusions in a unique manner.

3. EXPERIMENTAL ENVIRONMENT

In this section, we describe the data for experimentation, the evaluation indicators, the work environment, the adopted method and the compared methods.

3.1. Dataset and pre-treatments

3.1.1. Dataset

To evaluate our model, we used the NSL KDD [3] dataset. As mentioned above in the previous section, it is a derived version from KDD Cup 99 [6], that groups network traffic collected by 1998 DARPA IDS [4]. NSL KDD contains normal records, and records of attacks namely: DoS (Denial-of-Service) which destroy the service availability [13], Probe which extracts detailed information from the servers [14], U2R (User to Root) which try to exploit vulnerabilities in the system in order to obtain super user privileges [15], and R2L (Remote to Local) which send packets to a machine over a network who have no account on in order to lead to vulnerability issues and access secure information [16]. The distribution is illustrated in Table 1 and Table 2, Table 1 shows the distribution in two classes, whereas Table 2 shows the distribution in five classes.

Table 1. Distribution of dataset in two classes

Traffic	Number of samples
Normal	67343
Attack	58630
Total	125973

Table 2. Distribution of dataset in five classes

Traffic		Number of samples
Normal		67343
	DoS	45927
Attack	Probe	11656
	R2L	995
	U2R	52
Total		125973

The dataset does not include redundant records, it contains 43 columns, 42 columns define the characteristics of the recording as Duration, Protocol_Type, Service, Flag, etc..., and one column defines if it is a normal record or an attack, this column represents the label of the record.

3.1.2. Pre-treatment

For data preparation requirements before treatment, on the one hand, we have tried to normalize our dataset by converting character columns to numeric columns with the help of the famous 1-to-n encoding technique, Figure 1(a) and Figure 1(b) explain the differences before and after pre-treatment. And on another hand, we have separated the label column from the other columns via an Extract-Transform-Load (ETL).

0	tcp	ftp_data	SF	491	…	normal	20
0	udp	other	SF	146	…	normal	15
0	tcp	private	S0	0	…	neptune	19
0	tcp	http	SF	232	…	normal	21
0	tcp	http	SF	199	…	normal	21

a

0	1	1	1	491	…	1	20
0	2	2	1	146	…	1	15
0	1	3	2	0	…	2	19
0	1	4	1	232	…	1	21
0	1	4	1	199	…	1	21

b

Figure 1. (a) Dataset rows before pre-treatment, (b) Dataset rows after pre-treatment

3.2. Work environment, evaluation method and performance indicators

3.2.1. Work environment

The configuration of our machine is: the operating system is Windows 7, with Intel (R) Core (TM) i3 2370M CPU @ 2.40 GHZ (4 CPUs), and 4096MB of RAM. The ETL is Talend Open Studio (TOS) for Data Integration, which is an open source software [17], efficient, flexible and easy to handle [18].

3.2.2. Evaluation method

The k-fold cross validation method is employed to measure the success of a classifier, it splits the dataset into two subsets, the first for training and the second for testing [19]. The operation is repeated k times separately, and the average of the k performances is calculated and returned. The advantage of this method is that the entire dataset is used for both training and testing, which makes the evaluation more accurate. We adopted 5-fold cross validation to evaluate our model, if we increase the k, some attacks like U2R and R2L will decrease for each subset, and they can be neglected during the treatment. To separate the training subset and testing subset, we also employed the ETL (TOS).

3.2.3. Performance indicators

The model assessment indicators are:

$$Accuracy=TP+TN/(TP+TN+FP+FN) \quad (1)$$

$$Sensitivity=TP/(TP+FN) \quad (2)$$

$$False\ Positive\ Rate =FP/(FP+TN) \quad (3)$$

$$Precision=TP/(TP+FP) \quad (4)$$

$$Recall =TP/(TP+FN) \quad (5)$$

$$F\text{-}Measure=2*(Precision.Recall)/(Precision+Recall) \quad (6)$$

Where:

- Accuracy is the fraction of true detection overall data instances.
- Sensitivity defines the ability of the model to detect correctly.
- False Positive Rate (FPR) is calculated as the ratio of negative events wrongly classified as positive to the total negative events.
- Precision is the fraction of relevant instances among the all proposed instances.
- Recall is the fraction of relevant instances that have been found over the total of relevant instances.
- F-Measure gives the harmonic mean of precision and recall.
- TP, TN, FP and FN are retrieved from the confusion matrix, they mean respectively: True Positive, True Negative, False Positive and False Negative.

3.3. Adopted method and compared methods

3.3.1. Adopted method

As mentioned above, LSTM is a Deep Learning method, it is specially a Recurrent Neural Network (RNN) [20], which is characterized by its memory, that why it is adopted in this work, in order to memorize as long as possible attacks and predict new others. As shown in the Figure 2 [21] the LSTM gathers: an Input Gate which determines if a new input can transit or not, a Forget Gate which deletes information if is not important or let it impact the output, an Output Gate which determines the output, a single Cell which represents the Constant Error Carousel, and the activation functions which compute the activation of the three gates.

3.3.2. Compared methods

There are several methods in Machine Learning domain, so it is difficult to compare our suggested method to all these techniques, we will try to compare it only with the most efficient and popular of them, as SVM, KNN [22] and Decision Tree. SVM is a classifier established on margins, it uses small sample and achieves good generalization results [23]. KNN is a simple and efficient technique which uses the closest training examples to classify object in the feature space [24]. A decision tree is a method of classification, the various possible decisions are located at the ends of the branches (the leaves of the tree) and are reached according to decisions taken at each step [25].

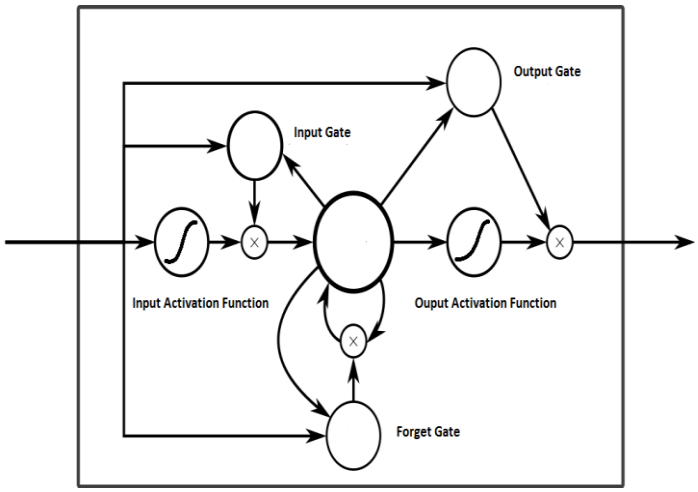

Figure 2. LSTM architecture

4. RESULTS AND ANALYSIS

This part is dedicated to announce and discuss the various obtained results. We have evaluated the model for two types of classification, binary and multi-classification. For binary classification the dataset is divided on two classes: class of normal records and class of attacks. For multi-classification, and Given U2R are not dense in term of the number of attacks, the classification results of this type of attack are not satisfactory, so we decided to group them with R2L attacks in one class, so the dataset is divided on four classes: class of normal records and three classes of three categories of attacks (Probe, DoS, U2R-R2L). We have evaluated the metrics: Accuracy, Sensitivity, False Positive Rate, Precision and Recall, and we have compared them with the others of the other classifiers.

Figure 3, Figure 4 and Figure 5 show respectively: Accuracy, average of Sensitivity and average of False Positive Rate (FPR) for binary and multi-classification. While Table 3 and Table 4 reveal Precision, Recall and F-Measure for binary and multi-classification. As indicated by Figure 3, the reached values of the Accuracy are 99,98 % for two classes classification and 99,93 % for four classes classification, which means that LSMT can properly memorize and identify traffics, and its detection capacity is better than the other machine learning classifiers. Also, as exposes Figure 4, the values of the average of the Sensitivity reached by the model LSTM are 99,986 % for binary recognition and 99,738 % for multi-recognition, this explains that the suggested model is very able to differentiate correctly between the different types of traffic, better than the other models. In addition, as noted by Figure 5, the values of average of False Positive Rate (FPR) achieved by LSTM are only 0,068 % for two classes classification and 0,023 % for multi-classes classification, which means that the margin of error of the detection of the method is minimal, and the values achieved are minimal compared to other classifiers.

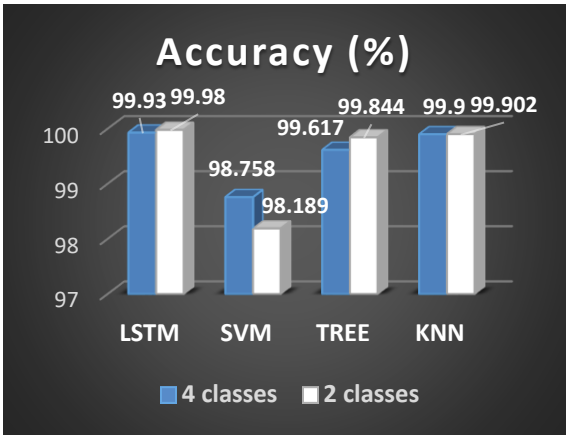

Figure 3. Accuracy for binary classification and multi-classification

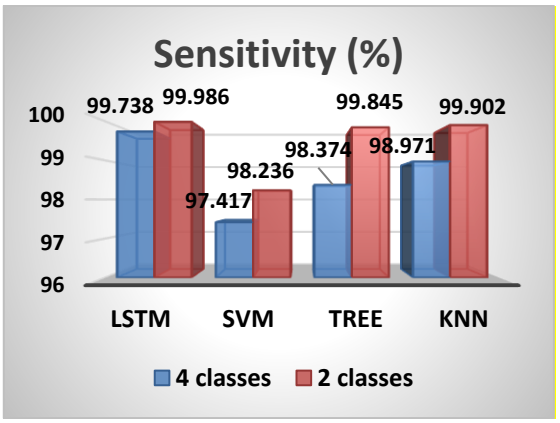

Figure 4. Average of Sensitivity for binary classification and multi-classification

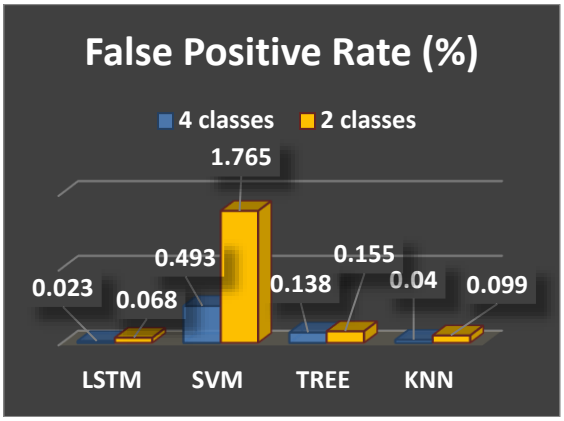

Figure 5. Average of false positive rate (FPR) for binary classification and multi-classification

Table 3. Recall, precision and F-Measure of two classes

Classifier	Class	Precision (%)	Recall (%)	F-Measure (%)
LSTM	Normal	99,999	99,973	99,986
	Attack	99,969	99,998	99,983
SVM	Normal	99,029	97,569	98,294
	Attack	97,254	98,902	98,071
Tree	Normal	99,887	99,820	99,853
	Attack	99,794	99,870	99,832
KNN	Normal	99,901	99,917	99,909
	Attack	99,904	99,886	99,895

Table 4. Recall, precision and F-Measure of four classes

Classifier	Class	Precision (%)	Recall (%)	F-Measure (%)
LSTM	Normal	100	99,938	99,969
	DoS	99,924	99,906	99,915
	U2R, R2L	95,896	99,106	97,475
	Probe	99,863	100	99,931
SVM	Normal	99,173	98,597	98,884
	DoS	99,128	99,782	99,454
	U2R, R2L	89,507	95,320	92,322
	Probe	95,803	95,968	95,885
Tree	Normal	99,821	99,632	99,726
	DoS	99,772	99,867	99,819
	U2R, R2L	86,824	95,033	90,743
	Probe	99,089	98,962	99,025
KNN	Normal	99,914	99,920	99,917
	DoS	99,974	99,980	99,977
	U2R, R2L	97,483	96,180	96,827
	Probe	99,743	99,803	99,773

The supreme values of Precision, as illustrated by Table 3 and Table 4, are those of our proposed Deep Learning model LSTM. For the identification of two classes, the Precisions reach up to 99,999 % and 99,969 % for respectively normal traffic and attack traffic, more than the other classifiers. For the identification of four classes, the Precision reaches up to 100 % for normal records, this is due to the density of this class of traffic, also for Probe and DOS attacks, the maximum Precisions achieved are respectively 99,863 % and 99,924 % more than the other classifiers, only one minimal value of Precision 95,896 % noted by LSTM (less than only that of KNN 97,483 %) in the case of the classification of the class U2R-R2L, explained by its minimum density. This justifies that the LSTM is generally very accurate more than the others. The values of Recall of LSTM for the identification of two classes, as shown in Table 3, are very high, 99,973 % for normal traffic and 99,998 % for attack traffic. Also, the values of Recall for the identification of four classes, as shown in Table 4, are also very high, 99,938 % for the normal class, 99,106 % for U2R-R2L class, 100 % for Probe class, and 99,906 % for DOS class (less than only that of KNN 99,980 %). This proves that LSTM can find normal instances and attack instances more than the other models. The experiment result has proved that the new method LSTM is very efficient, it can effectively memorize and differentiate between traffics: normal and attack, in the both cases of classification, binary and multi-classification.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new idea for NIDS established on the Deep Learning method LSTM, which will recognize attacks and keep a long-term memory of them, in order to block the other new attacks, and at the same time, will treat, with a unique manner, all type of these attacks. To validate the effectiveness of our new suggested approach, we employed the famous NSL KDD as dataset for training and testing, and Accuracy, Sensitivity, False Positive Rate, Precision and Recall as metrics for the evaluation, and we have compared the new method LSTM to the other Machine Learning classifiers. The experiment has demonstrated that the metrics of the detection of the LSTM method reach very high values more than the other classifiers. which proves that our new proposed method is effective for NIDS. In the future, we plan to implement really a new intelligent NIDS in the real world using our new proposed Deep Learning model LSTM.

REFERENCES

[1] Yann L., *et al*., "Deep learning," *Nature, International Journal of Science*, vol. 521, pp. 436-444, 2015.

[2] Gensler A., *et al*., "Deep Learning for Solar Power Forecasting–An Approach Using Autoencoder and LSTM Neural Networks," in: *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, Article no. 7844673, pp. 2858-2865, 2017.

[3] NSL KDD, [Online]. Available: https://github.com/defcom17/NSL_KDD.

[4] Niyaz Q., Sun W., Javaid A. and Alam M., "A Deep Learning Approach for Network Intrusion Detection System," In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT'15)*, United States ACM, pp. 21-26, 2015.

[5] Dong B. and Wang X., "Comparison Deep Learning Method to Traditional Methods using for Network Intrusion Detection," *8th IEEE International Conference on Communication Software and Networks*, pp. 581-585, 2016.

[6] KDD Cup 99, [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[7] Yang J., Deng J., Li S. and Hao Y., "Improved Traffic Detection with Support Vector Machine based on Restricted Boltzmann Machine," *Soft Computing*, vol. 21, no. 11, pp. 3101–3112, 2017.

[8] Yan R., Xiao X., Hu G., Peng S. and Jiang Y., "New Deep Learning Method to Detect Code Injection Attacks on Hybrid Applications," *Journal of Systems and Software*, vol. 137, pp. 67–77, 2018.

[9] Li Y., Ma R. and Jiao R., "A Hybrid Malicious Code Detection Method Based on Deep Learning," *International Journal of Security and Its Applications (IJSIA)*, vol. 9, no. 5, pp. 205–216, 2015.

[10] Diro A.A. and Chilamkurti N., "Distributed Attack Detection Scheme using Deep Learning Approach for Internet of Things," *International Journal Future Generation Computer Systems (FGCS)*, vol. 82, pp. 761-768, 2018.

[11] Lan L. and Jun L., "Some Special Issues of Network Security Monitoring on Big Data Environments," In: *2013 IEEE 11th International Conference on Dependable Autonomic and Secure Computing*, pp. 10-15, 2013.

[12] Saenko I., Kotenko I., and Kushnerevich A., "Parallel Processing of Big Heterogeneous Data for Security Monitoring of IoT Networks," In: 2017 *25th Euromicro International Conference on Parallel, Distributed and Networks-Based Processing*, pp. 329-336, 2017.

[13] El-Sofany H.F., El-Seoud S.A, Taj-Eddin, I.A.T.F., "A Case Study of the Impact of Denial of Service Attacks in Cloud Applications," *Journal of Communications*, vol. 14, no. 2, 2019.

[14] Aljawarneh S., Aldwairi M., Yassein, M.B., "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152-160, 2018.

[15] Sornsuwit P., Jaiyen S., "Intrusion detection model based on ensemble learning for U2R and R2L attacks," in: *2015 7th International Conference on Information Technology and Electrical Engineering, Envisioning the Trend of Computer, Information and Engineering (ICITEE)*, Article no. 7408971, pp. 354-359, 2015.

[16] Dubey S., Dubey J., "KBB: A hybrid method for intrusion detection," in: *IEEE International Conference on Computer, Communication and Control (IC4-2015),* Article no.r 7375704, 2016.
[17] Venkatesh Pulla V.S., Varol C., Al M., "Open source data quality tools: Revisited," *Advances in Intelligent Systems and Computing*, vol. 448, pp. 893-902, 2016.
[18] Majchrzak T.A., Jansen T. and Kuchen H., "Efficiency Evaluation of Open Source ETL Tools," In: (SAC '11) *Proceedings of the 2011 ACM Symposium on Applied Computing,* pp. 287-294, 2011.
[19] Refaeilzadeh P., Tang L. and Liu H., "Cross-Validation," *Encyclopedia of Database Systems, Springer,* 2009.
[20] Zhao Z., Chen W., Wu X., Chen P.C.Y., Liu J., "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems,* vol. 11, no. 2, pp. 68-75, 2017.
[21] Greff K., Srivastava R.K., Koutnik J., Steunebrink B.R. and Schmidhuber J., "LSTM: A Search Space Odyssey," *2017 IEEE Transactions on Neural Networks and Learning Systems,* vol. 28, pp. 2222-2232, 2017.
[22] Xie L., Fu Z.H., Feng W. and Luo Y., "Pitch-Density-Based Features and an SVM Binary Tree Approach for Multi-Class Audio Classification in Broadcast News," *Multimedia Systems,* vol. 17, no. 2, pp. 101-112, 2011.
[23] Kuang F., Xu W., and Zhang S., "A Novel Hybrid KPCA and SVM with GA Model for Intrusion Detection," *Applied Soft Computing,* vol. 18, pp. 178-184, 2014.
[24] Amin Aburomman A., and Ibne Reaz M.B., "A novel SVM-kNN-PSO Ensemble Method for Intrusion Detection System," *Applied Soft Computing,* vol. 38, pp. 360-372, 2016.
[25] Tsai C.F., Hsu Y.F., Lin C.Y., and Lin W.Y., "Intrusion Detection by Machine Learning: A Review," *Expert Systems with Applications,* vol. 36, no. 10, pp. 11994-12000, 2009.