International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

An Intrusion Detection Model Based on Deep Long Short Term Recurrent Neural Network

K. Narayana Rao, K. Venkata Rao, Prasad Reddy PVGD

Abstract: ***With the rapid increase of network based services and internet users on various platforms are becoming the major targets of attacks. Intrusion detection is the process of monitoring the attacks and analyzing their signs and violation of security policies which are occurring in the systems or networks. Intrusion Detection System is a prominent research area in security analysis and evaluation. In order to identify the attack type, we proposed Deep Long Short Term Memory-Recurrent Neural Network (DLSTM-RNN) method with seven optimizers and 500 epochs to train and test a dataset. Initially the data transformation, normalization are used to preprocess the data. The preprocessed train and test data is given input to the model. The bench mark NSL-KDD dataset used to train and test the model. The results are obtained for five-class classification (attack types).The model outperformed with adamax optimizer on NSL-KDD dataset. The metrics accuracy, detection rate, and false alarm rate areevaluated to ascertain the detection efficacy of the model. We compare the model to existing convolutional learning methods.***

Keywords: Deep Learning, Long Short Term Memory, Optimizer, Intrusion Detection.

I. INTRODUCTION

With the increase of internet usage, cyber security has become an important task. The identification of network attacks, especially unanticipated attacks are a technically key issue. There are a few conventional softwares to detect harmfull actions, such as control mechanisms, firewalls, encryption methods etc... These techniques have limitations, especially when the systems are facing a large number of attacks like Denial of Services (DoS), and the systems can get a higher detection rate of false positive and false negative. The significant researchers are actively focused on developing Intrusion Detection Systems (IDS) in an information security field. According to dynamic detection methods the IDS could be categorized into three types. They are Misuse-based, Anomaly-based IDSs and Stateful Protocol analysis. The misuse-based IDS detect the familiar attacks based on pre determined attack signature. Therefore, frequent updating of new signature is obligatory. However the MIDS cannot detect the rapidly growing zero day vulnerabilities and exploits. Anomaly-based IDSs are used to develop to detect any deviations from profiles of usual behaviour. Therefore,anomaly-based IDSs are more acceptable systems than misuse-based IDSs for discovering novel or unknownattacks without any antecedent knowledge. The stateful protocol analysis detection approach compares the detected actions and recognizes the unconventionality of the protocol state[1]. It takes advantage of both methodologies, such as signature based, and anomaly-based IDSs. In fact, IDS is generally equivalent to a classification problem, such as binary or multi-classification. The binary classification is, identifying whether the traffic is normal or abnormal. The five-category multi classification is identifying the traffic whether the attack is normal or one of the other four attack types, such as Denial of Services (DoS), Probing (Probe), User to Root (U2R), and Remote to Local (R2L). The motives behind IDS are to accurately determine the behavior of intrusive users by enhancing classification accuracy.

Recently, the researchers have been using Machine Learning techniques with the aim of improving the detection rates as compared to conventional intrusion detection techniques. The traditional methodologies can not efficiently solve the vast intrusion data classification in the real world environment. With the dynamic growth of data set size, the classification methods will induce decrease in the accuracy. In contrast, the deep learners are likely to extract better representation from the massive data to devise effective models. In this paper, we study few aspects which consist of Intrusion Detection Machine Learning techniques. Then, we continue to give a new method Long Short Term Memory-Recurrent Neural Network (LSTM-RNN) to improve the performance in this area.

This paper is organized as follows. In section II, we synopsize the related research study in the area of IDS. Section III, presents a description of LSTM-RNN architecture. The performance evaluation and it's measures are presented in section IV. This section highlights the evaluated methodology with a discussion of experiment results and a comparision with preceding research works on NSL-KDD dataset. Finally, the conclusion is disclosed section V.

II. RELATED STUDY

In the previous studies, the conventional IDSs areshown using as misuse or anomaly detection to detect unknown attacks. The researchers have been applying machine learning methodologies for intrusion detection.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
K. Narayana Rao*,Research Scholar, Department of Computer Science and Systems Engineering, Andhra University College of Engineering (A), Andhra University, Visakhapatnam, AP, India.
Prof. K. Venkata Rao, Professor, Department of Computer Science and Systems Engineering, Andhra University College of Engineering (A), and DeanAcademic Affairs, Andhra University, Visakhapatnam, AP, India.
Prof. Prasad Reddy, P.V.G.D, Sr. Professor, Department of Computer Science and Systems Engineering, Andhra University College of Engineering (A), and Vice-Chancellor, Andhra University, Visakhapatnam, AP, India.

Existing IDSs are heavily based on supervised learning methods such as SVM, KNN, Random Forest, etc… Our related study review [2], discussed various related works on IDS through machine learning. Kayacik et al. [3] proposed hierarchical two layer SOM, on 41-features of KDD data set. Ravinder Reddy et al. [4] proposed discriminant function with SVM to increase the effectiveness. Jiong Zhang et al. [5] applied Random Forest in misuse, anomaly, and hybrid detection.

Researchers also proposed several hybrid classification approaches by the amalgamation of existing methods. Dhikhi et al.[6] proposed CWS-IDS to find the anomalies, that combines the unsupervised Contractive Autoencoder (ContAE) and SVM. The method used Autoencoder for feature extraction and SVM for classifiying the attacks as binary or multi classification on NSL-KDD dataset. The results were compared with single SVM and Random forest methods. Nandini Rebello et al.[7] proposed hybrid model with K-means and Gradient Boosted tree classifier methods. The K-means clustering method, forms clusters from the input dataset and Gradient Boosted Tree classifier classify the tested data into binary classification. The evaluated results were compared with existing methodologies. Ahmed I. Saleh et al. [8] implemented Hybrid IDS for multi-class classification problems. Naïve Bayes Feature Selection (NBFS) technique used for reduction of dimensionality. The Optimized Support Vector Machine (OSVM) employed for outlier rejection. Prioritized KNN (PKNN) classifier classified the attacks successfully. However these methods are produced many false alarms and have low detection rate of attacks in IDS.

Sasanka Potluri et al. [9] implemented accelerated Deep Neural Network (DNN) to identify the anomalies in NSL-KDD dataset. Pavel Kachurka et al. [10] used Recurrent Neural Network (RNN) method to detect attacks which are unseen previously in real time. However, Machine Learning methods have some limitations, with the increase of input data size and unforeseen attacks. The upgraded learning methods are needed, particularly in the extraction of features and analysis of intrusion. Recently, researchers have shown an interest on IDS using Deep Learning Methods. It is a branch of Machine Learning, and has better learning capability of analysis of composite data. Sheraz Naseer et al. [11] implemented different Deep Neural Networks models for IDS including CNN, Autoencoders, and RNN. Many deep learning models are used GPU powered test-bed for train and test of NSL-KDD dataset.

III. PROPOSED METHODOLOGY

A) Recurrent Neural Network

Deep learning is a branch of Machine Learning, which is closer to Artificial Intelligence. Supervised and unsupervised learning models construct using Deep learning with higher levels of abstraction. Features are defined from lower levels. Neural network is a workhorses of Deep learning. In typical neural network, at time t, the neuron output is

$$y_i^t = \sigma(W_i x_t + b_i)$$

The weight matrix is W_i, bias is b_i, and the sigmoid activation function is σ. RNN is an enhancement of feed-forward neural networks with recurrent connections. RNN is a prominent model to train the sequence data. In RNN, at time $t-1$, the neuron output is fed back into the neuron. The new activation function becomes as

$$y_i^t = \sigma(W_i x_t + U_i y_i^{t-1} + b_i)$$

As these RNNs are repeated connections using previous inputs, they rely on the previous states for their current output. Unlike a conventional feed-forward neural network, RNN has cyclic connections. The cyclic connections make RNN as more powerful for modelling sequences. Let us assume that the input vector sequence is X, the hidden vector sequence is H, and the output sequence is Y. The input sequence given by $X=(x_1,x_2,....,x_T)$. A conventional RNN calculates the hidden sequence as $H=(h_1,h_2,….,h_T)$, and the output vector sequence is $Y= (y_1,y_2,...,y_T)$ with $t = 1$ to T as follows.

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

$$y_t = W_{hy}h_t + b_y$$

Where W is a weight matrix, b_i is bias, and σ is nonlinearity activation function.

Inorder to manage variable-length data input, conventional RNN uses Back Propagation Training Time (BPTT). In this model, it is first trained with trained data, then with saved output gradient error for each time step. However RNN is difficult to train, because gradient is exploding or disappearing while training with BPTT method.

B) Long Short Term Memory

Hochreiter et. al [12] proposed LSTM architecture. LSTM can learn long term dependencies, and overcome vanishing gradient descent problem. LSTM remembers information for a long period of time, it replaces each common hidden node by LSTM cell. The long term dependencies of LSTMs full fills by keeping an internal state, that is memory cell of LSTM neuron. The LSTM cell is displayed in fig. 1.

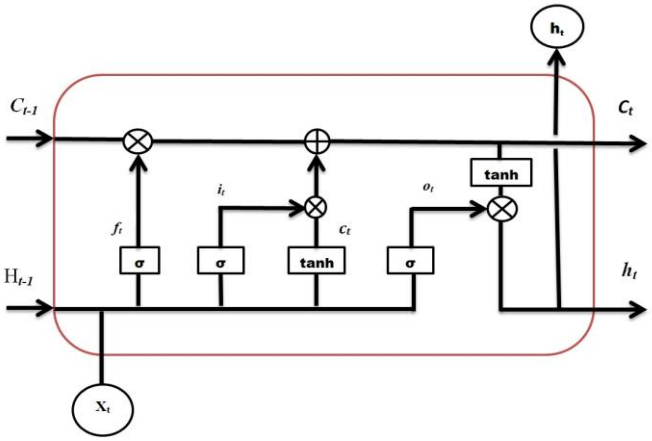


Fig. 1 LSTM Cell

Each LSTM cell has three gates such as input gate (i_t), forget gate (f_t), and output gate (o_t). The updates of input, forget, and output gates can be calculate as:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

The forget and input gate determine the contribution of the previous output and the current input, in the new cell state (c_t).

The output gate controls how much of C_t exposed as the output. The new cell state C_t and the output H_t can be calculated as :

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\ (W_{xc} h_{t-1} + b_c)$$

$$h_t = o_t \odot \tanh\ (c_t)$$

Where X_t, H_t, and C_t are input layer, hidden layer, and cell state at time t respectively. Besides, b_i,b_f,b_c, and b_o are bias at input, forget, cell, and output gates respectively. $\odot$ is element-wise multiplication, and σ is sigmoid function. W is weight matrix, in particular, the weight matrices W_{xi}, W_{xf}, W_{xc}, and W_{xo} which are connect from input layer to input gate, forget gate, cell state, and output gate respectively. The weights W_{hi}, W_{hf}, W_{hc}, and W_{ho} are connecting from hidden layer to input gate, forget gate, cell state, and output gate respectively. W_{ci}, W_{cf}, and W_{co} are weight matrices which connect from cell state layer to input gate, forget gate, and output gate respectively.

Our proposed DeepLSTM-RNN model used to apply for IDS with various optimizers. The proposed model performed as a multi classifier, and to classified the as Dos, Probe, U2R, R2L, and normal.

IV. EVALUATION & RESULTS

Like a existing deep learning approach , our proposed Deep LSTM-RNN classifiation model was developoed using tensorflow. Our model evaluation is done using tensorflow with personal GPU. The GPU cofigured with Intel® Core(TM) i5-8300H 2.30GHz, 8 GB RAM, and NVIDIA GTX 1050 and GPU has 64-bit windows 10 operating system.

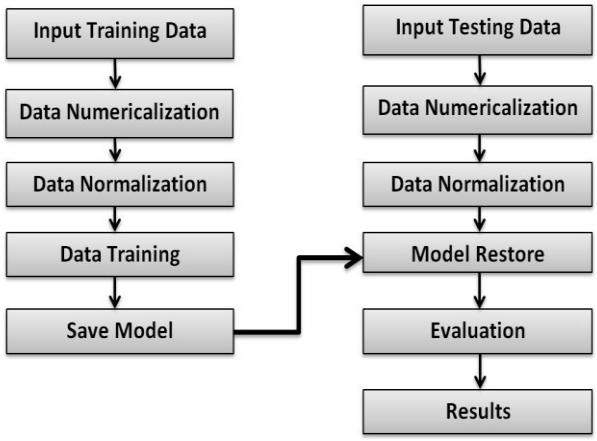


Fig. 2. An Overview of proposed Model

A) Metrics

The motivation of our proposed model is maximizing Accuracy and Detection Rate (DR), and curbing the False Alarm Rate (FAR). To perform our assesment, we used NSL-KDD dataset. In IDS research, this dataset is considered as bench mark dataset. The confusion matrix presented the actual/ expected and prediction classification result. The result of classification is predicted into five classes correctly and incorrectly. The proposed LSTM-IDS model is evaluated usingin the following metrics which are majorly used in IDS.

Accuracy: The percentage of the correctly classified instances versus a total number of instances.

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

True Positive Rate (TPR) or Detection Rate (DR): Part of positive instances correctly classified as positively.

$$TPR/DR = \frac{TP}{TP+FN}$$

Precision: The portion of correct prediction of intrusions versus total number of predicted intrusions.

$$Precision = \frac{TP}{TP+FP}$$

Recall: The portion of correctly predicted intrusions versus total actual intrusions.

$$Recall = \frac{TP+TN}{TP+FN+FP+TN}$$

False Positive Ratio: The portion of number of instances rejected incorrectly versus the total number of normal records.

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

The performance grows better, that means increases it the TPR and decreases the FPR. Therefore, we use the metric efficiency.

$$Efficiency = \frac{TPR}{FAR}$$

Where TP is predict anomaly records are as anomaly, FP is predict normal instances are incorrectly as attacks, TN is predict Normal instances as normal, and finally FN predict anomaly instanceare incorrectly as normal.

B) Dataset

DARPA initiative IDS-events at MIT Lincoln LAB in 1998. Later from DARPA network dataset files, KDD99 dataset was created by Lee and Stolfo[13], those were participated in DARPA team. The KDD99 can easily be used as a machine learning dataset. However it is far more used in IDS than DARPA dataset. The KDD dataset 38 attacks are divided into five main categories, such as Dos, Probe, R2L, U2R, and normal. The training and testing dataset contains 24, and 14 attacks respectively. The researchers identified several short comings in KDD dataset.

- It is heavily imbalanced i.e 80
- U2R and R2L are few in dataset
- Redundant datasets in both train and test
- It has large dataset, most of the studies used small percentage of it.

To reduce the inadequacy of KDD99 dataset, Tavallaee et al. [14] proposed NSL-KDD dataset. It was generated by removing redundant instances and decreasing the dataset size. The NSL-KDD dataset had 41 features, which are either continuous or discrete. The 41 features categorized into three groups, such as basic features, content features, and traffic features. The features No.1 to No.10 are basic features, No.11 to No.22 are content features, and No.23- No.41 are traffic features. Basic features encapsulate all the attributes extracted from TCP/IP connection.

Content features are suspicious behavior data, i.e failed login attempts. Traffic ctegory is categorized as same host with same service in current connection with respect to window interval. The NSL-KDD dataset feature type is continous, and symbolic. From the 41 features protocol_type, service, and flag features are Symbolic, and remaining features are Continous type. According to attack category the attacks are mapped in to four types of attack classes. The TABLE I shows attack classes.

TABLE I
NSL-KDD Attack types and its Classes [16]

Attack Class	Attack Type
DoS	neptune, smurf, back, teardrop, pod, land, apache2, mailbomb, processtable, udpstorm, worm.
Probe	satan, ipsweep, portsweep, nmap, mscan, saint.
R2L	warezclient, guess_passwd, warezmaster,imap, ftp_write, multihop, phf, spy, httptunnel, named,sendmail,snmpgetattack, snmpguess, xlock, xsnoop.
U2R	buffer_overflow, rootkit, loadmodule, perl, ps, sqlattack, xterm.

The table II shows NSL-KDD dataset distribution. There are 1,25,973 and 22,543 records in the train and test dataset respectively.

TABLE II
Distribution of NSL-KDD dataset sample distribution

	Normal	DoS	Probe	U2R	R2L	Total
Train	67,343	45927	11656	52	995	125,973
Test	9,710	7,460	2,421	67	2,885	22,543

C) Hyperparameters

The hyper-parameters are vital things to improve the result of neural network model. Depending on hyper-parameters values, the performance of the model changed. KalusGreff et al. [15] performed analysis on the impact of hyperparameters. We measured the performance of LSTM model for seven optimization functions with default optimization values for different hidden layers of the model. Table III shows optimization parameter values of each optimizer.

TABLE III
The optimizes values of the model

Optimizer	Optimizer Values
Adamax	lr=0.002, beta_1=0.9, beta_2=0.999, epsilon=None, decay=0.0
SGD	lr=0.01, momentum=0.0, decay=0.0, nesterov=False
Adagrad	lr=0.01, epsilon=None, decay=0.0
Adam	lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=None, decay=0.0, amsgrad=False
RMSprop	lr=0.001, rho=0.9, epsilon=None, decay=0.0
Nadam	lr=0.002, beta_1=0.9, beta_2=0.999, epsilon=None, schedule_decay=0.004
Adadelta	lr=1.0, rho=0.95, epsilon=None, decay=0.0

D) The model set up

Before training the dataset, we pre-processed the train and test dataset. Numericalization, Class Numericalization, and Normalization steps are involved in pre-processing.

(a) Numericalization

NSL-KDD dataset contain 38 numeric features and 3 nonnumeric features. The input values of the LSTM must be a numeric. We need to convert nonnumeric features into numeric features. The nonnumeric features are 'protocol_type', 'service', and 'flag'. The nonnumeric 'protocol_type' feature has 'tcp', 'udp', and 'icmp' features. These features as encoded with numerical values 1, 2, and 3. Similarly the 'service' and 'flag' features have 70 and 11 attributes respectively. These non numerical features are encoded as numerical values with help of one hot encoder.

(b) Class Numeralization

The attack type non numerical values are converted into numerical categories. For binary classification, the values 1 and 0 are assigned to normal and attack type respectively. In the multi-classification method the attack types are categorized as 0, 1, 2, 3, and 4 for DoS, Probe, R2L, U2R, and normal respectively. We used one hot encoder to convert it into numerical classification.

(c) Normalization

Some features minimum and maximum values difference are very large, such as duration [0,42908], src_bytes[0, 1379963888], dst_bytes[0,1309937401] etc.. We must have normalized numeric features for removing the effect of original feature scales. Each feature is normalized as

$$z_i = \frac{x_i - x_{Min}}{x_{Max} - x_{Min}}$$

Where $X = x_1, x_2, \ldots, x_m$ is the dataset with m samples, x_i is a feature vector, and z_i is the i^{th} normalized data. Where X_{min} is minimum value from data value, and X_{max} is maximum data value from feature vector. The min-max scalar works better if the distribution is not Gaussian or the standard deviation is quite small. Therefore all numeric features values are ranged between 0 and 1.

Therefore, the model input and output dimensions are 41 and 5 respectively. We apply LSTM-RNN model with batch size 100, and hidden layers sizes are 50, 75, and 100. The experiment evaluated with 500 epochs for specified hidden layer sizes. We used seven above mentioned optimizers such as Adamax, SGD, Adagrad, Adam, RMSprop, Nadam, and Adadelta. The loss function is sparse categorical cross entropy, which is suitable for multi classification. Table IV shows the accuracy of the model for various hidden layers with seven optimizers.

TABLE IV

Hidden layer size	**50**	**75**	**100**
Adamax	97.93	97.88	99.80
SGD	99.70	99.70	99.70
Adagrad	99.14	99.23	99.31
Adam	97.92	97.45	97.92
RMSprop	97.75	97.73	97.74
Nadam	97.70	98.64	97.38
Adadelta	97.47	97.50	97.29

E) Finding the model performance:

Depending the results of various hidden layers sizes, we could get the highest accuracywith 100 hidden layer size. From the results of 100 hidden layers, we set hyper parameters for training the model.

We implemented our model on IDS using NSL-KDD dataset with seven optimizes. The model is evaluated with sigmoid activation function. For training and tesing phase we used 1,25,973, and 22,543 records respectively. We design our measurement in two cases. In the first case, we measured the accuracy for seven optimizes. In the second case, we evaluated the metrics as multi-class classification. In case 1, we observed that the average classification performance of our implemented model using Adamax optimizer. Case 2 evaluated the multi class classification accuracy for seven optimizers.

From the results of three hidden layer size, SGD, and Adagrad optimizer given good result. RMSprop, adam and Adadelta optimizers have given approximately same result. Nadam gave different results and gave good result for layer size 100. Adamax optimizer outperform the accuracy of 99.80% with 100 hidden layer size. The Table V shows the evaluated results.

TABLE V
The classification performance of model

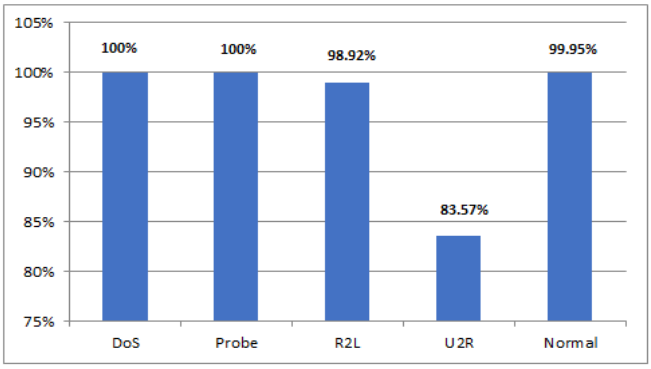
Optimizer	Accuracy	TPR	FAR	Precision	F1
RMSprop	0.9774	0.9970	0.0465	0.9632	0.9798
Adagrad	0.9930	0.9997	0.0155	0.9880	0.9938
Adadelta	0.9729	1.0	0.0591	0.9524	0.9756
Adam	0.9792	0.9997	0.0457	0.9637	0.9814
Adamax	**0.9979**	**0.9996**	**0.0043**	**0.9967**	**0.9982**
Nadam	0.9737	0.9995	0.0568	0.9544	0.9764
SGD	0.9969	1.0	0.0069	0.9947	0.9973

From the Table V our model outperform for Adamax optimizer with 99.84% accuracy, 99.96% is DR, and FAR is 0.0043. Table VI shows five-category classification confusion matrix of LSTM-RNN on the dataset KDDTest+ with Adamax optimizer.

TABLE VI
Confusion matrix for five-category classification

Actual / Predicted	DoS	Probe	R2L	U2R	Normal
DoS	7460	0	0	0	0
R2L	0	2421	27	0	0
U2R	0	0	2854	0	0
Probe	0	0	4	56	4
Normal	0	0	0	11	9706

Average percentage of every attack detection



The Table VII shows the multi classification accuracy for different optimizers. We achieved best classification results for the attacks DoS, and Probe, and for normal events.

TABLE VII

Optimizer	DoS	Probe	R2L	U2R	Normal
RMSprop	1.0	0.9995	0.8405	0.8358	0.9961
Adagrad	1.0	0.9847	0.9660	0.7313	0.9996
Adadelta	1.0	1.0	0.7944	0.7462	1.0
Adam	1.0	1.0	0.8426	0.8358	0.9996
Adamax	1.0	1.0	0.9892	0.8358	0.9995
Nadam	1.0	0.9954	0.8058	0.7910	0.9993
SGD	1.0	1.0	0.9996	0.0	1.0

F) Comparison:

Inorder to differentiate the performance of proposed LSTM-RNN model with different related work applied learning approachesto the NSL-KDD dataset and KDDCup'99 dataset. We constructed the training set and testing set from the NSL-KDD dataset. The training set consist of 1,25,973 samples, and testing set KDDTest$^+$ consist of 22,543 samples. In this experiment the detection rate of the proposed model got accuracy 99.97%, DR 99.96% and FAR is 0.43% on the test dataset KDDTest$^+$. The model outperformed with Adamax optimizer and sigmoid activation function.LSTM-RNN classifier [16] got the accuracy 96.93%, DR 98.88%, and FAR 10.04% with SGD optimizer on KDD Cup 99 dataset. The another LSTM-RNN classifier [17] performed accuracy 97.54%, DR 98.95%, and FAR 9.98% with Nadam optimizer on KDD Cup99 dataset. Table VIII shows the comparision of proposed model with other existing learning model on NSL-KDD dataset.

TABLE VIII
Comparision with other algorithm

	DR	FAR	Accuracy
RNN [18]	97.09%		81.29%
LMDRT- SVM [19]	99.20%	0.60	99.31%
OS-ELM [20]	98.26%	0.99	98.66%
LSTM-RNN [16]	98.88%	0.10038	96.93%
LSTM-RNN [17]	98.95%	0.998	97.54%
Our Proposed Model	**99.96%**	**0.0069**	**99.79%**

V. CONCLUSION

The proposed IDS LSTM-RNN evaluated on NSL-KDD dataset. The dataset is preprocessed, and normalized for both training and testing dataset. Inorder to find the proper optimizer, we took an experiment with 500 epochs and changed the values of hidden layer size. For training and testing phase, we took 1,25,973 and 22543 data samples respectively. The model outperformed with Adamax optimizer. Compared with other previous, and shallow classifier methods, our methodology outperformed with accuracy, and DR under multi classification.

In our future work, we will extend the capability of our model to handle zero-day attacks. Many of models obtained less percentage of accuracy to detect the U2R and Probe attacks. We will pay an attention to get good classification accuracy percentage of U2R and Probe attacks, reduce the training time, and come out with a more innovative approach.

ACKNOWLEDGMENT

Ministry of Electronics & Information Technology (MeitY), Government of India supported this work under Visvesvaraya PhD Scheme for Electronics & IT.

REFERENCES:

1. Karen Scarfone, and Peter Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS)”, National Institute of Standards and Technology, NIST Special Publication 800-94.
2. K. Narayana Rao, Prof. K. Venkata Rao, and Prof. Prasad Reddy P.V.G.D, “A Comprehensive survey of Machine Learning for Intrusion Detection”, International Journal of Research in Advent Technology, Vol.7(2), 2019, pp. 643-651.
3. H. Gunes Kayacik, A. Nur Zincir-Heywood, Malcolm I. Heywood, “A hierarchical SOM-based intrusion detection system”, Engineering Applications of Artificial Intelligence, 20 (2007), pp. 439–451.
4. R.Ravinder Reddy, Dr.Y Ramadevi, Dr.K.V.N Sunitha, “Effective Discriminant Function for Intrusion Detection Using SVM”, Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI) , 2016, 21-24.
5. Jiong Zhang, Mohammad Zulkernine, and Anwar Haque, “Random-Forests-Based Network Intrusion Detection Systems”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, 2008, VOL. 38.
6. Dhikhi T, and M.S. Saravanan, “ An Enhanced Intelligent Intrusion Detection System using Machine Learning”, International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019, Vol.8, Issue. 9.
7. Nandini Rebello, and Manamohan K, “Network Intrusion Detection System using K-Means Clusterin and Gradient Boosted Tree Classifier”, International Journal of Engineering and Advanced Technology (IJEAT), 2019, Vol. 8, Issue-3S.
8. Ahmed I. Saleh, FatmaM. Talaat, and LabibM. Labib, “A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers”, Springer Science+Business Media, 2017.
9. Sasanka Potluri, and Christian Diedrich,” Accelerated Deep Neural Networks for Enhanced Intrusion Detection System”, IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), 2016, 6-9 Sept.
10. Pavel Kachurka, and Vladimir Golovko, “Neural Network Approach to Real-Time NetworkIntrusion Detection and Recognition”, IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 15-17 September 2011.
11. Sheraz Naseer, Yasir Saleem, Shehzad Khalid, Muhammad Khawar Bashir, Jihun Han, Muhammad Munwar Iqbal , and And Kijun Han, “Enhanced Network Anomaly Detection Based on Deep Neural Networks”,IEEE Access, 2018 (48231-48246).
12. S. Hochreiter, J, and Schmidhuber, “Long Short-Term Memory”, Neural Computation”, 9 (8), 1997, pp 1735-7380.
13. Wenke Lee, Salvatore J. Stolfo, and Kui W. Mok “Mining in a Data-flow Environment: Experience in Network Intrusion”, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-99) ,1999.
14. Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set”, Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defence Applications (CISDA), 2009.
15. Klaus Greff, Rupesh K. Srivastava, Jan Koutnik, Bas R. Steunebrink, and Jurgen Schmidhuber, “LSTM: A Search Space Odyssey”, IEEE Transactions on Neural Networks and Learning Systems,Vol (28), Issue: 10, Oct. 2017, p.p 2222 – 2232.
16. Jihyun Kim, Jaehyun Kim, Huong Le Thi Thu, and Howon Kim, “Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection”, IEEE International Conference on Platform Technology and Service (PlatCon), 2016.
17. Thi-Thu-Huong Le, Jihyun Kim, and Howon Kim, “An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization, IEEE, International Conference on Platform Technology and Service (PlatCon), 2017.
18. Chuanlong Yin , Yuefei Zhu, Jinlong Fei, and Xinzheng He, “A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks”, IEEE Access, Vol 5, p.p 21954 – 21961.
19. Huiwen Wang , Jie Gu , and Shanshan Wang, “An effective intrusion detection framework based on SVM with feature augmentation”, Knowledge-Based Systems,Vol.136,2017, pp130–139.
20. Raman Singh, Harish Kumar , and R.K. Singla, “An intrusion detection system using network traffic profiling and online sequential extreme learning machine”, Expert Systems With Applications, Vol 42(22),2015, p.p 8609-8624.

AUTHORS PROFILE

Mr. K. Narayana Rao, is full time research scholar under Visvesvaraya Ph.D scheme, supported by Ministry of Electronics & Information Technology (MeitY), Government of India in department Computer Science and Systems Engineering, Andhra University College of engineering (A), Andhra University. His main research work focuses on Intrusion detection using Deep Learning.

Prof. K. Venkata Rao, is a Professor in Computer Science and Systems Engineering Department, Andhra University College of Engineering (A), Andhra University. He is presently Dean Academic affairs, Andhra University. He has held the position of Honorary Director of Computer Centre and Web Master, Andhra University. His research areas include Image Processing, Big Data, Web technologies, Deep Learning, and other areas.

Prof. Prasad Reddy P.V.G.D, is a Sr. Professor in Computer Science and Systems Engineering Department, Andhra University College of Engineering (A), Andhra University. He is presentlyVice-Chancellor of Andhra University. He has held the position of Rector and Registrar of Andhra University, and Head of the Department of computer Science and Systems Engineering. His research areas include Machine Learning, Soft Computing, Software Architectures, Knowledge Discovery from Databases, Image Processing, Number theory & Cryptosystems and other areas.