Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 8, July, 2021:7184 – 7198

Research Article

Building a Novel Intrusion Detection System using Long Short-Term Memory for Detecting Network Attacks

Dr. M. Narayanan[1], Dr. T. Poongothai [2], Dr. N. Satheesh[3]

[1]Professor, Department of IT, Malla Reddy Engineering College for Women, Secunderabad, India.
[2] Professor, Department of CSE, St. Martin's Engineering College, Secunderabad, India
[3] Principal & Professor in CSE, St. Martin's Engineering College, Secunderabad, India
[1]jayarajinfoster@gmail.com, [2]poongothait@gmail.com, [3]principal@smec.ac.in

Abstract

With the increased usage of Internet and the advancement of technology the network traffic is heavy and brought major challenge to conventional security mechanisms. Th
e attackers also trying to launch sophisticated attacks to exploit potential vulnerabilities. The traditional intrusion detection system (IDS) is not able to handle massive data. Also, the data available is imbalanced which seriously affect the performance of classifier in IDS. In this paper, sampling technique is used to handle imbalance problem of data set. In addition, a novel Long Short-Term Memory (LSTM) based IDS is introduced to detect the attacks using NSL-KDD data set. The results demonstrates that the proposed system achieve detection accuracy of 92% for binary classification.

Keywords: Deep Learning, Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Intrusion Detection System, NSL-KDD Dataset, UNSW-NB15 Dataset.

1. Introduction

With the monumental growth of Information and Communications Technology (ICT) systems and networks, network security becomes a primary concern. Also, with the advent of the technology and proliferation of mobile devices, Internet is increasingly integrated with human's life. Now a days, Internet becomes inevitable part of our daily activities. Meanwhile, it is necessary to defend networks from cyber-attacks. The severity of attacks and threats are increasing with unprecedented rate. Therefore, the network intrusion detection plays a vital role in providing security to the network. Intrusion detection system enable to detect known and unknown attacks efficiently.

Various machine learning methodologies have been developed for identifying various network attacks. Most of the traditional machine learning models are not effective to deal classification problem with massive amount of data. Although the shallow learning methods are not suitable

for application containing high dimensional learning. Therefore, many intrusion detection systems were developed using deep learning techniques. Deep learning is able to get better representations from data. With the rapid development of big data and computational power, deep learning models have been widely used in various fields.

Different deep learning methods have been introduced for intrusion detection [22,23, 24]. But the temporal information in network traffic not received much attention. In the communication the activities are occurring in timely manner, therefore the maintenance of sequential information provides additional information. It is essential to develop models to consider the temporal nature of data. Recurrent Neural Networks and its variants LSTM and Gated Recurrent Units (GRUs) can capture temporal information and provides comprehensive analysis on network traffic data.

In this paper, a LSTM based intrusion detection system is proposed to deal with temporal data. This method evaluates the detection ability with various attacks of NSL-KDD data set and UNSW-NB15 dataset.

The rest of the paper is structured as follows. Section 2 briefly explains the works related to intrusion detection system using deep learning. Section 3 elaborates the RNN and LSTM. Section 4 present the details about LSTM system with performance evaluation metrics. Section 5 gives the experimental results and comparative analysis. Section 6 provides the conclusion and summary.

2. Related Works

Gwon et al.[1] proposed LSTM and feature embedding technique for intrusion detection. LSTM is used for capturing sequential information and feature embedding uses categorical features for recognizing malicious activities. The performance of the intrusion detection model was assessed using the UNSW-NB15 dataset. They compared the performance of various LSTM methods. The results shown that among all the models, the LSTM with feature embedding was the best.

For the anomaly-based intrusion detection system, Althubiti et al. [2] proposed an effective strategy that utilizes Long-Short-Term-Memory. This method uses CIDDS dataset for evaluating the performance of their model. CIDDS-001 dataset comprises 13 features. The intrusion detection only considers 10 features among 13 features and omits three features. This detection method applies LSTM with rmsprop optimizer to detect the abnormal behaviors. Finally, the findings of LSTM are compared to that of SVM, Nave Bayes, and MLP.

Hossain et al. [3,4] proposed an LSTM model with optimal hyper-parameter values for detecting DDoS attacks. They employed the CICIDS2017 dataset for experimenting the intrusion detection system. Slow-rate DoS, DDoS LOIT, BoT ARES, and port scanning are all detected by this intrusion detection technique. This method uses multiclass classification model to detect all these attacks. RMSprop optimizer delivers higher classification results for multiclass classification problems, according to the findings of the evaluation.

LSTM-based intrusion detection model was proposed by Xiao et al. [6]. KDD 99 Dataset and UNSW-NB15 Dataset were used for the experiments. Cross entropy loss function, softmax

activation function, and Adam as an optimizer were used in the LSTM model. They looked at how learning rate, epochs, and hidden layer number affected accuracy, precision, false positive rate, and false negative rate.

The LSTM model was proposed by Kim et al.[7] for a network intrusion detection system. They experimented with the learning rate and hidden layer size to see how well their model worked. The findings revealed that the size of the hidden layers has a significant influence on the detection rate. According to their findings, increasing the hidden layer size improves detection rates while lowering false alarm rates. Due to the lack of U2R instances during training, this approach is unable to detect U2R instances.

Thi-Thu-Huong Le et al.[8] built an intrusion detection model using LSTM with six different optimizer. They conducted experiments in two stages. In the first stage, suitable hyperparameter values are determined. In the second stage, the performance of LSTM was analyzed with six optimizers using hyperparameter values. The experiments were conducted with KDD Cup99 data set. The results revealed that the LSTM model with the Nadam optimizer and a learning rate of 0.002 has a higher detection rate and a lower false alarm rate.

Using Gated Recurrent Units (GRUs) and a multilayer perceptron, Xu et al [9] devised an IDS model. For assessing the performance of the intrusion detection model, the data sets KDDCup 99 and NSL-KDD are used. To scale the feature data in data preparation, Min-Max normalisation is utilised. The experiments are done with varying hyperparameter values. Different methodologies are compared in terms of performance metrics, accuracy, detection rate, and false positive rate. The results indicate that GRU with MLP outperforms other methods.

For detecting multistage attacks, Xu et al [10] introduced a detection approach using a multi-layer LSTM network. This model uses combined feature extraction scheme in feature extraction layer. In the first stage, the traffic from different stages is collected. In the feature extraction layer of this model, the combined feature extraction approach is used. The traffic from the various stages is gathered in the first stage. Then, to detect multi-stage attacks, time series features are gathered and examined. To avoid the overfitting problem, the LSTM model includes a dropout function. CTU-13 and NSL-KDD datasets are utilized to assess the performance of the LSTM model. The model's detection performance is investigated by changing the learning rate.

Ferrag et al. [11] investigated at a variety of deep learning approaches for detecting network intrusion. Using the CSE-CIC-IDS2018 dataset and the Bot-IoT dataset, they studied the various data sets and conducted a comparison assessment of several deep learning models. The data was divided into seven groups.

In an IDS system, Yin et al. [12] used a recurrent neural network for implementation. This method employs the supervised learning classification method. The NSL-KDD dataset was utilised to detect the intrusions in this model. The detection model's performance is evaluated

using three performance indicators: accuracy, false positive rate, and true positive rate. With a learning rate of 0.1 and hidden nodes of 80, high detection accuracy is achieved.

For intrusion detection, Vinayakumar et al.[15] evaluated Recurrent Neural Networks and their variations, Long-Short Term Memory (LSTM) and Gated Recurrent Units (GRU). All these models are used to distinguish between normal and attack events of KDDCup 99 dataset and UNSW-NB15 data set. Experiments are done using all the features and minimal feature sets of datasets. The neural network for this IDS model contains 41 neurons in the input layer, 32 hidden layers and 5 neurons in the output layer. In comparison to RNN and GRU, the results showed that LSTM has a good performance.

Poongothai et al.[18] suggested a deep auto encoder-based intrusion detection system that is both effective and intelligent. To classify normal and abnormal events, they created a deep auto encoder (DAE) based intrusion detection model. The IDS is compared against classic machine learning techniques including linear regression, nave bayes classifiers, KNN, decision tree, and random forest using the NSL-KDD data set. Classic algorithms are outperformed by the deep autoencoder.

To detect threats and block intrusions, Boukhalfa Alaeddine et al.[19] suggested a Network Intrusion Detection System (NIDS) based on Long Short-Term Memory (LSTM). The results of binary and multi-classification of the NSL-KDD dataset were obtained by this model. The results are compared to traditional machine learning classifiers. The assault categories U2R and R2L are integrated into one class in multiclass classification. The accuracy of binary classification is 99.98%, and four-class classification is 99.93%, according to the findings.
We use Long Short-Term Memory to create an intrusion detection model in this paper.

3. Recurrent Neural Networks and Long Short-Term Memory

Recurrent Neural Networks are suitable for handling sequential data. RNN consists of input unit, hidden units and output units. In RNN, the information flows only in one direction from input to output. Hidden units act as a storage unit, which remember the end-to-end information. Recurrent Neural Network works based on conventional feedforward neural networks. But RNN consists of cyclic connections to model the sequences. RNN can memorize the past inputs and capture the temporal information of data. RNN performs the same task for every sequence of input so it is called recurrent and the output is depending on the previous inputs.

In RNN for each hidden layer, the input set can be denotedas $\{x_0, x_1, \ldots\ldots x_{t-1}, x_t, x_{t+1}, \ldots\ldots\ldots\ldots\}$ and the output set as $\{h_0, h_1, \ldots\ldots h_{t-1}, h_t, h_{t+1}, \ldots\ldots\ldots\ldots\}$. U, W, W are weight matrices from the input layer to the hidden layer, the hidden layerto the output layer and inside the hidden layer, respectively. In an RNN, the hidden units plays a most role in completing the task. The figure 1 shows the structure of RNN.
The input sequence is given as

$$h_t = F_w(h_{t-1}, x_t)$$

$h_t = \tanh(Vh_{t-1} + Ux_t)$

$O_t = Wh_t$

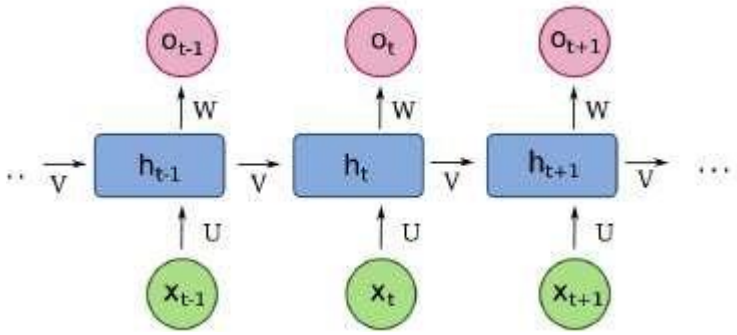

Figure 1: Structure of Recurrent Neural Network

Recurrent neural networks mainly used for classifying the sequence data. But RNN suffers from gradient vanishing problem. Due to the gradient vanishing problem, the RNN can retain information only short amount of time steps. RNN is not able capture the long-term dependencies. Therefore, the result of RNN won't be accurate. At each time step, the same weight is used for calculating the output. Therefore, LSTM is introduced to handle long-term dependencies [21].

LSTM is a special type of RNN [20]. LSTM is capable of handling long-term and short-term dependencies. LSTM treats the hidden layer as a memory [5]. LSTM solves the gradient vanishing problem with three gates. The memory cell includes three gates namely, input gate, forget gate and an output gate. LSTM uses sigmoid activation function and tanh activation function for selecting data. Figure 2 shows the structure of LSTM.

$F_t = \sigma (W_F x_t + U_F h_{t-1} + b_F)$

$I_t = \sigma (W_I x_t + U_F h_{t-1} + b_I)$

$O_t = \sigma(W_O x_t + U_O h_{t-1} + b_O)$

$c_t = Ft \odot ct\text{-}1 + I_t \odot \tanh (Wcx_t + Uch_{t-1} + bc)$

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

where x_t, h_t, and c_t are the input layer, hidden layer, and cell state at time t. Furthermore, b_I, b_F , b_C, and bo are bias at input gate, forget gate, cell state, and output gate, respectively. Furthermore, σ is sigmoid function, $\odot$ is used to represent element-wise multiplication. W is denoted by the Weigh Matrix. LSTM can capture the correlation between features and time series information in the long and short term using various gates.

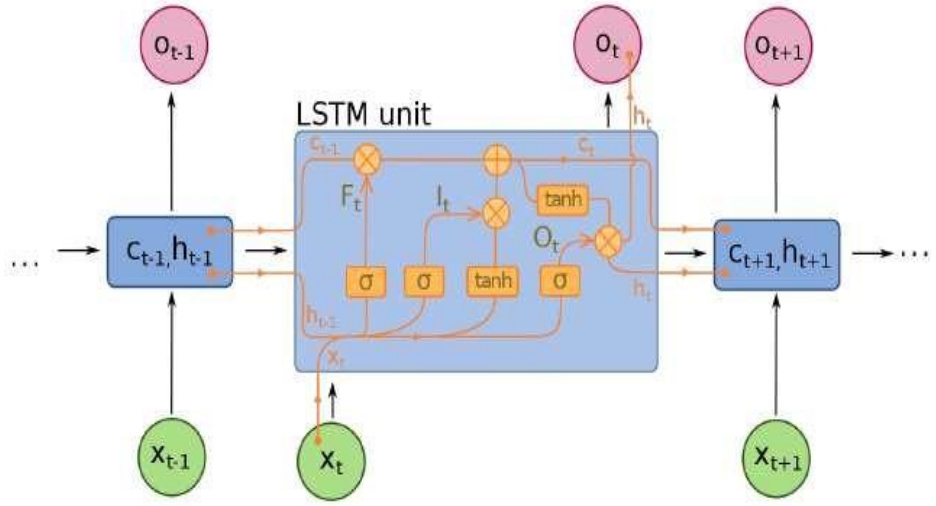

Figure 2: Structure of Long Short-Term Memory

4. Experimentation

Data Set

The intrusions are detected and the proposed model is evaluated using two datasets: the NSL KDD dataset and the UNSW-NB15 data set. Table 1 summarizes the attacks in NSL KDD dataset and table 2 provides the statistics of UNSW-NB15 data set.

The refined version of KDDCup99 intrusion data is NSL-KDD. It is a data set that may be used to compare the performance of various intrusion detection models [17, 25,26]. It is a commonly used data set for network intrusion detection that was created in 2009. Many researchers adopt NSL-KDD as a benchmark dataset to address issues with the KDD Cup 1999 dataset, which has too many redundant records. It excludes any records that are redundant or duplicate. It divides the KDD Cup dataset into different difficulty categories based on the number of learning algorithms that can successfully categorise the records. KDDTrain+, KDDTest+, and KDDTest–21 are three sub-files in the NSL-KDD dataset, with various normal records and four different types of attack records (see table). There are 42 features in the NSL-KDD data set. The KDDTrain+ dataset contains 125,973 network traffic samples, the KDDTest+ dataset contains 22,554 network traffic samples, and the KDDTest–21 dataset contains 11850 network traffic samples. These features are divided into basic features, content features, time based network traffic statistics features and host based network traffic statistics features.

UNSW-NB15 is a new dataset which reflects the complex and modern threat environment. It was created by IXIA PerfectStorm tool, Tcpdump tool, Argus tool, and Bro-IDS tool [13]. These tools are used for generating various types of attacks. The different types of attacks are DoS, Exploits, Generic, Fuzzers, Analysis, Backdoors, Reconnaissance, Shellcode, and Worms. This dataset is created with hybrid of normal and attack behaviours. The UNSW-NB15 dataset contains approximately two million and 540,044 vectors with 49 features. The 49th feature represents the category of event is an attack or normal which is denoted as either 0 or 1. The dataset is available in two forms; one is with general purpose records and another with connection records [27, 28].

Table 1: NSL KDD Data set

Attack Category	KDDTrain+	KDDTest+	KDDTest–21
Normal	67,343	9,710	2152
DoS	45,927	7,458	4342
Probe	11,656	2,422	2402
R2L	995	2,754	2754
U2R	52	200	200
Total	**125,973**	**22,544**	**11,850**

Table 2: UNSW-NB Data set

Features of UNSW-NB 15 dataset		16 hours
No._of_flows		987,627
Src_bytes		4,860,168,866
Des_bytes		44,743,560,943
Src_Pkts		41,168,425
Dst_pkts		53,402,915
Protocol types	TCP	771,488
	UDP	301,528
	ICMP	150
	Others	150
Label	Normal	1,064,987
	Attack	22,215
Unique	Src_ip	40
	Dst_ip	44

The experiments are done in Google Colaboratory using TensorFlow with Graphics Processing Unit (GPU). The methodology of the intrusion detection model is illustrated in the figure 3.

This model consists of the following stages: Data pre-processing, Feature Selection, Training and Testing.

Data pre-processing

Data features of the audit data are naturally inconsistent for training and testing process. Thus, the dataset need to be pre-processed before fed into the IDS classification model. The main purpose of the pre-processing is to transform the input audit data into suitable format for the training process. The data pre-processing stage consists of two steps namely, numeric transformation, normalization and data sampling.

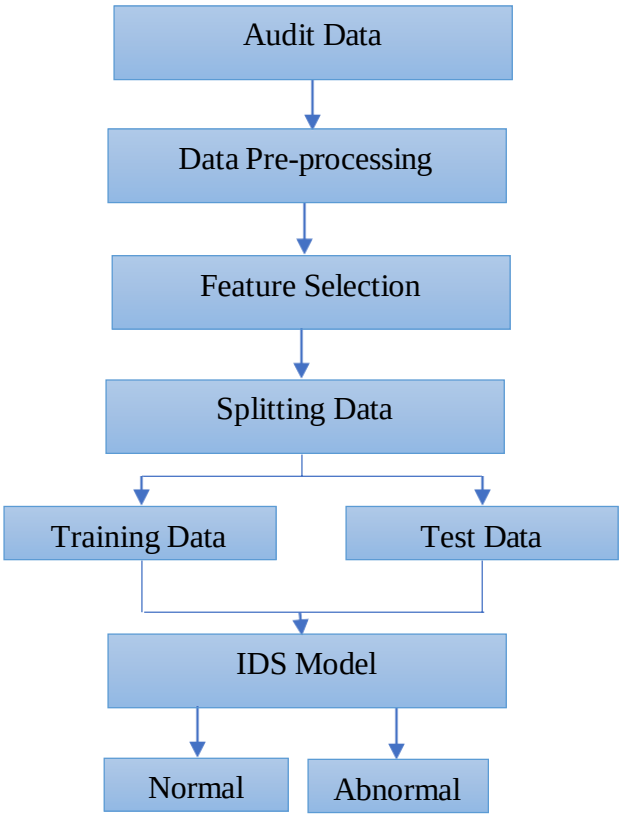

Figure 3 Intrusion Detection Model

In numeric transformation, non-numeric values are converted into numeric form. In NSL KDD data set, there are three non numeric features such as protocol_type, service and flag. These features are mapped into numerical values. The four attack categories (DoS, Probe, R2L and U2R) are represented as 1,2, 3 and 4 respectively. The normal category is represented with integer 0.

The values of feature are numerical and nonnumerical. Normalizing the feature values within the specified range. In this model, min-max normalization is used for representing the data [16]. The value of feature is normalized in the range of [0,1]

In general, NSL KDD data set is imbalanced due to the less number of R2L and U2R records. The number of R2L and U2R attack samples are less than 1% in the training set. As a result, the classification model biased towards the attacks with more records. The IDS model is not

able to detect the U2R and R2L records properly. In order to overcome this problem, an oversampling is applied on R2L and U2R attacks. In oversampling, the blocks of U2R and R2L are inserted randomly across data set. The result of oversampling gives dataset as a balanced one which is suitable for classification.

Feature Selection

The audit data may contain irrelevant, non-essential and insignificant features which have no effect on the result of the classification model. This will negatively impact the accuracy and increases the training and testing time. In NSL KDD data set, some of the features yield zero value. Those features can be removed and the size of the training set is reduced [14].

Training the Model

KDDTrain+ data set is used to train the model, and KDDTest+ and KDDTest–21 data sets are used to test it. Adam Optimizer is used for training and testing. Adam optimizer performs well in comparison with other optimizers. In the intrusion detection system, the Adam optimizer is suitable for the LSTM RNN model. [8, 21]. Selection of hyperparameters plays vital role in enhancing the accuracy of neural network model. Selection of learning rate helps in weight update. Increasing the number of hidden layers yields better results. The number of epochs represent the number of passes through the training data set. The weights will be updated at the end of each epoch. Batch size is the number of samples which is going to be used in order to train the network during its learning process. The table 4.3 shows the hyper parameters used in this model. The default values of Adam Optimizer are α = 0.002, $\beta 1$ = 0.9, $\beta 2$ = 0.999, and ϵ = 10-8

Table 3 Hyper parameters for the LSTM model

Name of the Hyperparameters	Value
Learning rate	0.01
Number of hidden layers	256
Number of epochs	100
Batch size	1024
Activation function	Sigmoid
Classification function	Softmax function

Detection algorithm based on LSTM

Algorithm: Detection algorithm
Input: A training set D X_i (i =1, 2, 3……., m)
Output: Detection result y_i
Step 1: Input the data to the forward layer of LSTM
Step 2: Define the Activation function.
Step 3: Calculate the loss function

Step 4: Update the weight function
Step 5: Get detection results.

5. Results and Discussion

To evaluate the performance of the intrusion detection model, the following performance metrics are calculated using the confusion matrix. The confusion matrix represents the actual and predicted class classifications. This is mainly used for binary classification. There are four values available in confusion matrix. True positive is denoted by p represents the number of anomaly records that are correctly identified as anomaly. False negative is denoted by q represents the number of anomaly records that are incorrectly identified as normal. False positive is denoted by r represents the number of normal records that are incorrectly identified as anomaly. True negative is denoted by *s* represents the number of normal records that are correctly identified as normal. These values are used to calculate the following metrics namely, Accuracy, Detection rate, Precision, Recall, False alarm and F-Score.

Accuracy of the IDS model is expressed as

$$Accuracy = (p + s) / (p + q + r + s)$$

The remaining metrics are calculated using the following formulae

$$Precision = p / (p + r)$$

$$Recall = p / (p + q)$$

$$False\ Alarm\ Rate = r / (r + s)$$

$$F\text{-}Score = 2p / (2p + q + r)$$

The recall, also known as True Positive Rate or Detection Rate, is defined as the ratio of successfully recognised attack connection records to the total number of attack connection records found. The fraction of benign occurrences wrongly categorised as harmful is measured by the false alarm rate, also known as the false positive rate. The calculation of the F-score is mostly used to assess the performance of imbalanced classification issues. The F-Score is a single statistic that combines precision and recall. The harmonic mean of precision and recall is calculated using the F-Score.
Mathew Correlation Coefficient (MCC) is another important measure to evaluate the performance of binary classifier [14]. The formula for calculating MCC is given below.

$$MCC = (p * s) - (q * r)/sqrt ((p + r) (p + q) (s + p) (s+ q))$$

The value of MCC ranges between -1 and 1. When the classifier is accurate the value of MCC is 1, indicating that the classification is correct. When the classifier is not accurate the value of MCC is -1 indicating that the classification is wrong.

The performance of the intrusion detection model is measured using NSL-KDD dataset with binary classification and multiclass classification. The IDS model is tested with four learning rates 0.1, 0.05, 0.01 and 0.005 respectively. The hidden layer size is varied with 16, 32, 64, 128 and 256. The different number of epochs are 0, 20, 40, 60, 80 and 100. The accuracy of the intrusion detection system is measured with changing the hidden layer size, learning rate and number of epochs. To evaluate the performance of LSTM based intrusion detection model two methods of analysis performed. One is binary classification and another is five class classification based on NSL-KDD data set.

Classification performance for Binary Classification

In binary classification the records are classified into two categories either normal or attack. The figure shows the confusion matrix for the binary classification. The performance metrics for binary classification is shown in table 4

Table 4 Confusion matrix of NSL-KDD (KDDTest+) dataset for binary classification

Predicted class / Actual class	Attack	Normal
Attack	10295	2308
Normal	828	8883

Classification performance for Multiclass Classification

The Table 5 shows the confusion matrix for the multiclass classification. In multiclass classification, the attack data set is divided in to five categories namely, Normal, DoS, U2R, R2L and Probe. The accuracy of multiclass classification is deteriorated compared with binary classification. The DoS attack is a kind of flooding attack. In this attack the target object is denied for receiving and sending the packets. The detection of this attack is relatively easy by carefully examining the connection packets. Probe attack is trying to gather information from the network. This attack steals the information from the network in an unauthorized manner.

In U2R attack, the attacker trying to gain the access permission of root user by bypassing the authentication mechanism. They will do series of illegal operations with privileged account. The detection of this attack is difficult due to the unavailability of the adequate training data.

In R2L attack, the attacker is trying to gain local access to a remote computer. The frequency of occurrence of this attack is very less and the accuracy of the detection system also less due to the less sample available in training data. The performance metrics for each category is shown in table. The results indicates that the accuracy of DoS attack is better comparing with the remaining attacks.

Table 5 Confusion matrix of NSL-KDD (KDDTest+) dataset for multilabel classification

Predicted class / Actual class	Normal	DoS	U2R	R2L	Probe
Normal	9164	142	4	12	264
DoS	854	6258	11	0	195
U2R	1024	0	236	0	87
R2L	167	0	18	28	124
Probe	158	204	0	5	2031

Evaluation Metrics for Binary Classification and multiclass classification on NSL-KDD (KDDTest+) dataset is shown in table 6.

Table 6 Evaluation metrics for binary and multiclass classification

Classification	**Category**	**Precision (%)**	**Recall (%)**	**Accuracy (%)**	**False Alarm Rate (%)**	**F-Score (%)**	**MCC (%)**
Binary Classification	Attack	94.36	84.26	92	8.25	88.41	84.44
Multiclass Classification	DoS	93.25	91.23	91.45	2.16	89.78	88.24
	U2R	75.58	78.95	79.62	1.65	68.36	75.84
	R2L	78.64	80.12	78.64	2.94	72.61	76.37
	Probe	79.68	81.75	80.88	4.98	78.67	77.81

The performance of the proposed system is evaluated using NSL-KDD data set. Our model achieved a detection accuracy of 92% for binary classification.

The results for the UNSW-NB15 dataset are presented in the table 7.

Table 7 Results for UNSW-NB15 dataset

Category	**Precision (%)**	**Recall (%)**	**Accuracy (%)**	**False Alarm Rate (%)**	**F-Score (%)**	**MCC (%)**
Normal	81.61	85.26	82.85	17.5	84.75	83.51
Generic	**88.25**	**94.52**	**96.93**	7.88	**95.76**	**97.35**
Exploits	85.61	89.29	91.02	14.71	88.95	91.98
Fuzzers	62.71	66.07	69.31	15.78	62.11	68.84
DoS	65.73	75.46	73.84	21.47	75.23	72.52
Reconnaissance	67.29	78.92	76.68	19.77	76.21	76.35
Analysis	10.53	22.84	23.58	15.8	24.29	22.59

Backdoor	7.16	11.54	10.56	7.56	11.78	11.81
Shellcode	38.12	49.82	51.32	**0**	52.49	44.25
Worms	9.85	12.47	11.11	1.27	11.96	11.87

6. Conclusion

In this paper, a novel intrusion detection model is proposed based on LSTM for network security. The detection accuracy of LSTM based intrusion detection was improved significantly for both binary classification and multiclass classification. It is very difficult to detect the U2R and R2L attacks due to imbalanced nature of data set. The proposed algorithm gives better results for U2R and R2L attacks by synthesizing the U2R and R2L attacks, thereby data samples of R2L and U2R attacks are increased and quality of data set is improved.

References

1. Gwon Hyeokmin, C. Lee, RakunKeum and Heeyoul Choi. “Network Intrusion Detection based on LSTM and Feature Embedding”,2019.
2. S. A. Althubiti, E. M. Jones and K. Roy, "LSTM for Anomaly-Based Network Intrusion Detection," 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, 2018, pp. 1-3, doi: 10.1109/ATNAC.2018.8615300.
3. M. D. Hossain, H. Ochiai, D. Fall and Y. Kadobayashi, "LSTM-based Network Attack Detection: Performance Comparison by Hyper-parameter Values Tuning," 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), New York, NY, USA, 2020, pp. 62-69, doi: 10.1109/CSCloud-EdgeCom49738.2020.00020.
4. M. D. Hossain, H. Ochiai, F. Doudou and Y. Kadobayashi, "SSH and FTP brute-force Attacks Detection in Computer Networks: LSTM and Machine Learning Approaches," *2020 5th* International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, 2020, pp. 491-497, doi: 10.1109/ICCCS49078.2020.9118459.
5. Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9, no. 8 (1997): 1735-1780.
6. S. Xiao, J. An and W. Fan, "Constructing an Intrusion Detection Model based on Long Short-term Neural Networks," 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), Singapore, 2018, pp. 355-360, doi: 10.1109/ICIS.2018.8466445.
7. J. Kim, J. Kim, H. L. Thi 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, doi: 10.1109/PlatCon.2016.7456805.

8. T. Le, J. Kim and H. Kim, "An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization," International Conference on Platform Technology and Service (PlatCon), Busan, 2017, pp. 1-6, doi: 10.1109/PlatCon.2017.7883684.
9. C. Xu, J. Shen, X. Du and F. Zhang, "An Intrusion Detection System Using a Deep Neural Network with Gated Recurrent Units," in *IEEE Access*, vol. 6, pp. 48697-48707, 2018, doi: 10.1109/ACCESS.2018.2867564.
10. M. Xu, X. Li, J. Ma, C. Zhong and W. Yang, "Detection of Multi-Stage Attacks Based on Multi-Layer Long and Short-Term Memory Network," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, pp. 1-6, doi: 10.1109/ICC.2019.8761487.
11. Ferrag, Mohamed Amine &Maglaras, Leandros&Moschoyiannis, Sotiris &Janicke, Helge, "Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study" Journal of Information Security and Applications. Vol.50, 2020. 10.1016/j.jisa.2019.102419.
12. 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, pp. 21954-21961, 2017, doi: 10.1109/ACCESS.2017.2762418.
13. N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, 2015, pp. 1-6, doi: 10.1109/MilCIS.2015.7348942.
14. Almi'ani, Muder & Abu Ghazleh, Alia & Al-rahayfeh, Amer & Atiewi, Saleh & Razaque, Abdul, "Deep Recurrent Neural Network For IoT Intrusion Detection System", Simulation Modelling Practice and Theory, Vol.101,2020.
15. R, Vinayakumar & KP, Soman & Poornachandran, Prabaharan, "Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS)", International Journal of Information System Modeling and Design. Vol.8. pp.43-63, 2017.doi:10.4018/IJISMD.2017070103.
16. H. Hou et al., "Hierarchical Long Short-Term Memory Network for Cyberattack Detection," in IEEE Access, vol. 8, pp. 90907-90913, 2020, doi: 10.1109/ACCESS.2020.2983953.
17. S. K. Sahu, S. Sarangi and S. K. Jena, "A detail analysis on intrusion detection datasets," 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, 2014, pp. 1348-1353, doi: 10.1109/IAdCC.2014.6779523.
18. T. Poongothai, K. Jayarajan, P.Udayakumar. "An Effective and Intelligent Intrusion Detection System using Deep Auto-Encoders". International Journal of Advanced Science and Technology, 29(9s), pp.3139 – 3154, 2020.
19. Alaeddine Boukhalfa, Abderrahim Abdellaoui, Nabil Hmina, Habiba Chaoui, LSTM deep learning method for network intrusion detection system. International Journal of Electrical and Computer Engineering. Vol.10, No.3, pp.3315-3322, 2020.

20. Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," in *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68-75, 3 2017, doi: 10.1049/iet-its.2016.0208.
21. S. Althubiti, W. Nick, J. Mason, X. Yuan and A. Esterline, "Applying Long Short-Term Memory Recurrent Neural Network for Intrusion Detection," *SoutheastCon 2018*, St. Petersburg, FL, 2018, pp. 1-5, doi: 10.1109/SECON.2018.8478898.
22. K. Wu, Z. Chen, and W. Li, A novel intrusion detection model for a massive network using convolutional neural networks, IEEE Access, vol. 6, pp. 50850_50859, 2018.
23. S. Z. Lin, Y. Shi, and Z. Xue, Character-level intrusion detection based on convolutional neural networks, in Proc. Int. Joint Conf. Neural Netw.(IJCNN), Jul. 2018, pp. 1-8.
24. Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks," Appl. Sci., vol. 9, no. 2, p. 238, 2019.
25. UNB. NSL-KDD Dataset. Available online: https://www.unb.ca/cic/datasets/nsl.html
26. Dhanabal, L.; Shantharajah, S. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. Int. J. Adv. Res. Comput. Commun. Eng. 2015, 4, 446–452.
27. ACCS. UNSW-NB15 Dataset. Available online: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/ cybersecurity/ADFA-NB15-Datasets/
28. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 IEEE Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6.