Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Detecting Intrusion In Internet Of Things (Iot) With Deep Learning Bi-Directional Recurrent Neural Network (BRNN)

Thai Son Chu[1], Mahfuz Ashraf[2]

[1]Lecturer, Lincoln Institute of Higher Education
[2]Dean, Lincoln Institute of Higher Education
[1] jason.chu@lincolnau.nsw.edu.au,
[2] dean@lincolnau.nsw.edu.au

Distributed Denial of Service (DDoS) attacks could lead to enormous problems, which raises important security issues in many organizations and corporations. By using Bi-directional Long Short-Term Memory Recurrent Neural Network (BLSTM RNN), this paper presents a novel deep learning technique for detecting DDoS attacks within the networks. To achieve highly effective intrusion detection models, we first train our deep learning neural network with multiple layers using a benchmark data set, CICDDOS2019. We examine the binary classification of normal and attack patterns on the web. The result of the experiment shows that our proposed model achieves excellent outcomes in precision, recall, F-1 score, and FPR. Our proposed BLSTM RNN model can detect attacks with over 98% accuracy on practical data sets, better than other Kumar models (31) and even our previous model (30).

Index Terms—Deep Learning, DDoS, Bi-directional Long Short-Term Memory, Recurrent Neural Network.

I. INTRODUCTION

An Intrusion Detection System (IDS) is a system that monitors network traffic and can identify any hostile activity or misuse [1]. An IDS is technically equivalent to a classification task, i.e. determining whether the network traffic is "normal" or "anomalous". A classification problem can be of two types: binary classification and multiclass classification. The system generates only two outputs in binary classification: "attack" and "normal". On the other hand, a multiclass classification also identifies attack types. This paper aims to employ binary classification for intrusion detection.

In general, to detect DDoS attacks, an integrated and cooperative Network Intrusion Detection System (NIDS) infrastructure is deployed at both the front end and the back end of each processing server to identify intrusions [2]. All NIDS installed on servers collaborate to update their signature databases by receiving signals from the central log. This enables correlation in the central registry, allowing for the detection of previously unknown attacks.

The cognitive module in this design classifies an attack by detecting intrusions using the misuse detection database and Snort. Snort attempts to identify the nature of the attack and transmits this information to the Alert System, which then rejects the packet. This method enables researchers to readily refresh the database for exploitation without modifying the existing regulations.

According to Tama and Rhee [5], most current ML algorithms rely on shallow learning and cannot correctly solve the intrusion classification problem in a real-time context due to the vast amount of data involved. Deep learning techniques have the potential to mine or extract enhanced representations from the data and can extract significantly more efficient features due to the rapid evolution of various datasets [3]. Deep learning was created in 2006 by G. Hinton et al. [4], and it has experienced a meteoric ascent in computational analytics over the years. This paper proposed a deep learning model to detect DDoS attacks that are based on the Bidirectional Long Short-Term Memory and Recurrent Neural Networks (BLSTM RNN) approach described by Graves and Schmidhuber [12]. It is the latest and one of the most popular deep-learning techniques to tackle this attack issue. This technique also means automating the discovery of abstractions from the raw data set.

Our paper discusses designing and implementing a BLSTM RNN network attack detection model. Accuracy, precision, true positive rate, and F1-score are used to evaluate the model's performance in binary classification. With the CICDDOS2019 dataset, simulations will be conducted. The experimental results demonstrate that our suggested BLSTM RNN model effectively detects five types of security assaults that a vulnerable network may encounter. This paper has main contributions as follows:

- We utilized BLSTM RNN for DDoS detection with high accuracy and quick convergence detection speeds.
- We utilized the systematic approach to propose this model with recent and comprehensive dataset.
- We conducted an evaluation on our model that detected DDoS attack with high accuracy and precision.

The following sections of this work are organized as follows. Section II consists of similar work about attack detection. Section III addresses the details of the proposed model for DDoS attack detection, including the CICDDOS2019 benchmark dataset, data preprocessing technique, and the evaluation matrix. Section IV focuses on the results of experiments and discussion. The final section provides the conclusion of this research.

II. RELATED WORK

A recent work by Tama and Rhee [5] proposes a DNN methodology nearby. Instead of employing outdated datasets, i.e. KDDCup 99 and NSL-KDD, the authors have evaluated the performance of DNN on new networking- related benchmark datasets CICDDOS2019 and UNSW- NB15. The study also reports biased results in the UNSW- NB15 dataset due to a

data imbalance issue. The distribution of one class in the UNSW-NB15 dataset is completely lower than the supplementary class [5]. The study could not also observe the performance differences between DNN and other algorithms. In this study, we have chosen the CICDDOS2019 benchmark dataset for evaluating our proposed BLSTM RNN model for detecting intrusions in the vulnerable network.

In recent years, deep learning has developed with lightning speed and is being utilized for detecting intrusions and outperforming conventional methods, such as in [6] and [7]. In [6], a deep learning approach has been applied using DNN for flow-based anomaly recognition. The investigational outcome reveals that the proposed technique could be used for anomaly detection in software- defined networks. In [7], the authors used a self-taught- learning (STL) algorithm using the standard NSL-KDD dataset. In relation to the performance of former studies, the approach proved to be more efficient. However, the group of studies emphasizes the capability of feature- reduction of DL techniques. It primarily employs the DL approach to pre-train the network, and the classification is then performed over the regular observation model. Nevertheless, it is rare to smear the DL technique for performing classification precisely. That remains a study deficit regarding deep learning performance in multiclass classification.

Fu et al. [8] introduce a novel intrusion detection technique for anomaly mining based on client- server network architecture. In this study, the authors show that anomalies are detectable by analyzing the patterns of the data of the perception layer, like temperate, humidity or anything that a networking object sensor could collect and report. The study uses an unsupervised algorithm for data mining in order to identify normal patterns. The Intel Lab Project dataset was used to evaluate the proposed system's effectiveness. Unfortunately, no accuracy was reported in Fu et al.'s work [8].

M. Sheikhan et al. [9] conducted an experiment claiming that RNNs are viewed as reduced-size RNNs. This model introduces a 3-layer RNN architecture having 41 input features and four attack classes as outputs for a misuse-based intrusion detection system. Besides, the RNN units of layers remain partly connected. As a result, the proposed RNNs never exhibit the capability of DL to prototype high-dimensional features. Moreover, the paper needs to exhibit the performance evaluation of the approach proposed in terms of binary classification.

In this research, we propose a deep learning approach to detect DDoS attacks in the vulnerable network by using bidirectional LSTM (BLSTM) recurrent neural networks. Related to former works, we have used the BLSTM-based model to classify and exclude pre-training. Also, we have used the CICDDOS2019 dataset, which has a distinct training set and testing set for evaluating the efficiency of the introduced model.

III. PROPOSED MODEL

The proposed model aims to detect the DDoS attack in data exchanged by vulnerable network servers at the transport layer of the vulnerable network. To do so, the model requires a process

design, which would accept data of network traffic as input, then process the input data and generate a two-fold classification: “Attack” or “Pass”.

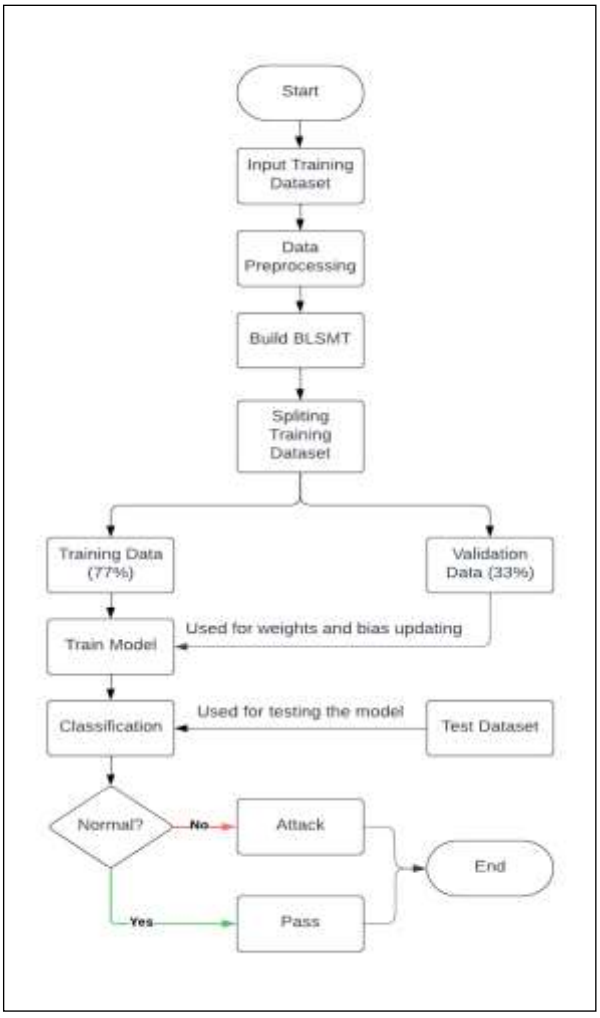

Fig. 1. Proposed model to detect DDoS with BLSTM RN

Figure 1 illustrates the detection in further depths. The sample dataset goes into the preprocessing stage where the data is encoded and normalized and fits into a data structure processable by the BLSTM RNN model. After preprocessing, the whole test data is split into two sets: Test set and validation set. The test set is used for training the BLSTM RNN network, and the validation set is used for validating the already trained network. After the completion of the validation process a separate testing dataset is used for testing the model classification accuracy and other performance measures.

A. Recurrent Neural Network Long Short-Term Memory

A Recurrent Neural Network (RNN) is a multilayer network that contains feedback loops and conveys information from the past to the present. The RNN's hidden layers store information like computer memory. RNNs constitute potent DNNs that use internal memory and loops to process sequence data [20].

Long Short-Term Memory (LSTM) extends Recurrent Neural Networks (RNNs). LSTM units incorporate the concept of gates. As to the vanishing gradient issue, RNNs are unable to learn context information over an extended period (i.e., the time between an input is received and when it is used to produce a prediction). Consequently, RNNs cannot learn from long-distance dependencies [21]. The usage of an LSTM architecture is one solution to this issue [21]. It eliminates the problem of the vanishing gradient and enables the retention of the extended duration of context data.

B. Bi-directional Recurrent Neural Network

Bidirectional LSTM (BLSTM) is derived from bidirectional RNN (BRNN) [10], which employs two distinct hidden layers to process the sequences of input from forward and backwards orientations. Figure 2 depicts a bidirectional LSTM with three consecutive time steps.

BLSTMs connect both hidden levels to a single output layer. Traditional RNNs are limited in that they can only utilize the past context of the input data stream. BLSTMs [11] rectify this issue by distributing information in forward and reverse orientations

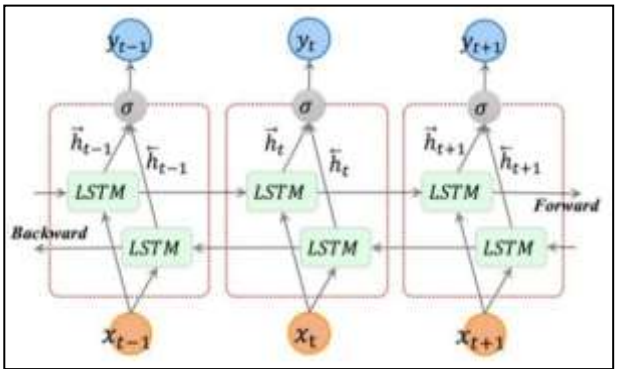

Fig. 2. Bi-directional LSTM architecture with 3 consecutive time steps.

The concept of BLSTM originates from BRNN processing the sequence data in both frontward and backward directions and using two distinct hidden layers. BLSTMs connect two hidden layers to the one output layer. One inadequacy of traditional RNNs is that they are only capable of using the previous context. BRNNs fix this by dispensing data in both directions. A BRNN calculates the forward hidden sequence $\vec{h}$, the backward hidden sequence $\overleftarrow{h}$ and the output sequence **y** by iterating the backward layer from t = T to 1, the forward layer from t =1 to T and then updating the output layer by the following equations:

$$\overrightarrow{h_t} = \mathcal{H}(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t+1} + b_{\overrightarrow{h}})$$
$$\overleftarrow{h_t} = \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}})$$
$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y$$

The BLSTM layer produces an output vector, ***YT***, which is calculated by the equation:

$$y_t = \sigma(\vec{h}_t, \overleftarrow{h}_t)$$

where σ function combines both the output sequences. The σ function could be of four kinds: summation, average concatenating, and multiplication function.

Merging BRNNs with LSTM evolves bidirectional LSTM [12], which can access long-range context in both input directions. Unlike the LSTM network, the forward and backward iterations of each layer in the BLSTM network are performed like the traditional neural networks. The combination of both forward and backward layers is considered as a single BLSTM layer. It has been shown that bidirectional models are better than regular unidirectional models in various fields, such as speech recognition and phoneme classification [13].

C. Data Processing

Moustafa and Slay [14] criticized that the NSL-KDD dataset and KDD'99 dataset did not characterize the up-to- date interventions in DDoS attacks, and they presented a comprehensive and all-inclusive dataset called the CICDDOS2019. This dataset encompassed several features from the KDD'99 dataset [15]. They further inspected the features of the CICDDOS2019 dataset and the KDD'99 dataset. Their study showed that the former KDD'99 dataset is less effective than UNSWNB15 features [15].

In this research, the dataset employed was CICDDOS2019 from the Canadian Institute for Cybersecurity, which is the most current and comprehensive dataset available. Besides the data of the latest DDoS attacks that are like accurate data, the dataset CICDDoS2019 also includes network traffic analysis results [19]. The analysis of network traffic utilizes CICFlowMeter-V3 and has labelled traffic justified to attack in CSV files, originator and target IPs, originator and target ports, protocols and time stamping. The CICDDOS2019 dataset is the most recent and effective dataset published for DDoS attack detection research work purposes. Furthermore, the entire dataset can be split into training and test samples for research studies. We use the 'CICDDOS2019_training-set.csv' and 'CICDDOS2019_test-set.csv' datasets for training and testing, respectively.

As a preliminary experiment, reduced dataset samples are selected randomly from the entire training set. In our study, we only evaluate the attributes proposed in [15]., The training dataset is manually processed by utilizing the method of Fu et al. [8], where we manually add some anomalous samples to the dataset to make it suitable for their research purposes. The

input dataset for intrusion detection would support the research objective. Further, the method showed the issue of expensively acquiring DDoS datasets labelled for intrusion detection. We manually retrieved the characteristics and attack kinds, following the manual manipulation approach. Thus, our resultant dataset has five features and two class labels: "Attack" and "Normal." Table I depicts the data set's structure. The first five columns denote the extracted features, the sixth column is the attack category, and the final column is the binary label.

TABLE I. DATASET STRUCTURE AFTER PRE-PROCESSING

Feature	**Description**
Fwd Packet Length Max	Maximum packet size in the forward (outgoing) direction
Fwd Packet Length Min	Minimum packet size in the forward direction
Min Packet Length	Minimum length of a packet
Max Packet Length	Maximum length of a packet
Average Packet Size	Average size of a packet
FWD Packets	Number of forward packets per second
Fwd Header Length	Header length of a forwarded packet
Fwd Header Length 1	Number of bytes in a header in the forward direction
Min_Seg_Size_Forward	Minimum segment size in the forward direction
Total Length of Fwd Packet	Packet size in the forward direction
Fwd Packet Length Std	Standard deviation of a packet in the forward direction
Flow IAT Min	Minimum time between two packets in the flow
Subflow Fwd Bytes	Average number of bytes in a sub flow in the forward direction
Destination Port	Address to receive TCP or UDP packets
Protocol	TCP or UDP for data transmission
Packet Length Std	Standard deviation of the packet length
Flow Duration	Duration of the flow in µs
Fwd IAT Total	Total time between two packets in the forward direction
ACK Flag	Count Number of packets with ACK
Init_Win_Bytes_Forward	Number of bytes in initial window in the forward direction
Flow IAT Mean	Mean time between two packets in the flow
Flow IAT Max	Maximum time between two packets in the flow
Fwd IAT Mean	Mean time between two packets in the forward direction
Fwd IAT Max	Maximum time between two packets in the forward direction

D. Evaluation Matrix

We have used a confusion matrix to describe how well our BLSTM RNN model works. The confusion matrix is a 2x2 dimensional matrix representing the relationship between the predicted and actual value, as shown in fig 3.

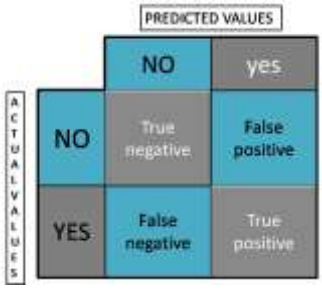

Fig. 3. Confusion Matrix

Accuracy: It defines the percentage of correct classification. Accuracy can be calculated by using the formula shown in Eq 1:

$$\frac{TP + TN}{x}$$

Where X is the total number of records.

Misclassification rate: Determines the percentage of wrong predictions. It can be calculated by using the formulae shown in Eq 2:

$$\frac{FP + FN}{x}$$

False Positive Rate (FPR): the percentage at which the system incorrectly rejects, as shown in (3):

$$\frac{FP}{number\ of\ actual\ normal\ records\ in\ x}$$

We also evaluate our model through precision, recall and f1-score values.

Precision is the ratio of correct positive predictions to the total positive predictions, as shown in (4):

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

Recall is the ratio of correct positive predictions to the number of actual positive records, as shown in (5):

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-Score is the harmonic mean of Precision and Recall, as shown in (6)

$$F1 - 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

These metrics are used to evaluate the proposed intrusion detection model.

IV. IMPLEMENTATION

We implemented the model in Spyder (a Python-based scientific interactive programming environment) using the TensorFlow library [24]. The implementation procedure is comprised of three essential phases: data preprocessing, BLSTM model training, and testing.

A. TensorFlow

Google released TensorFlow, which is an open-source software library for deep learning, in November 2015 [24]. The main purpose of TensorFlow is to define, train, and deploy machine learning algorithms. TensorFlow is a structure for machine learning that can adapt to various circumstances. It employs dataflow graphs and maps the graph's nodes and vertices across multiple computers using graphics processing units, multicore central processing units, and Tensor processing units. The architecture provides application developers with a responsive and adaptable platform, allowing them to experiment with distinct training algorithms and optimizations. TensorFlow supports several applications that concentrate on the training and implementation of deep learning neural networks and is widely used in ML research [23]. Its flexible dataflow representation enables expert users to achieve exceptional performance.

B. Phase of Data Preprocessing

The implementation phase begins with the preprocessing of data. This phase entails reading and storing the entire training dataset in memory. Next, feature extraction is carried out. After encoding the dependent variables, the data are normalized (feature scaling). To process the features, a TensorFlow data structure must be constructed to store the features and labels. Since an RNN is being utilized, the data must be transformed into time steps. Reshaping is the concluding stage of the data preprocessing procedure.

C. Phase of Training

Training is the second phase of implementation. Initially, the BLSTM RNN model was constructed using the Keras toolkit. Next is model compilation, followed by model training. Here, the file CICDDOS2019 training-set 5451.csv (reduced training-set) is divided into two subsets: 67% of the file will be used for training, while 33% will be used for the validation. The training subset is used to train the model, while the validation subset is used to evaluate the model's performance after each epoch. By training the model, we evaluate its performance and then continue training it after modifying its parameters until it functions adequately.

D. Testing phase

This is the phase of our system where we load the test dataset. It will be fed to our trained model for testing purposes. Then we record the evaluation matrix, which reflects the performance of our system.

V. RESULTS AND DISCUSSION

In this study, we have employed the Keras deep learning framework [27] and the Google TensorFlow library. The experiment is conducted on a Dell Inspiron 15 notebook with an Intel Core i7-8650U processor at 1.90 GHz, 16 GB of memory, and no GPU acceleration. It is a binary classification problem, which means that the classifier must decide whether each sample is "normal" or "attack." We utilized classification evaluation metrics, including accuracy, precision, recall, F-1 score, true positive rate, false positive rate, and error rate.

The network was trained with 5451 samples and tested with 1799 samples, which is 33% of the total number of training samples. The training samples are extracted from the file CICDDOS2019_training-set.csv. The classifier is then evaluated using 3,688 test samples. These test samples have been extracted from the file CICDDOS2019 testing-set.csv. The number of test samples used for classification is displayed in Table II.

Table III shows four values of the performance in the confusion matrix: True Positive, True Negative, False Positive and False Negative. Table IV summarizes the outcomes of the experiment. The proposed model can detect attacks using the sample of CICDDOS2019 dataset with 98% accuracy and 99% precision. The model generates zero false alarm rates and a very low incorrect detection rate of 0.04%, with an impressive recall and F1- score value of 98%. The proposed model could classify 4205 samples of data in 2.19 seconds on an Intel Core i7 1.9GHz Central Processing Unit (CPU).

TABLE II. NUMBER OF SAMPLES USED TO CLASSIFY

Class	Sample size
Attack	4094
Normal	112

TABLE III. RESULTS OF SIMULATION OF 4 PERFORMANCE VALUES IN CONFUSION MATRIX

Parameter	Number of Samples
TP	5027
TN	1
FP	10
FN	166

TABLE IV. COMPARATION ON ACCURACY, PRECISION, RECALL AND F1- SCORE TO OUR PREVIOUS MACHINE LEARNING MODEL

Model	Accuracy	Precision	Recall	F1 - score
ML RF [30]	0.97	0.98	0.96	0.99
DL LSTM [31]	0.98	0.98	0.97	0.97
DL BLSTM RNN	0.98	0.99	0.96	0.99

This model demonstrates that the BLSTM RNN algorithm implemented is able to detect DDoS attack with higher accuracy and precision percentage than the previously ML experiment [30]. The comparison of this model to a recent deep learning long short-term memory model [31] shows the greatest accuracy 98%. The accuracy measure of three models is shown in Table IV.

VI. CONCLUSION AND FUTURE WORK

This paper shows a new model for detecting DDoS attacks based on the BLSTM RNN. The BLSTM RNN can do deep learning well and use the training dataset to learn detailed features. We use the Keras framework for deep learning and the Google TensorFlow library to build the proposed new model. The implemented BLSTM was applied to a subset of the CICDDOS2019 dataset, which had previously been used in a number of works on DDoS attack networks. The detection was based on binary classification, distinguishing between normal and dangerous patterns. We tested our proposed model against the CICDDOS2019 dataset demonstrating over 98% accuracy. It also shows an impressive performance in terms of true/false positive rates, precision and recall values.

For future advancements, additional experiments will be conducted to analyze the proposed BLSTM RNN model using large data sets from published data sets, particularly data sets containing the specific attack traffic. Also, the developed model will be improved to make it better at detecting and to make it easier to choose between different detection parameters.

REFERENCES

[1] S. Santosh Kumar, B. Vignesh, B. Vignesh And S. Rajarajan, 2021, Intrusion Detection System (Ids) Using Deep-Learning, International Journal Of Engineering Research Technology (Ijert), Icradl 2021 (Volume 09 – Issue 05).

[2] Z Chiba, N.Abghour, K.Moussaid, A.El Omri, And M.Rida, A Cooperative And Hybrid-Network Intrusion Detection Framework In Cloud Computing That Based On Snort & Optimized Back Propagation Neural Network," Procedia Comput. Sci., Vol. 83, 2016, Doi: 10.1016/J.Procs.2016.04.249.

[3] C. Yin, Y. Zhu, J. Fei, And X. He, Deep Learning Approach For Intrusion Detection System Using Recurrent Neural Networks, Vol. 5, 2017.

[4] Y. Lecun, Y. Bengio And G. Hinton, Deep Learning, Nature, Vol. 521, No. 7553, Pp. 436_444, May 2015.

[5] B. A. Tama And K. H. Rhee, "Attack Classification Analysis Of The Network Via Deep Learning Approach," Research Briefs On Information & Communication Technology Evolution (Rebicte), 2018. Doi: 10.22667/Rebicte.2017.11.15.015.

[6] A. Javaid, W.Sun, Q. Niyaz, And M. Alam, A Deep Learning Approach In Network Intrusion Detection System, 9th Eai International Conference Bio-Inspired Information Communication Technology (Bionetics), New York, Ny, Usa, May 2016, Pp. 21-26.

[7] T. A. Tang, D. Mclenon, L. Mhamdi, S. A. R. Zaidi And M. Ghogho, A Deep Learning Approach For Network Intrusion Detection In Software-Defined Networking," In Proc. Int. Conf. Wireless Netw. Mobile Commun. (Wincom), Oct. 2016, Pp. 258-263.

[8] Fu Et Al., "An Intrusion Detection Scheme Based On Anomaly Mining In The Internet Of Things", In Ieee International Conference On Wireless, Mobile & Multimedia Networks

(Icwmmn 2011), Beijing, 2011, Pp. 315-320. Doi: 10.1049/Cp.2011.1014.

[9] Bajaj And Arora, "Improving The Intrusion Detection Using Discriminative Machine Learning Approach And Improve The Time Complexity By Data Mining Feature Selection Methods", International Journal Of Computer Applications (0975-8887), Volume 76-No.1, August 2013.

[10] M. Schuster And K. K. Paliwal, "Bidirectional Recurrent Neural Networks," Ieee Transactions On Signal Processing, Vol. 45, No. 11, Pp. 2673–2681, 1997.

[11] Mike Schuster And Kuldip K Paliwal, "Bi-Directional Recurrent Neural Networks," Signal Processing, Ieee Transactions On, Vol. 45, No. 11, Pp. 2673–2681, 1997.

[12] A. Graves And J. Schmidhuber, Framewise Phoneme Classification With Bi-Directional Lstm And Other Neural Network Architectures, Neural Networks, Vol. 18, No. 5, Pp. 602–610, 2005.

[13] Z. Cui, S. Member, R. Ke, S. Member, And Y. Wang, "Deep Stacked Bidirectional And Unidirectional Lstm Recurrent Neural Network For Network-Wide Traffic Speed Prediction," Pp. 1–12, 2018.

[14] K. Peffers, T. Tuunanen, M. Rothenberger, And S. Chatterjee, Design Science Research Methodology In Information Systems Research, Journal Of Management Information Systems, Vol. 24, No. 3, Pp. 45-77, 2007.

[15] T. Janarthanan And S. Zargari, "Feature Selection In Cicddos2019 And Kddcup'99 Datasets," 2017 Ieee 26th International Symposium In Industrial Electronics (Isie), Edinburgh, 2017, Pp. 1881-1886. Doi: 10.1109/Isie.2017.8001537

[16] A. Javaid, Q. Niyaz,W. Sun, And M. Alam, Deep Learning Approach For An Intrusion Detection System, The 9th Eai Int. Conference Bio- Inspired Inf. Communication Technol. (Bionetics), New York, Ny, Usa, May 2016, Pp. 21-26.

[17] Z. Dewa And L. A. Maglaras, "Data Mining And Intrusion Detection Systems," Vol. 7, No. 1, Pp. 62–71, 2016.

[18] M. Elkhodr, S. Shahrestani, And H. Cheung, "The Internet Of Things: New Interoperability, Management And Security Challenges," Vol. 8, No. 2, Pp. 85–102, 2016.

[19] Z. Cui, S. Member, R. Ke, S. Member, And Y. Wang, "Deep Stacked Bidirectional And Unidirectional Lstm Recurrent Neural Network For Network-Wide Traffic Speed Prediction," Pp. 1–12, 2018.

[20] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, And K. Takeda, "Bidirectional Lstm-Hmm Hybrid System For Polyphonic Sound Event Detection Mitsubishi Electric

[21] B. A. Tama And K. H. Rhee, "Attack Classification Analysis Of The Network Via Deep Learning Approach," Research Briefs On Information & Communication Technology Evolution (Rebicte), 2018. Doi: 10.22667/Rebicte.2017.11.15.015.

[22] P. Barham Et Al., "Tensorflow: A System For Large Scale Machine Learning," Pp. 265–284.

[23] N. Buduma, Tensorflow For Deep Learning—Implementing Neural Networks. Usa: O'reilly Media, Inc., 2016.

[24] V. Timčenko And S. Gajin, "Machine Learning Based Network Anomaly Detection For The Environment."

[25] G. Li And Z. Yan, "Data Fusion For Network Intrusion Detection: A Review," 2018.

[26] H. A. Abdul-Ghani, D. Konstantas, And M. Mahyoub, "A Comprehensive The Attacks Survey Based On A Building-Blocked Reference Model," Vol. 9, No. 3, 2018.

[27] M. Abomhara And G. M. Køien, Cyber Security And Internet Of Things (Iot): Vulnerabilities, Threats, Intruders, Vol. 4, Pp. 65–88, 2015.

[28] D. Kumar, R.K., Pateriya, R. K. Gupta, V. Dehalwar, A. Sharma, "Ddos Detection Using Deep Learning", 2023 Procedia Computer Science, Vol. 218, Pp. 2420-2429, 2023.

[29] T. S. Chu, W. Si, S. Simoff And Q. V. Nguyen, "A Machine Learning Classification Model Using Random Forest For Detecting Ddos Attacks," 2022 International Symposium On Networks, Computers And Communications (Isncc), 2022, Pp. 1-7, Doi: 10.1109/Isncc55209.2022.9851797.2022

[30] Deepak Kumar, R.K. Pateriya, Rajeev Kumar Gupta, Vasudev Dehalwar, And Ashutosh Sharma. 2023. Ddos Detection Using Deep Learning. Procedia Comput. Sci. 218, C (2023), 2420–2429. Https://Doi.Org/10.1016/J.Procs.2023.0