Detecting Distributed Denial of Service in Network Traffic with Deep Learning

Muhammad Rusyaidi[1], Sardar Jaf[2]
Faculty of Technology, School of Computer Science
Sunderland University, UK, Sunderland
SR6 0DD, United Kingdom

Zunaidi Ibrahim[3]
Mechanical Engineering, Universiti Teknologi Brunei
Tungku Highway, Gadong BE1410
Brunei Darussalam

***Abstract*—COVID-19 has altered the way businesses throughout the world perceive cyber security. It resulted in a series of unique cyber-crime-related conditions that impacted society and business. Distributed Denial of Service (DDoS) has dramatically increased in recent year. Automated detection of this type of attack is essential to protect business assets. In this research, we demonstrate the use of different deep learning algorithms to accurately detect DDoS attacks. We show the effectiveness of Long Short-Term Memory (LSTM) algorithms to detect DDoS attacks in computer networks with high accuracy. The LSTM algorithms have been trained and tested on the widely used NSL-KDD dataset. We empirically demonstrate our proposed model achieving high accuracy (~97.37%). We also show the effectiveness of our model in detecting 22 different types of attacks.**

Keywords*—Cybersecurity; Cyber-attack; DDoS attack; machine learning; deep learning; recurrent neural networks; long short-term memory

I. Introduction

COVID-19 pandemic has caused a great deal of fear, worry, and a significant shift in our way of life. Organizations have had to adapt to the requirement for remote working on a large scale and rapidly. Because COVID19 has produced or expanded applications and use cases of digital technologies, this pandemic is proven to be a motivator for digital transformation. Despite the pandemic, the world can still interconnect with each other through a network with rapid development in IoT4.0 technologies. This involves millions of data bytes being produced, processed, converted, exchanged, or shared and utilized to produce an outcome in specific applications. This involves the security elements to protect sensitive data and the privacy of each individual user of cyberspace or network. Distributed Denial of service (DDoS) is a type of attack in which the victim's resources are depleted, rendering them unable to handle valid requests. Nonetheless, the number of DDoS attacks and the amount of DDoS traffic are increasing, requiring more research into the detection of such security risks. Therefore, the use of machine learning to ensure the intensity of this data is very important. In this study, we examine network traffic behaviors for cyber detection by the application of various machine learning algorithms to improve the accuracy of DDoS attack detection.

DDoS attacks are common network exploitation type of cyber-attack. The attacker creates network exhaustion to legitimate users by causing a computer or network system to crash, stopping them from accessing server or the Internet, either temporarily or continuously. According to Singh et al. [1], the DDoS attack is one of the most common and major cyber-attacks. Ray et al. [2] also states that more advanced technology is needed to improve DDoS attack detection in computer networks. Since detecting DDoS attacks is a difficult task before any mitigation measures can be performed, cybersecurity fundamentals are required to design a system that can detect threats. DDoS attacks were initially detected by traffic engineers using rule-based approach. This strategy have fallen behind the dynamic and evolving nature of DDoS attacks. Academics and industry are researching the prospect of integrating machine learning into DDoS detection process because of their immense potential and success in various Computing domains. Threats can be recorded more rapidly and correctly with machine learning algorithms, such as Naïve Baysian, K-Nearest Neighbor, Random Forest and Recurrent Neural Network.

In this study, we focus on exploring the effectiveness of deep learning algorithms to improve the accuracy of DDoS attack detection in order to better analyze network traffic activities for cyber threat detection.; Also, we aim to discover a feature selection strategy that, when combined with a machine learning system, can improve DDoS detection accuracy rates. Our selected deep learning algorithm is based on a Recurrent Neural Network (RNN) classifier to distinguish between normal and attack traffic.

The remaining of this paper is organized as follow: Section II describes the literature review; Section III describes our methodology. The results from our experiments are presented in Section IV and we discuss our finding in Section V. We compare our results with previous research in Section VI. In Sections VII and VIII we conclude the paper and outline our future work, respectively.

II. Literature Review

Recent research has demonstrated the effectiveness of machine learning application in detecting DDoS attacks. In this section, Sambangi et al. [3] developed a machine learning model to predict DDoS and botnet attacks by using machine learning algorithm with multiple linear regression. They used the most widely used CICIDS 2017 benchmark dataset with entire packet payloads in pcap format, which is extensively used in labeled network flows. They also demonstrated that their machine learning model could detect DDoS attacks using

the regression analysis technique. Yuan et al. [4] showed that Recurrent Neural Network surpasses Random Forest in terms of generalization. the effectiveness of deep learning, where reduced the error rate from 7.517% to 2.103% using an ISCX2012 dataset, compared to traditional machine learning methods. They experiment uses the ISCX2012 dataset, made available by the University of New Brunswick in 2012. Guerre- Manzanares et al. [5] proposed the concept of employing hybrid feature selection models to lower the size of the feature to achieve more accurate results. The dataset contained 115 features. To limit the number of features, the filter; wrapper; and hybrid models were used for choosing the potential feature. These features were then loaded into a K-Nearest Neighbor (KNN) and Random Forest model, both of which had a high accuracy of 99%.

Sabeel et al. [6] presented the idea of using two deep learning models (deep neural network and long short-term memory) for binary prediction of unknown Denial of Service (DoS) and DDoS attacks. The models were evaluated on the benchmark CICIDS2017 dataset. According to Sabeel et al. [6], the models fail to detect unknown threats accurately. However, after retraining the deep learning models by merging newly synthesized datasets with the old ones, the True Positive Rate (TPR) achieved was 99.8% and 99.9% for DNN and LSTM, respectively. Rusyaidi et al. [7] demonstrated deep learning effectiveness in DDoS detection. Elsayed et al. [8] demonstrate that combining RNN with an autoencoder allows input traffic to be classified into two categories: normal and malicious. Elsayed et al. [8] and Catak et al. [9] have used Deep learning to deal with a high degree of complex nonlinear interactions. They were making it a possible tool for identifying network attacks. By using 70% of the input data for training, Elsayed et al [8]. model showed the best results when compared to existing traditional machine learning techniques, resulting in 99% accuracy in their proposed method.

The experiment conducted by Gadze et al. [10], they used RNN and LSTM in the software-defined networking (SDN) controller to identify and mitigate DDoS attacks. It was gathering certain network parameters when operating in a normal and also when subjected to DDoS attack. The number of packets received and transferred at each switch, the packet count (number of packets per flow), the protocol type (TCP, UDP, or ICMP), the Source IP, and the Destination IP were some of the main features. They looked at three different possibilities. In the first case, 80% of the data was used for training, while 20% was used for testing. In the second instance, 70% of the data was used for training and 30% for testing. In the third situation, 60% of the data was used for training and 40% for testing. RNN and were used for detecting and mitigating DDoS attacks. The 70/30 (train–test ratio) split yields improved model accuracy compared to the 80/20 and 60/40 split ratios.

Ugwu et. al. [11] compared the results of traditional machine learning algorithms such as Naive Bayes (NB), Decision Tree (DT), and Support Vector Machine (SVM). The suggested LSTM and Singular Value Decomposition (SVD): deep learning algorithms demonstrate a significant improvement. Data pre- processing is performed on the network data, which includes data normalization and feature conversion methods. The normalization method requires limiting network feature values to a narrow range of values and feature conversion method requires transforming non-numeric feature values to numeric. Kasim [12] used dimensional reduction features in the autoencoder (AE) model and Support Vector Machine (SVM) classifier to classify encoded data as DDoS or normal. AE-SVM successfully distinguishes between normal and DDoS attack traffic. The min-max method was used to normalize their data between 0 and 1, and the training vectors for the AE model were created. With the encoding process, the trained model delivered feature learning and feature reduction. The results showed that the AE-SVM method performed well in terms of low false-positive DDoS detection rates and fast anomaly detection.

Gormez et al. [13] demonstrate that by using traditional machine learning algorithms, ensemble, and deep feature extraction methods, Bayesian optimization is faster than traditional grid search optimization. However, it requires more computing resources than the train-test step. The scikit-library of Python is used to implement the experiment classification methods. Network traffic packet data was captured and converted into connection records. They used three types of features: basic features, time-based features, and connection-based features. Basic features are characteristics that can be easily derived from packet headers by counting specific packet properties for the connection. Before evaluating a model's performance, hyper-parameter optimization allows researchers to fine-tune its hyper-parameters. The Bayesian optimization process is used to create samples of hyper-parameter values to locate the optimums.

Hossain et al. [14] highlighted that one of the most important criteria in evaluating the performance of network attack detection systems is the availability of labeled dataset. According to their experimental data, the optimal hyper-parameter combinations were used for constructing their robust intrusion detection system. LSTM multiclass classification was used in the experiment, with 80% of the dataset used for training and 20% for testing. Hyper-parameter adjustment was also used to investigate the performance. The experiment results demonstrated that deep learning models (LSTM) have become emerging technology for network attack detection systems.

III. Methodology

A. Dataset Preparation and Pre-Processing

The NSL-KDD dataset from the University of New Brunswick Lab, which includes 125,973 network packets with 22 different types of attacks, as shown in Table I.

According to Tang et al. [15], one of the most up-to-date datasets for Intrusion Detection System (IDS) evaluation is the NSL-KDD dataset. There are 41 features in this dataset, divided into three categories: fundamental, content-based, and traffic- based features. DoS, probe, U2R, and R2L are the four types of attacks. We use the NSL-KDDTrain+ dataset train our DDoS detection system, while the NSL-KDDTest+ dataset is used to test it. As a result, the NSL-KDDTest+ dataset is a useful indicator of a model's zero-day attack resistance. To distinguish between genuine and malicious traffic, the use of

DoS as a basis is utilized. Table II summarizes the dataset's features. These features are not ordered on a scale. These attributes are further passing to the next phase for normalization.

A dataset may have missing values, irrelevant features, categorical data, or other flaws that prevent a machine learning algorithm from analyzing it. In some cases, standardization, data normalization, and other issues might prevail in some circumstances. The NSL-KDDTrain+ customized datasets have missing values, irrelevant features, and an issue with the categorical column. The following data cleaning and preparation processes were included in this project and will be discussed in the paragraph below. The selected dataset contains 4,898,431 data records.

There are a few rows/columns in the customized dataset that do not have a number (NaN) or have infinite values. In the NSL-KDD dataset, not all values are filled, and some have strings. Research made by Nimbalkar et al. [17] shows that the captured network traffic is unsuitable for machine learning models due to noise, which includes NaN and missing data. These settings must be fixed before any further operations can be performed. To address the problem of NaN values, a variety of approaches can be used, as stated in Nimbalkar et al. [17]. One method involves removing rows or columns with a particular number of NaN values, while the other approaches involve replacing a missing value with another value, such as the mean, median, mode, or other statistical measures of a column, a row, or a group of data. The selection must be made wisely based on the information available about the dataset. We are replacing the NaN values with mean and median at the features in this project since there are some features that have missing values. Some columns do not include the information needed to classify traffic as normal or malicious. As a result, constant columns are useless for any detection process. In the dataset, there is one attribute, num_outbound_cmds, which is always 0 for all rows in the training and test data. We remove this attribute because it could otherwise result in performance degradation and unnecessary complications. Therefore, for algorithms that demand numerous samples of one or more-time steps and features, we reshaped two-dimensional data where each row represents a sequence of three-dimensional array.

TABLE I. 22 DIFFERENT TYPES OF ATTACKS ALZAHRANI ET AL. [16]

Attack Categories	Training Set Attack Names	Test Set Attack Names
DoS	Back, land, Neptune, pod, smurf, teardrop	Back, land, Neptune, pod, smurf, teardrop, (mailbomb), process table, udpstorm, apache2, worm
Probe	Ipsweep, nmap, portsweep, satan	Ipsweep, nmap, portsweep, satan, mscan, saint
U2R	Buffer overflow, load module, perl, rootkit	Buffer overflow, load module, perl, rootkit, sqlattack, xterm, pst
R2L	ftp-write, guess-passwd, imap, multihop, phd, spy, warezmaster	ftp-write, guess-passwd, imap, multihop, phf, spy, warezmaster, xlock, xsnoop, snmpguess, snmpgetattack, HTTP tunnel, send-mail, named, warez client

TABLE II. FEATURE OF NSL-KDD DATASET

1	Duration
2	Protocol_type
3	Service
4	Flag
5	Src_bytes
6	Dst_bytes
7	Land
8	Wrong_fragment
9	urgent
10	Hot
11	Num_failed_logins
12	Logged_in
13	Num_compromised
14	Root_shell
15	Su_attempted
16	Num_root
17	Num_file_creations
18	Num_shells
19	Num_access_files
20	Num_oubound_cmds
21	Is_host_login
22	Is_guest_login
23	Count
24	Srv_count
25	Serror_rate
26	Srv_serror_rate
27	Rerror_rate
28	Srv_rerror_rate
29	Same_srv_rate
30	Diff_srv_rate
31	Srv_diff_host_rate
32	Dst_host_count
33	Dst_host_srv_counts
34	Dst_host_same_srv_rate
35	Dst_host_diff_srv_rate
36	Dst_host_same_src_port_rate
37	Dst_host_srv_diff_host_rate
38	Dst_ host_serror_rate
39	Dst_host_srv_serror_rate
40	Dst_host_rerror_rate
41	Dst_ host_srv_rerror_rate

Each sequence has several time steps, each with one observation which is a feature. There are enough data records in the NSL-KDD dataset for training and testing. The availability of data records and the lack of redundant records, which can prevent false detection of DDoS attack, help in improved learning accuracy. After the dataset had been cleaned and pre-processed, we trained and tested LSTM and RNN algorithms. The dataset is split to train and test sets. These sets are necessary for training the estimator and subsequently evaluating the performance of the associated model. In this project, we use NSL-KDDTrain+ for training and NLS-KDDTest+ for testing. However, a common practice is to split for training and testing machine learning algorithms, as has been done by Rusyaidi et al. [7] and Gadze et al. [10].

Creating a model for classification or other similar tasks is at the basis of machine learning-based work. This is what the training phase accomplishes. The training dataset, produced before in the data split phase, is used to train a machine learning algorithm on a section of the whole dataset. An algorithm that has been trained produces a model that has learned from the data. There are a variety of classification estimators available. In this work, RNN and LSTM algorithms were used. These estimators were chosen for their ease of use, widespread use in the literature, and solid performance in related work by Yuan et al. [4], Elsayed et al. [8] and Gadze et al. [10].

B. Deep Learning Model Development

Traditional feedforward neural networks have the problem of assuming data to be unrelated. The feedback loops of the hidden units are the major difference between a RNN and a feedforward neural network. RNNs can process a sequence of inputs and save their state while processing the next sequence of inputs in deep learning. The essential information is stored in the node's memory and will be used for learning in future time steps as shown in Nazih et al. [18]. However, RNN has some issues remembering long-term memories as stated in Staudemeyer et al. [19]. Thus, it does not work well with long sequences. As a result, problems with RNNs such as vanishing gradient and short-term memory, can be solved using a type of RNN known as Long Short-Term Memory Networks (LSTM).

According to Laghrissi et al. [20], LSTM is a Recurrent Neural Network that can recall more context information than RNN and select what information is significant and not important by using distinct cell states. Different gates and a cell state are included in the LSTM. Althubiti et al. [21] explain that LSTM has a sigmoid function that produces numbers between 0 and 1. If the activation function's value is 0, the information is lost; if the value is 1, the information is saved. The input gate changes the state of the cell. The previous hidden state and the current input are sent into the input gate. The tan and sigmoid activation functions are included, as well as their multiplied values. The cell state is now computed by adding the output of the input gate point by point. Finally, the output gates determine the value of the next concealed state.

The LSTM algorithm overcomes the limitation of RNNs by learning long-term dependencies. Another distinction is that, whereas RNNs have only one neural network layer.

LSTM has four neural network layers that interact with each other. In this project, the input and embedding layers are used first, followed by one LSTM layer with dense layers as depicted in Fig. 1. Note that mean absolute error is used as loss function, Adam as optimizer function, Accuracy as performance metrics.

LSTM is particularly suited for data sequence applications because of its unique design. Fig. 1 shows the model looks up the embedding for each character, converts two-dimensional data into a three-dimensional array, executes the LSTM batch size, timestep, and LSTM units with the embedding as input, then applies the dense layer to generate result accuracy prediction results. Many previous research Laghrissi et al. [20], Althubiti et al. [21], Gadze et al. [10], and Sabeel et al. [6] indicated that LSTM is an effective approach for learning long-term dependencies and efficiently representing the relationship between current occurrences and historical events. In this paper, we adopted LSTM for the design of our deep learning architecture.

C. Training and Testing Proposed LSTM – RNN Model

Feature selection is a key issue in machine learning projects. Guerra-Manzanares et al. [5] highlighted that one aspect of dimensionality reduction is feature selection. Not all the features in a dataset are equally essential for detecting the attack. In many cases, increasing the number of characteristics above a particular threshold has no discernible effect on classification performance. It simply adds to the complexity and delays in performance. Not only that, but it may also lead to overfitting and a decline in classification performance. As a result, we look for a minimal number of features that can appropriately identify the traffic in a dataset wherever possible. For this, we use the wrapper technique, namely correlation feature selection, which is supported in Scikit-learn.

The easiest strategy to train a model is to train the specific attack types to avoid being attacked by the same sort of attack. The 22 various forms of attacks (as shown in Table I) were utilized for training the model to reinforce it. The attributes of each attack have distinct values. These features and attack types were part of the training set, which was 80% of the full NSL- KDDTrain+ dataset. We use 20% of the NSL-KDDTest+ database for testing our model. The test set is separate from the training set.

D. Proposed Model Architecture

Fig. 2 shows the main components of the proposed system: the pre-processing, adaptive attribute selection, and classification of DDoS attack type. The procedure of subsystems is divided into three stages:

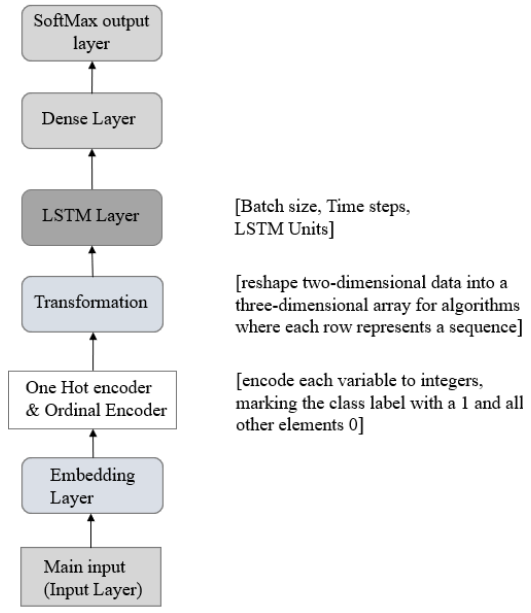

Fig. 1. The Proposed System Architecture representing each Layer.

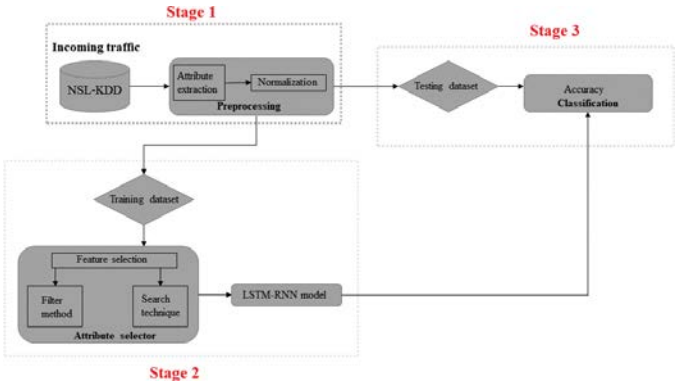

Fig. 2. The Main Components of the Proposed System Architecture with Representing the Processing Stage.

Stage 1: The Pre-processing stage involves collecting and normalizing attributes from network traffic. Data is separated into subgroups for training and testing. 80% of the data in NSL-KDDTrain+ is set as a training dataset for use in attribute selector (Stages 2), while the remaining 20% of the data in NSL-KDDTest+ is set to test dataset for use in Stage 3.

Stage 2: Various automatic threshold procedures are used in the Attribute Selection Subsystem to determine the minimum number of attributes.

Stage 3: classification and detection of DDoS attacks.

In order to have a more practical structure of the results, all of the experiments were organized into three stages, as shown in Fig. 2. Two experiments were carried out in Stage 1. These tests were conducted using estimators set to their default settings. No extra parameter tuning or feature selection work was done here; instead, a basic percent split technique was applied. A series of feature selection experiments were carried out in Stage 2. Once again, a percent split technique was applied without taking cross-validation into account. In Stage 2, multiple experiments comprising a feature selection operation were carried out.

Finally, the classification is in charge of detecting traffic data as DDoS in Stage 3. The results of stages 1, 2, and 3 were successfully achieved. The proposed machine learning model improved the DDOS attack detection approaches and increased the DDOS detection accuracy with a combination of features selection, adam optimizer, mean absolute error, oneHotEncoder strategy. Hence, there are test accuracy results after being implemented in the module. In the sections below, discussions are included that go along with it.

IV. Experimental Results

We have implemented our deep learning architecture in TensorFlow with Keras backend. We used the mean absolute error approach to verify the model's loss while learning the deep neural network, which comprises two hidden layers. The "Adam" optimization function was used. "one-hot encoder" and "Ordinal Encoder" libraries from the "Sklearn" library also have been used to convert order-like values to numeric numbers. We trained the model with 150 epochs with a batch size of 44.

The training accuracy of the model used the sample loss and accuracy using a batch size of 44, as shown in Fig. 5. We have experimented with varying learning rates, but the selective learning rate of 0.013 produced the best result in our experiment. After 150 epochs, where the training and validation performance converged, the model achieved the highest training accuracy of 98.21% and 0.0211 error rate. Fig. 3 shows the training and validation loss, and Fig. 4 shows the training validation accuracy.

Fig. 5 illustrates training accuracy and loss value of the model during train phrase. We have tested the model on test data, the model performance tested on the test set produced 97.37% accuracy as shown in Fig. 6.

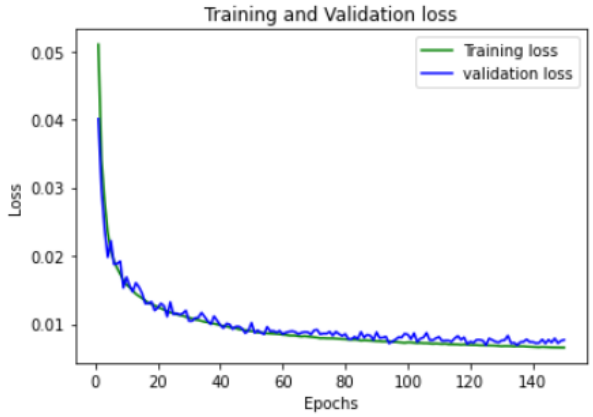

Fig. 3. Training and Validation Loss over 150 epochs.

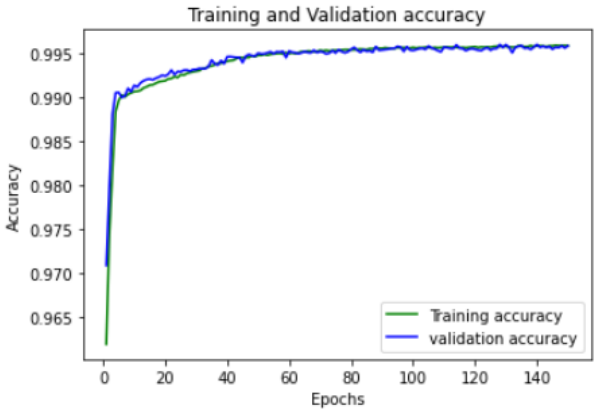

Fig. 4. Training and Validation Accuracy Graph over 150 epochs.

```
125973/125973 - 98s - loss: 0.0339 - accuracy: 0.9743
125973/125973 - 94s - loss: 0.0234 - accuracy: 0.9811
125973/125973 - 92s - loss: 0.0219 - accuracy: 0.9817
125973/125973 - 92s - loss: 0.0211 - accuracy: 0.9821
```

Fig. 5. Training Performance of our Model.

```
1656/1656 [==============================]
value Loss:  0.030921630561351776
value Accuracy:  0.9736678004264832
```

Fig. 6. Test Performance of our Model.

V. Result

The NSL-KDD dataset, on the other hand, was utilized to test this approach. The proposed machine learning-based categorization solution for DDoS attacks has high accuracy in testing. From the results of the trained model, it was observed the proposed model's accuracy is 98.21%, which is almost a perfect method to prevent and protect the 22 different types of attacks, including DDoS attacks. Moreover, the LSTM evaluation model generates a 97.37% accuracy in the test set, the algorithm fits the patterns of the dataset with a 97.37% accuracy.

VI. Comparison of Related Work

Table III illustrates the comparison results of the accuracy between the proposed LSTM model with various deep learning methods using the same NSL-KDD dataset. Alkahtani et al. [22] conducted an experiment in which they chose the essential network features. To detect the anomaly in cybersecurity threats, these features were analyzed by classifying algorithms. SVM and KNN algorithms and deep learning based on the LSTM-RNN model were used to develop machine learning models. When compared to the KNN and LSTM, the SVM method produces better results. In the KDD Cup '99 and NSL-KDD datasets, the SVM method performed better than the LSTM-RNN and KNN methods. Their deep learning technique, based on the LSTM-RNN algorithm, had a high accuracy of 93.55%, but it couldn't surpass SVM's performance. Furthermore, they split the data into 70/30 train-test ratios in their experiment, while the proposed model utilized an 80/20 train-test ratio. As a result, the 80/20 split ratio produces better model accuracy than the 70/30 split ratio used in the LSTM-RNN algorithms.

TABLE III. Accuracy Comparison for Various Deep Learning Technique with Proposed Model using NSL-KDD Dataset

Authors	**Technique**	**Accuracy testing model (%)**
Alkahtani *et. al.* [22]	LSTM-RNN. Support Vector Machine (SVM). K-Nearest Neighbor (K-NN).	93.55 (LSTM RNN) 96.53 (SVM) 87.65 (KNN)
Tang *et. al.* [15]	Gated Recurrent Unit Recurrent Neural Network (GRU-RNN)	89.00
Niyaz *et. al.* [23]	Self-taught Learning (STL), a deep learning-based technique	88.39
Ugwu *et. al.* [11]	LSTM + SVD	90.59
Proposed model	LSTM-RNN	97.37

The "Adam" optimizer for DNN optimization was utilized in our study, which reduces the loss and optimizes the model. To transform order-like values to numeric numbers, the researcher uses the "OneHotEncoder" and "OrdinalEncoder" libraries. Despite their excellent performance, our proposed model has achieved better outcomes with a testing accuracy model of 97.37% and a loss value of 0.0287 from 52977 sample test packets. This evaluation reveals that the LSTM is an effective solution for preventing and protecting against 22 different sorts of attacks.

The accuracy of our proposed method against the other approaches is significantly different in these comparisons. In the NSL-KDD dataset, our LSTM-RNN beats models that utilize all 41 features for training and testing. When compared to previously implemented deep learning methods in Alkahtani et al. [22], Tang et al. [15], Niyaz et al. [23], Ugwu et al. [11], the proposed model of LSTM-RNN did very well on the evaluation of the test data. This comparison demonstrates how our method's clear phases are predictable, accurate, effective, and authoritative.

Tang et al. [15] claim that when using a GRU-RNN method, their Deep Recurrent Neural Network (DNN) methodology obtained an accuracy of 88.39%. They use a Nadam optimizer and a mean squared error (MSE) model in their experiment. Our proposed model was developed using the Adam optimizer, which is the best optimizer. According to Kandel et al. [24], each optimizer is compared differently depending on the architecture, and the Adam optimizer has the best performance on the dataset in evaluation. They also used the mean square error (MSE) method to remove and appreciate the average error. Mean absolute error (MAE) was utilized in the proposed model. As a consequence, MSE outperformed the MSE technique in terms of interpretation.

Niyaz et al. [23] use NIDS based on sparse autoencoder and soft-max regression. As a result, they claim that the NSL-KDD dataset's Normal and anomaly (2-class) classification yielded an accuracy of 88.39%. By monitoring their method, autoencoder trains to effectively represent a manifold on which the training data resides. It was done by utilizing the mean square error (MSE) approach, which does not show an average error.

Ugwu et al. [11] designed the LSTM and SVD deep learning methods to show considerable improvement. They pre- processed the network data by converting features and normalizing the data. Non-numeric feature values were converted to numeric values using the feature conversion method. Their feature conversion method is nearly identical to the feature selection technique employed in our proposed model. However, our proposed model outperformed theirs.

VII. Discussion

The machine learning detection approach was proposed and addressed in identifying a DDoS attack in this research study. This research has led to the understanding that many traditional machines learning, and deep learning methods can be used to detect a DDoS attack. However, when deciding whether to use traditional machine learning or deep learning with a large dataset, deep learning was considered due to its ability to solve difficult issues involving finding hidden patterns in data. It has a deep understanding of the complex relationships among a huge number of interdependent variables. Deep learning algorithms can create far more efficient decision rules. Deep learning is particularly effective in this study because the NSL-KDD dataset frequently requires dealing with unstructured data. Our findings reveal that LSTM is a nearly ideal strategy for preventing and protecting against 22 different types of

attacks. Classical machine learning, on the other hand, can be a preferable solution for smaller jobs that require less complex feature engineering and do not require the analysis of unstructured data.

VIII. Conclusion

The study has focused on presenting and demonstrating the design, implementation, and testing of a Detecting DDoS by Machine Learning solution to provide end-users with machine learning-based detection of DDoS attacks. End-users can re-route all traffic to an external server with DDoS mitigation capabilities hosted. The designed model solution allows researchers to build network-based detection models for network attacks using multiple machine learning methods, primarily classification. This result concludes that the objective to explore the type and study the characteristics of a DDoS attack from the viewpoint of machine learning was successfully achieved.

From the observation of the results for the proposed machine learning method, the LSTM RNN-based classification algorithm enhanced the detection of DDoS attacks. Pre-processing, attribute selection, and a detection and prevention system are the three components that the researcher proposes. The LSTM evaluation model fitting with the LSTM algorithm is demonstrated in the final phase. Significant testing was carried out, and the findings reveal that LSTM-RNN greatly surpasses existing DDoS attack detection systems. The algorithm learns the dataset's patterns with a 97.37% accuracy with a 0.0309 value loss. It was considerably easier to train numerous models in a short amount of time with TensorFlow, Google's second-generation machine learning framework. The LSTM recurrent neural network algorithm has been shown to have higher accuracy in detecting DDoS attacks in this study. These results covered achieving the objectives to improve DDoS attack detection approaches using a machine learning model to analyze network traffic activities for cyber threat detection; and to discover a feature selection strategy that, when combined with a machine learning system, can improve DDoS detection rates.

In a comparison of related work, the accuracy of our proposed method against all the other methods is significantly different. Our LSTM-RNN outperforms models that use all 41 features for training and testing in the NSL-KDD dataset. The proposed LSTM-RNN performed very well on the evaluation of the test data when compared to previously applied deep learning methods in Alkahtani et al. [22], Tang et al. [15], Niyaz et al. [23], Ugwu et al. [11]. This demonstrates how our method's phases are predictable, accurate, effective, and authoritative.

IX. Future Work

Even though DDoS packets or attacking packets are known, DDoS detection is not perfect. In the future, there will always be diverse approaches. However, this scope is not defined as one of the projects ' objectives that should be achieved. A solution for improvement could be to add more datasets to the proposed system; another feature Selection technique that can be used; And other classifiers could be added to improve attack detection; Use of confusion matrix to show the mistaken type of attacks. In the future direction of this research, I suggest using advanced deep learning algorithms to build a predictive analytics model to develop an automated system that can react based on current situations to analyze incoming data in networks. It could decide on defense mechanisms, evaluation and provide safety data on what is going on in a network.

References

[1] K. Singh, K. S. Dhindsa, and D. Nehra, "T-CAD: A threshold based collaborative DDoS attack detection in multiple autonomous systems," J. Inf. Secure. Appl., vol. 51, p. 102457, 2020.

[2] T. Ray, "DDoS defense: new tactics for a rising shadow industry," Network. Secure., vol. 2020, no. 4, pp. 6-7, 2020.

[3] S. Sambangi and L. Gondi, "A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression," Proceedings, vol. 63, no. 1, p. 51, 2020.

[4] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS Attack via Deep Learning," 2017 IEEE Int. Conf. Smart Comput. SMARTCOMP 2017, pp. 1-8, 2017.

[5] A. Guerra-Manzanares, H. Bahsi, and S. Nomm, "Hybrid feature selection models for machine learning based botnet detection in IoT networks," Proc. - 2019 Int. Conf. Cyberworlds, CW 2019, pp. 324-327, 2019.

[6] U. Sabeel, S. S. Heydari, H. Mohanka, Y. Bendhaou, K. Elgazzar, and K. El-Khatib, "Evaluation of Deep Learning in Detecting Unknown Network Attacks," 2019 Int. Conf. Smart Appl. Commun. Networking, SmartNets 2019, 2019.

[7] M. Rusyaidi and Z. Ibrahim, "A Review: An Evaluation of Current Artificial Intelligent Methods in Traffic Flow Prediction," IOP Conf. Ser. Mater. Sci. Eng., vol. 917, no. 1, 2020.

[8] M. S. Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, "DDoSNet: A Deep-Learning Model for Detecting Network Attacks," Proc. - 21st IEEE Int. Symp. a World Wireless, Mob. Multimed. Networks, WoWMoM 2020, pp. 391-396, 2020.

[9] F. O. Catak and A. F. Mustacoglu, "Distributed denial of service attack detection using autoencoder and deep neural networks," J. Intell. Fuzzy Syst., vol. 37, no. 3, pp. 3969-3979, 2019.

[10] J. D. Gadze, A. A. Bamfo-Asante, J. O. Agyemang, H. Nunoo-Mensah, and K. A.-B. Opare, "An Investigation into the Application of Deep Learning in the Detection and Mitigation of DDOS Attack on SDN Controllers," Technologies, vol. 9, no. 1, p. 14, 2021.

[11] C. C. Ugwu, O. O. Obe, O. S. Popoola, and A. O. Adetunmbi, "A distributed denial of service attack detection system using long short term memory with Singular Value Decomposition," Proc. 2020 IEEE 2nd Int. Conf. Cyberspace, CYBER Niger. 2020, pp. 112-118, 2021.

[12] Ö. KASIM, "An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks," Comput. Networks, vol. 180, no. June, 2020.

[13] Y. Gormez, Z. Aydin, R. Karademir, and V. C. Gungor, "A deep learning approach with Bayesian optimization and ensemble classifiers for detecting denial of service attacks," Int. J. Commun. Syst., vol. 33, no. 11, pp. 1-16, 2020.

[14] M. D. Hossain, H. Ochiai, D. Fall, and Y. Kadobayashi, "LSTM-based Network Attack Detection: Performance Comparison by Hyper-parameter Values Tuning," Proc. - 2020 7th IEEE Int. Conf. Cyber Secur. Cloud Comput, pp. 62-69, 2020.

[15] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi, and M. Ghogho, "Intrusion detection in sdn-based networks: Deep recurrent neural network approach," Adv. Sci. Technol. Secur. Appl., pp. 175-195, 2019.

[16] A. O. Alzahrani and M. J. F. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," Futur. Internet, vol. 13, no. 5, 2021.

[17] P. Nimbalkar and D. Kshirsagar, "Feature selection for intrusion detection system in Internet-of-Things (IoT)," ICT Express, vol. 7, no. 2, pp. 177-181, 2021.

[18] W. Nazih, Y. Hifny, W. S. Elkilani, H. Dhahri, and T. Abdelkader, "Countering ddos attacks in sip based voip networks using recurrent neural networks," Sensors (Switzerland), vol. 20, no. 20, pp. 1-15, 2020.

[19] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks," pp. 1-42, 2019.

[20] F. E. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," J. Big Data, vol. 8, no. 1, 2021.

[21] S. Althubiti, W. Nick, J. Mason, X. Yuan, and A. Esterline, "Applying Long Short-Term Memory Recurrent Neural Network for Intrusion

[22] Detection," Conf. Proc. - IEEE SOUTHEASTCON, vol. 2018-April, 2018.

[23] H. Alkahtani, T. H. H. Aldhyani, M. Al-Yaari, and M. Y. Alzahrani, "Adaptive Anomaly Detection Framework Model Objects in Cyberspace," 2020.

[24] Q. Niyaz, W. Sun, A. Y. Javaid, and M. Alam, "A deep learning approach for network intrusion detection system," EAI Int. Conf. Bio-inspired Inf. Commun. Technol., 2015.